

Утвержден

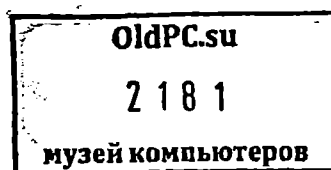
26.00152-01 31 01-ЛУ

Операционная система МОС ВП

Описание применения

26.00152-01 31 01

Листов 315/5К



1987

Перв. примен.

26.00152-01

Литера

Министерство приборостроения, средств автоматизации
и систем управления

Операционная система МОС ВП

Описание применения

Лист утверждения

26.00152-01 31 01-ЛУ

Представители предприятия-разработчика:

Руководитель темы
Зам. директора

----- Хрущев С.Н.
" " ----- 1987 г.

Ответственный исполнитель темы
Зав. отделением

----- Остапенко Г.П.
" " ----- 1987 г.

Зав. лабораторией

----- Аксенов А.В.
" " ----- 1987 г.

Ст. инженер

----- Иглина О.В.
" " ----- 1987 г.

Техник

----- Чачаева Т.Н.
" " ----- 1987 г.

Нормоконтролер

----- Скворцов В.А.
" " ----- 1987 г.

1987

Операционная система МОС ВП

Описание применения

00152-01 31 01

Листов 309

1987

АННОТАЦИЯ

Документ "Описание Применения" предназначен для начального знакомства пользователя с операционной системой МОС ВП.

В разделе 1 описывается назначение операционной системы МОС ВП и условия ее применения, указана конфигурация аппаратных средств, которые поддерживает система.

В разделе 2 приведены состав и основные функции программного средства.

В разделе 3 содержится описание задачи и кратко описаны компоненты операционной системы МОС ВП.

Раздел 4 содержит рекомендации по освоению операционной системы МОС ВП.

Документ содержит два приложения:

- 1) перечень устройств, поддерживаемых операционной системой МОС ВП;
- 2) основные термины МОС ВП.

СОДЕРЖАНИЕ

1.	НАЗНАЧЕНИЕ СИСТЕМЫ И УСЛОВИЯ ПРИМЕНЕНИЯ	8
2.	СОСТАВ И ФУНКЦИИ ПРОГРАММНОГО СРЕДСТВА	14
3.	ОПИСАНИЕ ЗАДАЧИ. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ	18
3.1.	Распределение ресурсов	20
3.2.	Защита	21
3.3.	Управление процессами	23
3.3.1.	Планирование процессов	23
3.3.1.1.	Приоритеты процессов	24
3.3.1.2.	Состояние процесса. Системные события	25
3.3.1.3.	Очереди состояний	26
3.3.1.4.	Выбор процесса для выполнения	26
3.3.1.5.	Механизм квантования времени	27
3.3.2.	Управление выполнением процесса	27
3.3.2.1.	Локальные флаги событий	28
3.3.2.2.	Асинхронные системные прерывания (AST)	28
3.3.3.	Средства межпроцессной связи	29
3.3.3.1.	Общие флаги событий	29
3.3.3.2.	Почтовые ящики	30
3.3.3.3.	Глобальные секции	30
3.4.	Управление памятью	31
3.4.1.	Виртуальное адресное пространство	32

3.4.2.	Отображение виртуальных страниц процесса на страницы физической памяти	32
3.4.3.	Преобразование виртуального адреса в физический	33
3.4.4.	Рабочий набор процесса. Обмен страниц	34
3.4.5.	Балансный набор. Обмен рабочих наборов	37
3.4.6.	Возможности управления виртуальной памятью, предоставляемые пользователю	38
3.5.	Организация ввода-вывода	39
3.5.1.	Вспомогательные управляющие процессы	42
3.5.2.	Драйверы устройств	42
3.6.	Система управления данными СУД-32	43
3.6.1.	Организация файлов	44
3.6.1.1.	Последовательная организация	45
3.6.1.2.	Относительная организация	46
3.6.1.3.	Индексная организация	46
3.6.2.	Методы доступа к записям	47
3.6.2.1.	Последовательный метод доступа	49
3.6.2.2.	Произвольный метод доступа по ключу или по относительному номеру записи	49
3.6.2.3.	Произвольный метод доступа по адресу записи в файле	49
3.6.3.	Обработка файлов и записей	50
3.7.	Программы системного обслуживания	51
3.7.1.	Программы системного обслуживания для управления процессами	52
3.7.2.	Программы системного обслуживания для	

	управления памятью	53
3.7.3.	Программы системного обслуживания ввода-вывода	53
3.7.4.	Программы системного обслуживания логических имен	54
3.7.5.	Программы системного обслуживания флагов событий	55
3.7.6.	Программы системного обслуживания асинхронных системных прерываний (AST)	55
3.7.7.	Программы системного обслуживания исключительных ситуаций	56
3.7.8.	Программы системного обслуживания таймера и преобразования времени	56
3.7.9.	Программы системного обслуживания для изменения режима	57
3.8.	Командные интерфейсы	57
3.8.1.	Перечень команд диалогового командного языка (DCL)	59
3.8.1.1.	Информационные и управляющие команды	59
3.8.1.2.	Команды управления пакетом и командными процедурами	62
3.8.1.3.	Команды управления томами и устройствами	64
3.8.1.4.	Команды для работы с файлами	65
3.8.1.5.	Команды для разработки программ и управления их выполнением	67
3.9.	Режим совместимости	70

3.10.	Наборы инструкций	74	
3.10.1.	Типы данных	76	
3.10.2.	Режимы адресации	81	
3.10.3.	Инструкции основного режима	84	
3.11.	Системы программирования	95	
3.11.1.	Система программирования на языке Макроассемблер СМ 1700	96	
3.11.2.	Система программирования ФОРТРАН	97	
3.11.3.	Система программирования КОБОЛ	98	
3.11.4.	Система программирования БЕЙСИК	98	
3.11.5.	Система программирования ПЛ/1	99	
3.11.6.	Система программирования ПАСКАЛЬ	99	
3.11.7.	Система программирования КОРАЛ	100	
3.11.8.	Система программирования БЛИСС	101	
3.11.9.	Система программирования МОДУЛА-2	102	
3.11.10.	Система программирования СИ	102	
3.11.11.	Система программирования ДИАМС/МОС ВП	103	
3.12.	Средства разработки программ	103	
3.13.	Обеспечение надежности функционирования МОС ВП	107	
4.	РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ	108	
Приложение 1.			
	Перечень устройств, поддерживаемых операционной системой МОС ВП	109	
Приложение 2. Основные термины МОС ВП			113

Перечень ссылочных документов

308к

1. НАЗНАЧЕНИЕ СИСТЕМЫ И УСЛОВИЯ ПРИМЕНЕНИЯ

Операционная система МОС ВП (латинский эквивалент названия VIRTUAL MEMORY OPERATING SYSTEM - VOS) является многофункциональной операционной системой, поддерживающей виртуальную память для 32-разрядной малой ЭВМ СМ 1700, и предназначена для широкого применения в различных отраслях народного хозяйства. Она позволяет использовать технические средства СМ ЭВМ, включая разработки 3-ей очереди, для решения разнообразных задач проектирования на базе высокопроизводительных АРМ'ов, автоматизации управления производством, научного эксперимента, решения информационно-поисковых и информационно-аналитических задач, задач организационно-управленческого характера и других, где требуется одновременное выполнение значительного числа заданий с интегрированной информационной моделью объекта управления.

Операционная система МОС ВП (далее по тексту МОС ВП) обеспечивает пользователя развитыми средствами подготовки, отладки и выполнения программ, написанных на языках программирования МАКРОАССЕМБЛЕР, ФОРТРАН, БЕЙСИК, КОБОЛ, ПАСКАЛЬ, ПЛ/1, БЛИСС, СИ, МОДУЛА-2, КОРАЛ, в режимах разделения времени, реального времени и в пакетном режиме.

Под управлением МОС ВП могут успешно выполняться задачи различных типов, включая задачи, интенсивно использующие центральный процессор, задачи с большой нагрузкой на ввод-вывод, задачи реального времени, а также любые комбинации этих задач. Максимальное количество параллельных про-

00152-01 31 01

цессов в МОС ВП составляет 8192. Однако, реальное количество задач, выполнение которых может обеспечиваться одновременно при хорошей производительности, зависит от ресурсов вычислительного комплекса.

МОС вп функционирует при следующей минимальной конфигурации технических средств:

- процессор СМ 1700;
- озу емкостью 1Мбайт;
- накопитель на сменных магнитных дисках емкостью 14 Мбайт СМ 5408 - 2 штуки;
- консольное устройство загрузки на кассетной магнитной ленте СМ 5218;
- консольное устройство печати с клавиатурой СМ 6380;
- алфавитно-цифровой дисплей СМ 7224 (ВТА 2000-17).

Дополнительно МОС ВП поддерживает следующие устройства:

- алфавитно-цифровой дисплей с графическими возможностями СМ 7225 (ВТА 2000-18);
- видеотерминал растровый графический цветной СМ 7317;
- накопитель на магнитных дисках типа "Винчестер" емкостью 150 Мбайт СМ 5504;
- внешнее запоминающее устройство емкостью 20 Мбайт на базе накопителя на магнитных дисках типа "Винчестер" СМ 5514;
- устройство внешней памяти на магнитных лентах емкостью 40 Мбайт СМ 5309, СМ 5316;
- алфавитно-цифровое устройство параллельной печати с

расширенным набором символов.

Полный перечень устройств, поддерживаемых МОС ВП, и их символические обозначения представлены в Приложении 1.

МОС ВП является диск-резидентной операционной системой. Часть системы помещается в память ЭВМ в процессе загрузки. Остальные части загружаются в основную память и удаляются оттуда по мере необходимости, в результате чего достигается гибкая настройка под конкретный режим работы. При минимальной конфигурации технических средств объем резидентной части МОС ВП не превышает 150 Кбайт.

МОС ВП поставляется в сгенерированном виде на магнитной ленте. Вместе с операционной системой пользователь получает автономную программу для восстановления системного диска с магнитной ленты. После восстановления системного диска выполняется процедура запуска, которая производит автоматическую настройку операционной системы на конкретную конфигурацию технических средств.

После процедуры запуска пользователь может работать с операционной системой с любого терминала. В составе МОС ВП имеется процедура для нормального завершения работы (эта процедура сообщает пользователям, работающим в диалоговом режиме, о необходимости отключения, останавливает дисководы и т. д.).

Для обеспечения связи пользователей с ЭВМ, эффективной работы в интерактивном, пакетном и режиме реального времени, а также планирования и управления ресурсами вычислительного комплекса (далее по тексту вк) МОС ВП предостав-

ляет следующие возможности:

- программист может создавать, транслировать и выполнять программы, написанные на различных языках программирования, использовать систему управления данными СУД-32, доступную из любого языка программирования и обеспечивающую независимую от устройства обработку данных. Программы системного обслуживания, предназначенные в основном для системных программистов, также могут быть доступны программисту для запроса специальных средств в операционной системе;
- системный программист имеет средства для разработки структурированных программ и прикладных систем, может создавать библиотеки макроопределений, объектных модулей и модулей образа, добавлять новые процедуры в общую исполнительную библиотеку и работать с ними, не модифицируя существующие программы и утилиты, использовать программы системного обслуживания и возможности присвоения логических имен для написания программ, не зависящих от устройства;
- администратор системы определяет полномочия пользователей, управляет доступом к данным и их защитой, предоставляет привилегии, контролирует использование ресурсов, анализирует системную информацию. Информация о полномочиях пользователей записывается в специально поддерживаемый и защищенный файл, называемый файлом авторизации пользователя. Администратор может создавать, просматривать и обновлять этот файл в

любое время;

- оператор осуществляет запуск и прекращение работы МОС ВП, управляет заданиями (изменение приоритета процесса, уничтожение заданий и т.д.), обслуживает запросы на монтирование и размонтирование томов, управляет очередями на пакетную обработку и печать, запускает диагностические программы. Специальная системная программа выдачи сообщений оператору (ОРСОМ) является главным помощником оператора. ОРСОМ собирает и выдает сообщения всех пользователей и пользовательских программ, посылаемые операторам. Любой оператор может отвечать на запросы пользователя, а ОРСОМ будет напоминать операторам о необработанных запросах.

В МОС ВП собирается статистика о ее работе, помогающая системным программистам и администраторам добиваться максимальной производительности. Статистическая информация включает:

- данные об операционной системе и заданиях: текущее количество процессов, интерактивных пользователей, пакетных заданий, дата и время загрузки, а также текущая дата и время;
- данные об использовании режимов работы процессора: время работы процессора в каждом режиме (для определения загрузки вычислительной системы);
- данные о страничных отказах: количество и типы страничных отказов (для эффективного управления

памятью);

- данные об операциях ввода-вывода: информация о типах и о количестве операций ввода-вывода;
- данные о загрузке сети;
- гистограммы времени реакции: время, которое требуется МОС ВП для инициализации запросов пользователя.

Применяя соответствующие команды, пользователи системы могут получать информацию о состоянии устройств ввода-вывода, страничном обмене, использовании центрального процессора, состоянии текущего процесса и т.д.

2. СОСТАВ И ФУНКЦИИ ПРОГРАММНОГО СРЕДСТВА

МОС ВП является иерархической операционной системой и обеспечивает многоуровневое обслуживание программ пользователя. Всего таких уровней четыре - по числу режимов доступа к процессору в ВК СМ 1700 (режим ядра, режим управления, режим супервизора и режим пользователя). На режимах доступа к процессору в МОС ВП основана одна из форм защиты: программы и данные каждого уровня защищены от вмешательства программ менее привилегированных уровней.

Самый привилегированный уровень (режим доступа к процессору - режим ядра) занимают программы, непосредственно управляющие использованием аппаратных средств. Они составляют ядро МОС ВП.

В состав ядра входят компоненты:

- управление процессами;
- управление памятью;
- управление вводом-выводом;
- программы системного обслуживания, выполняющие роль интерфейса внешних уровней с компонентами ядра.

Следующий уровень занимает система управления данными СУД-32 (режим доступа к процессору - режим управления), реализующая различные способы организации и обработки данных, хранящихся на внешних запоминающих устройствах.

Уровень, соответствующий режиму супервизора, занимает интерпретатор командного языка, обеспечивающий непосредственный интерфейс с пользователем.

00152-01 31 01

Наименее привилегированный уровень (режим доступа к процессору - режим пользователя) занимают:

- средства разработки программ;
- утилиты;
- программы, написанные пользователем.

Операционная система МОС ВП обеспечивает выполнение следующих функций:

- управление виртуальной памятью объемом до 4,3 Гбайт;
- управление оперативной памятью от 1 до 5 Мбайт, динамически распределяемой между многими пользователями;
- работу многих пользователей в режиме реального времени, разделения времени и в пакетном режиме одновременно;
- выполнение до 8192 процессов одновременно;
- выполнение непривилегированных задач, разработанных под управлением операционной системы ОС РВ для 16-разрядных СМ ЭВМ с магистральной архитектурой, в режиме совместимости ЭК СМ 1700;
- выполнение программ пользователя объемом до 1 Гбайта;
- управление выполнением задач по приоритетам и с помощью квантования времени процессора;
- изменение приоритетов по инициативе оператора или программы;
- автоматическую подстройку приоритетов задач, не работающих в режиме реального времени;

00152-01 31 01

- выполнение программ пользователя, составленных из модулей, написанных на разных языках программирования;
- запуск задач по инициативе оператора или других задач;
- автоматический рестарт операционной системы после сбоя;
- управление машинными часами и таймером;
- управление доступными пользователю ресурсами вычислительного комплекса;
- обслуживание операций ввода-вывода в реальном масштабе времени;
- взаимодействие задач пользователей;
- управление файлами с последовательной, относительной и индексно-последовательной организацией;
- управление файлами на различных внешних накопителях: магнитных дисках, гибких дисках, магнитных лентах;
- возможность организации сетей ЭВМ на базе ВК СМ 1700 и других моделей СМ ЭВМ с магистральной архитектурой;
- возможность подключения разработанных пользователем драйверов дополнительных внешних устройств;
- средства анализа функционирования операционной системы и средства подстройки операционной системы с целью повышения эффективности ее пропускной способности;
- генерацию операционной системы под конкретные функ-

00152-01 31 01

циональные требования и состав технических средств.

Кроме того, функциональные возможности МОС ВП расширяют следующие программные средства, работающие под ее управлением:

- системы управления базами данных;
- средства построения и взаимодействия с открытыми сетями ЭВМ и сетями СМ-ЕС ЭВМ;
- программы-эмуляторы терминалов ЕС ЭВМ;
- базовые графические средства.

Данные программные средства разработаны в рамках отдельных тем и представляют собой автономно поставляемые компоненты.

3. ОПИСАНИЕ ЗАДАЧИ. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Основной задачей МОС ВП, как операционной системы с мультипрограммированием, является обеспечение одновременного нахождения в основной памяти нескольких программ пользователей и разделение времени процессора, пространства памяти и других ресурсов между активными пользовательскими заданиями, а также защита заданий пользователя во время их взаимодействия и совместного использования программ и данных.

Программа, готовая к выполнению, называется образом. Образ создается путем трансляции исходного текста программы и компоновки полученных объектных модулей.

Условия выполнения образа определяются контекстом. Контекст состоит из аппаратного контекста (состояние регистров процессора) и программного контекста (описание всех выделенных процессу ресурсов).

Два или более пользователей могут выполнять некоторый образ совместно, т.е. образ может быть разделен, и выполняется в двух или более различных контекстах.

Основной единицей планирования в МОС ВП является процесс. Процесс состоит из контекста и виртуального адресного пространства, в котором выполняется образ.

Процессы могут быть запущены следующими способами:

- регистрация пользователя. Когда пользователь, работающий в диалоговом режиме, включается в систему, система создает для него процесс. Этот процесс соз-

дает окружение, в котором пользователь может взаимодействовать с системой для указания имен модулей, которые должны быть выполнены, а также для выполнения других действий:

- явное создание. Любой пользователь может указать создание нового процесса. При создании процесса пользователь указывает имя модуля, который будет выполнен при старте процесса;
- пакетная обработка. Любой пользователь может указать имена модулей, которые должны быть выполнены, и предоставить эту информацию системе в качестве пакетного задания с терминала.

Процесс может создавать подпроцессы. Процесс, который создает и запускает подпроцессы, называется процессом-владельцем. Он управляет выполнением подпроцессов и устанавливает, какие из его привилегий будут разрешены подпроцессу. Процесс-владелец может управлять формированием подпроцессов, а также уничтожать подпроцесс. Когда процесс-владелец заканчивается, удаляются все принадлежащие ему подпроцессы.

Процесс и все его подпроцессы составляют задание. Все процессы, включенные в задание, планируются независимо, но они разделяют общие ресурсы, выделенные этому заданию. Задание является основной учетной единицей в МОС ВП.

Задания могут объединяться в группы. Группы являются взаимоисключающими, так что, если задание принадлежит одной группе, оно не принадлежит любой другой группе. Процесс с соответствующей привилегией может управлять выполнением

других процессов в некоторой группе. Процессы одной группы могут использовать средства межпроцессной связи.

3.1. Распределение ресурсов

Ресурсы вычислительного комплекса - процессор, физическая память и периферийные устройства. МОС ВП управляет многими заданиями одновременно, и каждое задание может требовать различные ресурсы. Операционная система дает возможность заданиям распределять ресурсы согласно их индивидуальным требованиям и защищает каждое задание, его данные от других заданий в системе.

МОС ВП управляет распределением ресурсов динамически, осуществляя управление памятью, распределение устройств и обработку ввода-вывода, и статически - через авторизацию пользователей.

Администратор системы отвечает за создание записи в файле авторизации для каждого пользователя. Файл авторизации обеспечивает операционную систему квотами и лимитами ресурсов для каждого задания. Например, существуют квоты и лимиты, которые управляют:

- использованием времени процессора;
- числом подпроцессов, которые задание может создавать;
- числом одновременно открытых файлов;
- использованием процессом виртуальной и физической памяти;
- числом одновременных передач ввода-вывода.

Кроме того, в файле авторизации определяются привилегии каждого пользователя. Существует 64 различных привилегии, которые могут быть индивидуально предоставлены пользователю. Среди них есть привилегии, которые дают заданию право:

- изменить приоритет процесса;
- выполнить программу, написанную пользователем, в более привилегированном режиме доступа;
- выполнять функции оператора;
- получить средства межпроцессной связи;
- управлять выполнением других процессов в группе.

При выполнении образ может получать только такие привилегии и квоты, которые предоставлены непосредственно данному пользовательскому заданию в файле авторизации, если этот образ не является известным образом. Известные образы устанавливаются администратором системы, и пока они выполняются, они обеспечивают вторичное динамическое множество привилегий, предоставленных пользователю.

При выполнении известного образа процесс имеет привилегии и квоты, предоставленные пользователю в файле авторизации, плюс дополнительные привилегии выполнения, предоставленные специально этому образу.

3.2. Защита

Основа защиты данных в МОС ВП - код идентификации пользователя (UIC). UIC содержит два числа: номер группы и номер члена в группе. Администратор системы назначает для

00152-01 31 01

каждого пользователя UIC в файле авторизации пользователя. Образам назначаются или отменяются привилегии доступа к данным, основанные на UIC пользователя.

Когда создается файл или средство межпроцессной связи, им назначается UIC и код защиты. UIC определяет, какая группа пользователей или программ и какой член в группе управляют доступом к данным.

Код защиты используется для 4-х типов доступа (читать, писать, выполнять, уничтожать).

Каждый тип доступа может быть задан или отменен для:

- владельца: пользователя, чей UIC такой же, как UIC, назначенный данным;
- группы: каждого пользователя, чей номер группы в UIC такой же, как у владельца;
- системы: каждого пользователя, чей номер группы UIC является привилегированным системным номером группы;
- всех: каждого пользователя, чей номер группы UIC, отличается или такой же, как UIC назначенный данным.

Используя данную схему защиты, МОС ВП может иметь файлы и средства межпроцессной связи, доступ к которым разрешен только пользователям, имеющим некоторый UIC, или только пользователям некоторой группы, или всем пользователям. Кроме того, так как каждая структура данных имеет свой собственный код защиты, можно защищать каждую структуру данных с некоторым UIC на различной основе. Системные UIC в основном резервируются для системных программистов и администратора, системных программ и данных. Это позволяет

пользователю защищать файл от доступа всех, кроме владельца или группы, и разрешает системе доступ к файлам для таких операций как сжатие.

Доступ по чтению и записи применяется как для файлов, так и для средств межпроцессной связи. Доступ по уничтожению используется только для файлов, а доступ по выполнению применяется только к файлам программного образа.

Привилегии и квоты, предоставленные в файле авторизации пользователя, управляют созданием и уничтожением средств межпроцессной связи.

3.3. Управление процессами

МОС ВП обеспечивает совместное выполнение многих процессов, осуществляя их планирование, основанное на принципе наивысшего приоритета, а также предоставляя средства для управления выполнением процессов и обеспечения межпроцессного взаимодействия.

3.3.1. Планирование процессов

Процессы получают время процессора для выполнения своих образов на основе принципа приоритетной работы, управляемой событиями. Всякий раз, когда происходит событие, такое как прерывание по завершению ввода-вывода, мос вп обслуживает это событие, а затем передает управление наиболее приоритетному процессу из числа готовых к выполнению (даже если процесс, имевший управление до этого собы-

тия, может продолжать выполнение).

Планирование процессов осложняется ограниченностью физической памяти ЭВМ. Если существующее число процессов слишком велико для размещения в физической памяти, то система хранит некоторые из процессов на диске. Чтобы хранящийся на диске процесс смог продолжить свое выполнение, он должен быть вновь размещен в памяти. Загрузку процессов в физическую память и выгрузку их на диск осуществляет специальный системный процесс обмена, называемый процессом обмена рабочих наборов.

3.3.1.1. Приоритеты процессов

В целях планирования в МОС ВП определено 32 уровня программных приоритетов (от 0 до 31), где 31 - высший приоритет. Приоритеты с 0 по 15 используются для планирования обычных процессов (т.е. процессов, не критических по времени, и работающих в режиме разделения времени). Приоритеты с 16 по 31 выделены для процессов реального времени.

При создании процесса ему присваивается некоторый базовый приоритет. Приоритет процессов реального времени остается неизменным на весь период выполнения. Приоритет обычных процессов автоматически подстраивается планировщиком. В каждый момент времени процесс имеет некоторый текущий приоритет (не ниже базового и не выше 15), изменяющийся всякий раз при возникновении некоторого системного события на величину приращения, соответствующую этому событию.

При планировании процесса на выполнение его текущий приоритет уменьшается на 1, а при завершении состояния ожидания приоритет процесса увеличивается. (Наибольшую величину приращения имеют события окончания ввода-вывода с терминала).

Такой механизм подстройки приоритетов позволяет создать наиболее выгодную ситуацию для процессов с большой нагрузкой на ввод-вывод, т.е. процессы вычислительного характера практически постоянно находятся на своем базовом приоритете, а процессы, выполняющие частые операции ввода-вывода, будут иметь приоритет гораздо больше базового, что позволит им прерывать выполнение вычислительных процессов.

3.3.1.2. Состояние процесса. Системные события

В каждый момент времени процесс находится в определенном состоянии, например, готов к выполнению, ожидает события и т.д. Все возможные состояния процесса являются взаимоисключающими. Изменение состояния процесса обусловлено возникновением так называемого системного события.

Системное событие - это событие, которое влияет на возможность выполнения процесса в системе. Системные события могут быть внешними для текущего выполняемого процесса, например, завершение ввода-вывода и прерывание таймера. Системные события также могут быть внутренними для текущего выполняемого процесса: процесс может выдать запрос на ожидание или запрос на приостановку, запросить или освободить

системный ресурс, например, страницу памяти.

3.3.1.3. Очереди состояний

Для каждого состояния существует очередь процессов, находящихся в этом состоянии. Процесс представлен в очереди своим программным блоком управления (РСВ) , являющимся частью контекста процесса.

Для готовых к выполнению процессов существует два набора очередей состояния:

- 1) для процессов, которые находятся в физической памяти;
- 2) для процессов, которые находятся на диске.

Каждый набор состоит из 32 очередей (по одной для каждого приоритетного уровня), что ускоряет поиск готового к выполнению процесса с наивысшим приоритетом.

Остальные очереди предназначены для процессов, которые ожидают выполнения некоторого условия для перехода в состояние готовности к выполнению. Эти очереди содержат все подобные процессы независимо от приоритета.

При изменении состояния процесса его РСВ удаляется из одной очереди и помещается в другую.

3.3.1.4. Выбор процесса для выполнения

Для выполнения всегда выбирается процесс, который находится первым в очереди готовых к выполнению резидентных процессов с наивысшим приоритетом.

При прерывании выполнения текущего процесса (появился готовый к выполнению процесс с более высоким приоритетом, истек квант времени) его состояние сохраняется, и он перемещается в конец очереди соответствующего приоритета, что обеспечивает чередование процессов в очереди и равномерное распределение времени процессора между процессами с одинаковым приоритетом. Затем из очереди снова выбирается самый приоритетный процесс, который становится текущим.

3.3.1.5. Механизм квантования времени

Каждый процесс независимо от приоритета получает квант времени (определяющий время его пребывания в физической памяти), что обеспечивает процессу некоторый минимальный интервал времени для выполнения полезной работы и дает возможность обычным процессам (с приоритетом меньше 16) поочередно использовать процессор.

На работу процессов реального времени, квантование не влияет.

3.3.2. Управление выполнением процесса

Управление выполнением процесса осуществляется при помощи:

- локальных флагов событий;
- асинхронных системных прерываний (AST).

Кроме того, программы системного обслуживания обеспечивают управление процессом, позволяя процессу:

- создавать и ликвидировать подпроцессы;
- приостанавливать и возобновлять выполнение процесса.

3.3.2.1. Локальные флаги событий

Локальный флаг события - это бит состояния, используемый для регистрации события, например, такого, как завершение операции ввода-вывода или истечение интервала времени. Каждый процесс имеет в своем распоряжении два кластера локальных флагов событий (по 32 флага в каждом). Процесс может установить, очистить и прочесть флаги событий, а также ожидать установления одного или нескольких локальных флагов событий.

3.3.2.2. Асинхронные системные прерывания (AST)

Асинхронное системное прерывание (AST) - это программно-моделируемое прерывание, используемое для обозначения события в процессе. Программа AST - это программа, которая получает управление при возникновении некоторого аСТ. Программы AST обеспечивают эффективные средства для обработки событий, которые могут возникать асинхронно с выполнением процесса. Например, можно определить программы AST для обработки запросов ввода-вывода и восстановления сбоя питания. При возникновении этих событий операционная система объявляет AST. Когда выдается AST, операционная система прерывает процесс и выполняет соответствующую программу AST.

3.3.3. Средства межпроцессной связи

МОС ВП предоставляет процессам средства межпроцессной связи для синхронизации выполнения, передачи сообщений, разделения общих данных, а именно:

- общие флаги событий;
- почтовые ящики;
- глобальные секции.

Защита и право доступа к этим средствам определяются привилегиями процесса и кодом идентификации пользователя.

3.3.3.1. Общие флаги событий

Помимо существующих для каждого процесса локальных флагов событий процессы одной группы могут использовать для синхронизации своей работы общие флаги событий.

Каждый процесс в группе может быть связан с двумя кластерами общих флагов событий (каждый кластер содержит 32 флага). Процесс может читать, устанавливать, очищать или ожидать установки общих флагов событий.

Например, один процесс в группе может установкой некоторого флага отмечать завершение определенной функции, а другой процесс или процессы могут ожидать (или проверять), когда этот флаг будет установлен.

3.3.3.2. Почтовые ящики

Почтовый ящик - это ориентированное на записи виртуальное устройство ввода-вывода, созданное процессом. Почтовые ящики могут использоваться процессами для обмена сообщениями и передачи информации о состоянии. Процесс может защищать свои почтовые ящики от возможности чтения и/или записи другими процессами вне его группы. Процесс может запросить извещение AST, когда что-нибудь записывается в его почтовые ящики, и может назначать почтовому ящику логическое имя.

Для чтения и записи информации в почтовые ящики используются операторы ввода-вывода языков высокого уровня, процедуры системы управления данными СУД-32 или программы системного обслуживания ввода-вывода.

3.3.3.3. Глобальные секции

МОС ВП поддерживает высокую степень разделения программ и данных, используя глобальные секции. Глобальная секция - это часть памяти (копия всего или части файла образа или данных), доступная нескольким процессам одновременно.

Глобальные секции могут создаваться процессом динамически или могут быть постоянно представлены в системе. Глобальные секции могут разделяться всеми процессами системы, или только процессами конкретной группы, или только процессами в конкретном задании. Один или более процессов могут иметь к глобальной секции доступ по записи, в то время как

для других процессов в системе, группе или задании возможен только доступ по чтению.

3.4. Управление памятью

Одной из главных функций операционной системы является распределение и обслуживание ресурсов вычислительной системы. Одним из наиболее важных ресурсов является физическая память. В мультипрограммной системе память совместно используется (разделяется) несколькими процессами. Число и вид совместно выполняющихся процессов, а соответственно и их требования к памяти, изменяются быстро, в то время как объем памяти, доступный процессам, остается неизменным. Задачей операционной системы становится динамическое распределение памяти на уровне процесса и между отдельными процессами.

Способ организации памяти, используемый в МОС ВП, известен как виртуальная память. Виртуальная память представляет собой совокупность физической (основной) памяти эвм и дисковой памяти, обращение к которой осуществляется посредством виртуальной адресации. При таком способе организации памяти не используемые в данный момент части программы и данных хранятся на диске и перемещаются в основную память по мере необходимости.

Задачей средств управления памятью является создание и поддержка информации, используемой для преобразования виртуальных адресов программы в адреса физической памяти.

3.4.1. Виртуальное адресное пространство

Размер виртуального адресного пространства в МОС ВП составляет 2^{**32} байтов. Это пространство разделено на две части - адресное пространство системы и адресное пространство процесса (каждое размером 2^{**31} байтов).

В свою очередь, адресное пространство процесса разделено на область программы (P0) и область управления (P1). Область программы (от 0 до 2^{**30} байтов) содержит образ, выполняемый в настоящий момент. В области управления (от 2^{**31} до 2^{**30} байтов) находится управляющая информация для процесса, поддерживаемая системой, такие, например, структуры данных, как стеки всех четырех режимов доступа к процессору.

Виртуальное адресное пространство системы располагается в первой половине адресного интервала от 2^{**31} до 2^{**32} . Здесь содержатся структуры данных, необходимые для управления всеми процессами и обслуживания всех физических и виртуальных страниц системы.

3.4.2. Отображение виртуальных страниц процесса на страницы физической памяти

Виртуальная память разбита на страницы. Каждая виртуальная страница отображается на страницу физической памяти или на блок на диске (все они имеют одинаковый размер - 512 байтов). На одну физическую страницу может последовательно отображаться любое количество виртуальных страниц.

выделение физических страниц производится динамически, поэтому страницы, относящиеся к одному процессу, могут быть расположены физически не непрерывно. В то время как в виртуальной памяти они могут быть смежными.

Для обслуживания связи между виртуальными страницами процесса и физическими страницами памяти, на которые они отображаются, используются таблицы страниц, создаваемые и поддерживаемые средствами управления памятью.

Для каждого процесса поддерживается две таблицы страниц: для области программы и области управления виртуального адресного пространства процесса. Каждой странице области соответствует элемент таблицы страниц (PTE), идентифицирующий ее состояние. Если виртуальная страница находится в физической памяти, то PTE содержит номер физической страницы, если виртуальная страница не находится в физической памяти, то pte содержит информацию, необходимую для поиска страницы на диске.

Виртуальное адресное пространство системы описывается системной таблицей страниц. С помощью этой таблицы можно определить местоположение любой виртуальной страницы системной области.

3.4.3. Преобразование виртуального адреса в физический

Во время выполнения процесса происходит преобразование виртуальных адресов в физические. Необходимая для этого информация содержится в таблицах страниц.

3.4.4. Рабочий набор процесса. Обмен страниц

При выполнении процесса в каждый момент времени только часть его страниц находится в физической памяти. Рабочий набор процесса представляет собой совокупность расположенных в физической памяти виртуальных страниц процесса, к которым процесс может непосредственно обращаться, не вызывая прерываний из-за отсутствия страниц. Остальные страницы данного процесса находятся или в физической памяти, но не в рабочем наборе, или на диске (в файле образа или в файле обмена страниц).

Рабочий набор — динамическая характеристика процесса, он имеет предельные значения максимального и минимального размеров. Система определяет минимальное требуемое число страниц, которые должны быть в рабочем наборе процесса, а администратор системы определяет максимальное число страниц, допустимых в рабочем наборе процесса, в файле авторизации пользователя. Размер рабочего набора процесса влияет на эффективность использования виртуальной памяти и на количество рабочих наборов процессов.

Процесс может увеличить или уменьшить свой рабочий набор в допустимых пределах с помощью команд диалогового языка или программ системного обслуживания.

Когда процесс обращается к виртуальному адресу, находящемуся в странице, которой в настоящее время нет в физической памяти, происходит прерывание по страничному отказу. Прерывание запускает программу обмена страниц, которая делает необходимую страницу доступной в физической памяти,

что позволяет процессу продолжить работу.

Обмен страниц в МОС ВП имеет несколько особенностей:

- возможность читать несколько страниц одновременно;
- страничный обмен процесса происходит за счет самого процесса, не затрагивая систему в целом;
- поддержка пула страниц, из которого процесс может получить недавно удаленные страницы без обращения к диску;
- запись на диск только модифицированных страниц, которые освобождены из рабочего набора процесса и запись их по мере накопления;
- активизация процесса, ожидающего ввода-вывода по страничному отказу, для выполнения программ, АСТ, когда они выдаются.

Когда система впервые активизирует образ, в память читаются страницы из файла образа на диске. Можно организовать кластерное считывание страниц, что позволяет образу некоторое время выполняться без возникновения страничных отказов.

Впоследствии процесс обменивает страницы только при выполнении образа, который требует больше страниц, чем процесс может иметь в своем рабочем наборе.

Программа обмена страниц работает только со страницами данного процесса. Она не освобождает страницы одного процесса для удовлетворения потребностей другого. Тем самым обеспечивается то, что только те процессы, которые нуждаются в страничном обмене, подвергаются процедуре обмена стра-

ниц.

Если не все страницы считаны сразу, то в некоторый момент может произойти обращение к странице, которая не была считана. Возникает страничный отказ. Если лимит размера рабочего набора еще не достигнут, программа обмена страниц считывает требуемую страницу с диска.

Если рабочий набор полон, то для помещения в физическую память требуемой страницы, необходимо какую-либо другую страницу из рабочего набора удалить.

Освобожденная из рабочего набора страница помещается в один из следующих списков:

- список свободных страниц;
- список модифицированных страниц.

Список свободных страниц используется для следующих целей:

- служит источником страниц, доступных для нового использования;
- выполняет роль кэша для страниц, относительно недавно удаленных из рабочего набора. Если процессу вновь потребуется страница, которая уже попала в список свободных, но еще не была использована, программа обмена страниц вернет эту страницу из списка свободных в рабочий набор процесса.

Список модифицированных страниц содержит страницы, содержимое которых должно быть записано на диск, перед тем, как эту страницу можно будет использовать для размещения другой виртуальной страницы. Этот список также служит в

качестве резерва страниц, которые могут быть легко возвращены в рабочий набор.

Использование этих структур данных позволяет сокращать обращение к дисковой памяти и увеличивать производительность системы.

3.4.5. Балансный набор. Обмен рабочих наборов

Совокупность рабочих наборов процессов, находящихся в физической памяти, называется балансным набором.

Суммарные потребности этих наборов в физической памяти должны быть сбалансированы с размерами физической памяти эвм. Если требования на физическую память не могут быть удовлетворены за счет изменения размеров рабочих наборов процессов, то рабочие наборы этих процессов (низкоприоритетных или невыполняющихся) могут быть полностью переписаны (выгружены) на диск для того, чтобы освободить место для более приоритетного процесса.

Пересылку рабочих наборов между балансным набором и внешней памятью на диске выполняет процесс обмена рабочих наборов.

Главная задача этого процесса - обеспечить размещение в балансном наборе высокоприоритетных процессов, готовых к выполнению, что обеспечивает эффективный способ расширения ограниченных ресурсов памяти.

3.4.6. Возможности управления виртуальной памятью, предоставляемые пользователю

Процессор обеспечивает программиста большим виртуальным адресным пространством и быстрым преобразованием адреса. Операционная система МОС ВП обеспечивает программиста максимально эффективными алгоритмами отображения и страничного обмена. Компиляторы языков высокого уровня и компоновщик используют механизмы управления памятью, оптимально распределяя память для большинства требований программирования.

МОС вп обеспечивает пользователя средствами управления и оптимизации управления памятью. Администратор системы может управлять запросами на распределение памяти системы в целом, устанавливая такие параметры, как требуемое и минимальное количество доступных страниц, и параметры для индивидуальных заданий пользователя, такие как максимальный размер рабочего набора. Можно создать процесс, которому не нужен страничный обмен, создав для него рабочий набор с размером, равным требуемой виртуальной памяти, или уменьшить страничный обмен, выбрав размер рабочего набора, удовлетворяющий среднему запросу в страницах.

Программист может оказывать влияние на распределение памяти следующими способами:

- правильно разрабатывать структуру программы;
- использовать программы системного обслуживания.

Эти средства позволяют оптимизировать расположение программы. Редко подвергаются страничному обмену те прог-

раммы, в которых код и данные, используемые во время каждой стадии обработки, содержатся в возможно меньшем числе виртуально смежных страниц.

3.5. Организация ввода-вывода

МОС ВП поддерживает устройства ввода-вывода из номенклатуры технических средств СМ ЭВМ, включая накопители на магнитных дисках и магнитных лентах, алфавитно-цифровые печатающие устройства и дисплеи. Кроме того, операционная система предоставляет пользователям возможность достаточно простого включения в конфигурацию вычислительного комплекса собственных (нестандартных) устройств ввода-вывода.

Выполнение операции ввода-вывода начинается с задачи процессом запроса на ввод-вывод в канале, который был предварительно назначен операционной системой для конкретного устройства ввода-вывода. Канал - это логический путь, связывающий процесс пользователя с необходимыми ему устройствами. Запросы ввода-вывода ставятся в очередь в соответствии с приоритетом; запросы одного и того же приоритета обслуживаются по принципу "первым пришел - первым обслужен".

Запросы ввода-вывода могут выполняться на следующих уровнях:

- диалоговый командный язык (DCL). Такие команды как EDIT, CREATE, APPEND, COPY, PRINT, SORT и DELETE обеспечивают возможность манипулировать файлами;
- программы, написанные на языках программирования КОБОЛ, ФОРТРАН, БЕЙСИК, ПАСКАЛЬ, а также на других

00152-01 31 01

языках высокого уровня, могут осуществлять ввод-вывод, используя операторы данных языков программирования;

- система управления данными СУД-32. Процедуры системы управления данными СУД-32 могут вызываться из программ, написанных на языке макроассемблер и на языках программирования высокого уровня, для выполнения ввода-вывода, не зависящего от устройств;
- программы системного обслуживания очереди запросов ввода-вывода. Средства прямого обращения к операционной системе. Программисты могут использовать этот уровень ввода-вывода для выполнения специальных функций, зависящих от конкретного устройства.

Запросы ввода-вывода могут обрабатываться косвенно, путем обращения к процедурам системы управления данными СУД-32, или непосредственно, путем обращения к программам системного обслуживания очереди запросов ввода-вывода. В конечном итоге, все операции ввода-вывода запрашиваются путем обращения к программам системного обслуживания очереди запросов ввода-вывода, т.к. если процесс использует процедуры системы управления данными СУД-32, то эта система формирует соответствующий запрос от имени процесса.

Управление вводом-выводом в МОС ВП осуществляют следующие компоненты:

- программы системного обслуживания, которые вызываются для запроса операции ввода-вывода;
- вспомогательные управляющие процессы, которые выпол-

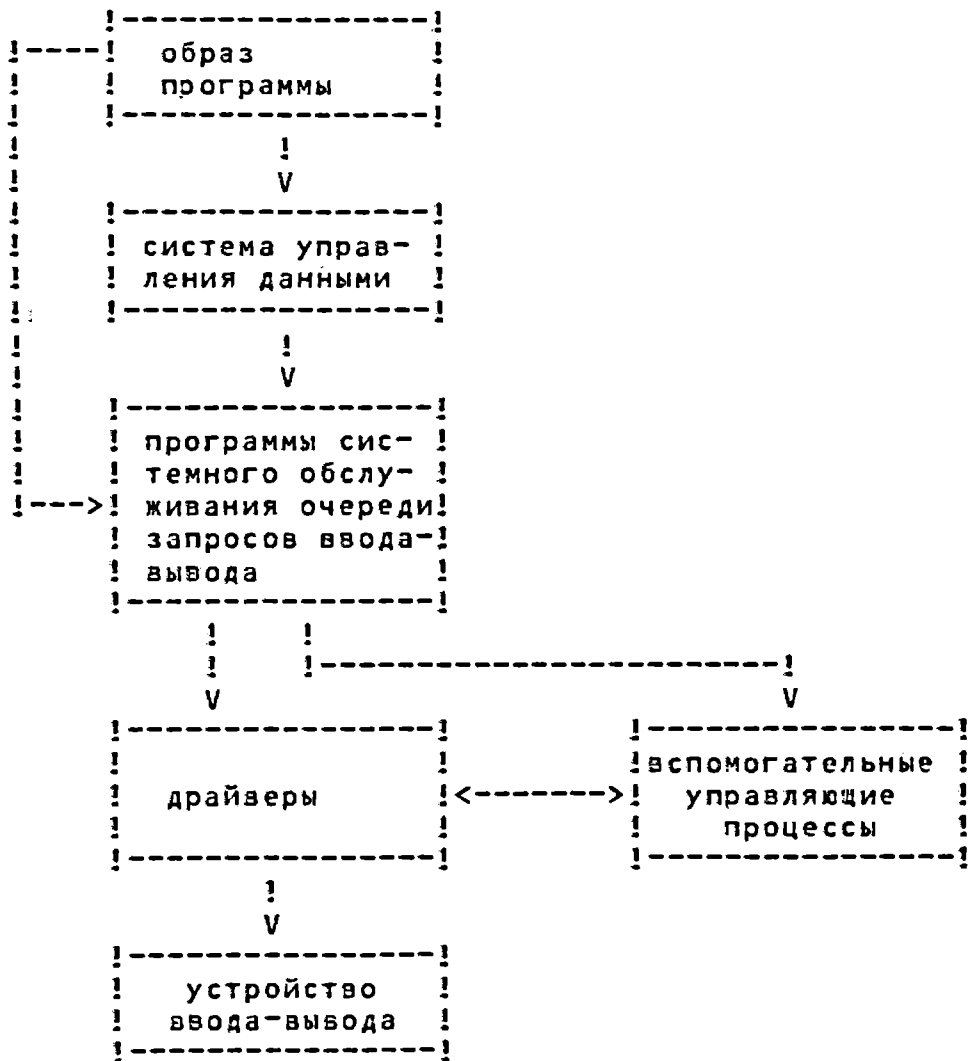
00152-01 31 01

няют обработку файло-ориентированных запросов ввода-вывода;

- драйверы устройств, которые управляют выполнением устройстве.

Система управления данными СУД-32 занимает более внешний и менее привилегированный уровень в структуре МОС ВП.

На рисунке представлена взаимосвязь между образом программы и компонентами ввода-вывода операционной системы.



3.5.1. Вспомогательные управляющие процессы

Управляющие процессы ввода-вывода, которые называются вспомогательными управляющими процессами, используются для обработки запросов ввода-вывода, ориентированных на файловую структуру. Каждый такой процесс обеспечивает обслуживание файловой структуры и управляет доступом к носителям для конкретных типов устройств. В МОС ВП имеется три вида вспомогательных управляющих процессов: для дисков, магнитных лент и сетей. Если устройства некоторого типа отсутствуют в системе, то отсутствует и вспомогательный управляющий процесс для данного типа устройств.

3.5.2. Драйверы устройств

Драйвер устройства управляет выполнением операций ввода-вывода непосредственно на внешнем устройстве, реализуя следующие функции:

- определяет периферийное устройство для системы;
- инициализирует устройство и/или его контроллер при загрузке системы и после сбоя питания;
- преобразует программные запросы на операции ввода-вывода в специфические для устройства команды;
- активизирует устройство;
- обрабатывает прерывания устройства;
- выдает информацию об ошибках устройства;
- передает данные и состояние от устройства программе, выдавшей запрос.

Драйверы устройств работают во взаимодействии с операционной системой МОС ВП. Операционная система выполняет всю обработку ввода-вывода, независимую от устройств. Для выполнения обработки ввода-вывода, специфической для конкретного устройства, операционная система передает управление драйверу этого устройства. Так как различные периферийные устройства выполняют различные команды и требуют различной настройки, операционная система должна поддерживать соответствующий драйвер для каждого типа устройства.

МОС ВП содержит драйверы для всех устройств, входящих в состав вычислительного комплекса ЭВМ СМ 1700. Кроме того, пользователь может разрабатывать и включать в систему дополнительные драйверы для нестандартных устройств.

3.6. Система управления данными СУД-32

Система управления данными СУД-32 (далее по тексту СУД-32), входящая в состав МОС ВП, осуществляет реализацию доступа к файлам данных, хранящимся на всех томах внешних запоминающих устройств и имеющих различные способы организации. (Под организацией файла понимают физическое расположение записей в файле). Организация файла и среда хранения определяют способы обработки данных.

СУД-32 обеспечивает эффективное и гибкое хранение данных, их выборку и корректировку. Программист, разрабатывающий прикладные программы обработки данных на языках программирования высокого уровня, определяет тип организации файла, форматы содержащихся в нем записей и методы доступа.

всю же реализацию физического размещения и выборку данных на носителях осуществляет интерфейс СУД-32, который используется в компиляторах соответствующих языков.

Программы на языке макроассемблер осуществляют связь с СУД-32 посредством размещенных в программе пользователя управляющих блоков (блок доступа к файлу - FAV и блок доступа к записи - RAB). Специальные макрокоманды СУД-32 осуществляют обработку файлов данных.

Создание или реорганизация (изменение способа организации) файла может также осуществляться при помощи обслуживающих программ СУД-32.

3.6.1. Организация файлов

Целью организации файла является упорядочение записей для обеспечения наиболее быстрой и гибкой их обработки. Организация файла определяется на этапе его создания; в дальнейшем ее нельзя изменить. Имеется, однако, обслуживающая программа копирования данного файла на другой том или на другую область того же тома с переписью содержимого старого файла в новый файл с отличной от первоначальной организацией. В СУД-32 доступны следующие три типа организации файла:

- 1) последовательная;
- 2) относительная;
- 3) индексная.

Записи в файле могут иметь один из следующих форматов:

- 1) записи фиксированной длины;
- 2) записи переменной длины;
- 3) записи переменной длины с постоянной управляющей частью.

Первые два формата могут использоваться в файлах с любой организацией. Третий формат допустим только в файлах с последовательной и относительной организацией.

3.6.1.1. Последовательная организация

При последовательной организации файла записи в нем упорядочены по принципу "одна за другой". Последовательная организация может быть реализована на всех типах устройств; более того, для всех устройств, исключая магнитные диски, такая организация является единственно допустимой. Записи расположены на носителе в том порядке, в котором они были записаны.

Добавление новых записей в файл с последовательной организацией может быть осуществлено только в его конец. Это объясняется тем, что физическое расположение каждой записи фиксировано относительно всех других записей файла. Следовательно, отсутствует неиспользованное пространство между записями, в которое можно было бы поместить новую запись.

3.6.1.2. Относительная организация

Файл с относительной организацией состоит из набора ячеек фиксированной длины, которые последовательно пронумерованы от 1 до N. Номер ячейки определяет ее расположение относительно начала файла.

Каждая ячейка может содержать единственную запись, однако необязательно, чтобы все ячейки содержали записи, обычно бывают и пустые ячейки. Поскольку номера ячеек являются уникальными, их используют для идентификации ячеек и содержащихся в них записей. Такой способ организации файла применим только для магнитных дисков.

3.6.1.3. Индексная организация

При индексной организации файла доступ к записям осуществляется по значению поля ключа. Размещением записей в индексном файле полностью управляет СУД-32. Поле ключа должно присутствовать в каждой записи. Расположение и длина ключа задаются при создании файла и должны быть одинаковыми во всех записях. Обычно в качестве поля ключа выбирается одно из информационных полей записи.

Для индексного файла должен быть определен по крайней мере один ключ, называемый первичным ключем. Кроме того, можно определить дополнительные, или альтернативные ключи. Альтернативные ключи, также как и первичный, могут быть использованы для идентификации записи при поиске. При записи в индексный файл СУД-32 использует значения первичного и

альтернативных ключей для построения индексов, представляющих собой древовидные структуры, а затем использует созданные индексы для доступа к записям. Индексная организация файлов предназначена только для данных, хранящихся на дисках.

3.6.2. Методы доступа к записям

Метод доступа к записи определяет способ занесения записи в файл и получения ее из файла. Для различных организаций файлов методы доступа также различны, что объясняется физическим упорядочением записей в файле. СУД-32 обеспечивает три метода доступа к записи:

- 1) последовательный;
- 2) произвольный по ключу или по относительному номеру записи;
- 3) произвольный по адресу записи в файле (RFA).

Для обработки записей в файле могут применяться различные методы доступа, кроме того можно изменять метод доступа в процессе работы с файлом. В табл.1 приводятся допустимые комбинации методов доступа к записи и организации файлов.

Таблица 1

Методы доступа к записи и организация файлов

Допустимый метод Доступа	Организация файла					
	Последовательная		Относительная		Индексная	
последовательный	да	да	да	да	да	да
	1)					
произвольный по относительному номеру записи	нет		да		нет	нет
произвольный по ключу	нет		нет		да	да
	2)					
произвольный по RFA	да		да		да	да

1) Произвольный доступ по относительному номеру записи для последовательных файлов допускается для записей фиксированной длины, расположенных на магнитных дисках.

2) Произвольный доступ по адресу записи в файле возможен только для магнитных дисков.

3.6.2.1. Последовательный метод доступа

Последовательный метод доступа к записям может быть использован для всех видов файлов OIД-32 и для всех видов устройств, ориентированных на запись, включая почтовый ящик. При таком методе доступа записи выбираются и заносятся в файл в порядке, установленном организацией файла.

3.6.2.2. Произвольный метод доступа по ключу или по относительному номеру записи

При произвольном доступе порядок обработки записей определяется пользовательской программой, а не организацией файла. Каждый запрос программы на обработку записи идентифицирует требуемую запись, указывая относительный номер записи: (для относительных файлов или для последовательных файлов, расположенных на диске и имеющих записи фиксированной длины) или значение ключа (для индексных файлов).

3.6.2.3. Произвольный метод доступа по адресу записи в файле

При любой организации файла записи могут идентифицироваться по адресу записи в файле (RFA), если файл расположен на диске. Этот метод является единственным методом произвольного доступа для последовательных файлов с записями переменной длины.

3.6.3. Обработка файлов и записей

Программы пользователя имеют возможность обрабатывать файлы СУД-32 на двух различных уровнях: на уровне файла и на уровне записи. На уровне файла обеспечивается выполнение операций над файлом как над единым целым. Имеется также возможность управлять доступом к файлу со стороны других пользователей и разделять файл, т.е. осуществлять его совместное использование несколькими пользователями.

На уровне файла программа может выполнять следующие операции:

- 1) создание файла;
- 2) открытие уже существующего файла;
- 3) изменение атрибутов файла;
- 4) расширение размера пространства, отведенного для файла;
- 5) закрытие файла;
- 6) удаление файла с диска.

На уровне записи СУД-32 обеспечивает выполнение следующих операций:

- 1) получение записи из файла;
- 2) занесение в файл;
- 3) поиск записи в файле;
- 4) замена содержимого записи в файле;
- 5) удаление записи из файла.

Программа пользователя может управлять операциями по обработке записей файла синхронно и асинхронно. При работе в синхронном режиме программа, выдавая запрос на выполне-

ние операции над записью, снова получает управление только после полного завершения обработки запроса. Большинство языков высокого уровня поддерживают только синхронные операции. В асинхронном режиме программа может получить управление до окончательного завершения операции над записью. Подробная информация о применении СУД-32 содержится в [1].

3.7. Программы системного обслуживания

Программы системного обслуживания предназначены для управления ресурсами, доступными процессам, обслуживания межпроцессного взаимодействия и выполнения основных функций МОС ВП, таких как координация операций ввода-вывода. Другими словами, программы системного обслуживания обеспечивают интерфейс с компонентами ядра операционной системы МОС ВП.

Программы системного обслуживания могут вызываться из программ, написанных на языке макроассемблер и языках программирования высокого уровня в соответствии со стандартными соглашениями о вызове процедур в операционной системе МОС ВП.

В целях обеспечения правильной работы системы и защиты целостности системы и процессов пользователя использование ряда программ системного обслуживания ограничено. Доступ к таким программам определяется наличием у процесса соответствующих привилегий и квот определенных системных ресурсов.

Программы системного обслуживания можно разделить на следующие функциональные группы:

- 1) программы системного обслуживания для управления процессами;
- 2) программы системного обслуживания для управления памятью;
- 3) программы системного обслуживания ввода-вывода;
- 4) программы системного обслуживания логических имен;
- 5) программы системного обслуживания флагов событий;
- 6) программы системного обслуживания асинхронных системных прерываний (AST);
- 7) программы системного обслуживания исключительных ситуаций;
- 8) программы системного обслуживания таймера и преобразования времени;
- 9) программы системного обслуживания для изменения режима.

3.7.1. Программы системного обслуживания для управления процессами

Программы системного обслуживания для управления процессами предоставляют пользователю возможность создавать, удалять процессы, управлять их выполнением (переход в состояние спячки, приостановка процесса, активизация и планирование активизации процесса, установка привилегий и приоритета), получать информацию о процессе или задании.

3.7.2. Программы системного обслуживания для управления памятью

Программы системного обслуживания для управления памятью пользователь может вызывать для непосредственного управления использованием виртуальной памяти в целях повышения эффективности работы программ.

Программы системного обслуживания для управления памятью позволяют:

- управлять размером рабочего набора процесса;
- увеличивать или уменьшать область программы и область управления;
- создавать и удалять виртуальное адресное пространство;
- блокировать и разблокировать страницы в рабочем наборе и физической памяти;
- создавать и отображать секцию (т.е. файл на диске, содержащий данные или разделяемые образы программ);
- устанавливать защиту страниц.

3.7.3. Программы системного обслуживания ввода-вывода

Программы системного обслуживания ввода-вывода разрешают пользователю непосредственно управлять ресурсами ввода-вывода МОС ВП, обеспечивая выполнение следующих функций:

- назначение и отмена назначения канала для устройства

ввода-вывода;

- захват устройства и освобождение распределенного устройства;
- обслуживание очереди запросов ввода-вывода;
- отмена запросов ввода-вывода в канале;
- форматирование выводных строк с преобразованием цифровых данных в строки кода КОИ-8;
- создание и удаление почтовых ящиков;
- установка сообщений в очередь к системным процессам.

3.7.4. Программы системного обслуживания логических имен

Программы системного обслуживания логических имен предоставляют средства для поддержки и доступа к символьной строке, называемой логическим именем, а также для установления эквивалентности пары имен. Логические имена удобно использовать для обозначения устройств ввода-вывода и файлов, что позволяет создавать программы, не зависящие от конкретного типа устройства и имени файла.

Программист имеет возможность создать, удалить и преобразовать логическое имя, а также создать таблицу логических имен.

3.7.5. Программы системного обслуживания флагов событий

Флаги событий используются процессом для синхронизации последовательности операций в программе. Кроме того, флаги событий могут использоваться группой процессов для обеспечения взаимной синхронизации. Программы системного обслуживания флагов событий позволяют устанавливать, сбрасывать, считывать флаг события, помещать процесс в состояние ожидания установки флага события, создавать и удалять кластер общих флагов событий, обеспечивать установку и отмену связи с таким кластером.

3.7.6. Программы системного обслуживания асинхронных системных прерываний (AST)

Выполнение процесса может прерываться при возникновении некоторых событий (таких, как завершение операции ввода-вывода). Такие прерывания происходят асинхронно по отношению к выполнению процесса, отсюда и их название - асинхронные системные прерывания.

Программы системного обслуживания асинхронных системных прерываний могут разрешить или запретить AST, объявить AST, установить AST по восстановлению после сбоя питания.

3.7.7. Программы системного обслуживания исключительных ситуаций

Исключительная ситуация - это событие, которое происходит синхронно с выполнением инструкции и прерывает выполнение образа, например, арифметическое переполнение, деление на нуль и т.д. При возникновении исключительной ситуации управление получает соответствующая программа обработки. Пользователь может определять собственные программы обработки, в противном случае, управление получает системный обработчик исключительных ситуаций.

Программы системного обслуживания исключительных ситуаций позволяют определять программы обработки для специальных случаев (например, программу, которая получит управление при изменении режима доступа).

3.7.8. Программы системного обслуживания таймера и преобразования времени

Программы системного обслуживания таймера и преобразования времени используются для планирования программных событий на конкретное время дня или по истечении заданного интервала времени и выполняют следующие действия:

- выдать системное время;
- преобразовать время (из системного формата в числовой; из системного формата в строку символов и наоборот);
- установить таймер;

- отменить запрос таймера;
- запланировать активизацию процесса;
- установить системное время.

3.7.9. Программы системного обслуживания для изменения режима

Программы системного обслуживания для изменения режима позволяют процессу изменить режим доступа на режим управляющей программы или режим ядра, а также изменять указатель стека для менее привилегированного режима.

Подробное описание программ системного обслуживания приведено в [2].

3.8. Командные интерфейсы

Пользователь обращается к операционной системе МОС ВП при помощи командного языка. С системой поставляются диалоговый командный язык (DCL), который является основным средством связи пользователя с МОС ВП, и программа связи с оператором (MCR), предназначенная для обеспечения совместимости с операционной системой ОС РВ.

Командный язык является интерактивным, понятным, легким для использования и крайне гибким. Он дает возможность пользователю регистрироваться в системе, работать с файлами, разрабатывать и тестировать программы, а также получать информацию о системе. Кроме того, он разрешает пользователю расширять и переопределять свой набор команд и легко писать

командные процедуры.

Командный язык прост в эксплуатации, так как его команды могут быть сокращены, существует подсказка для необходимых аргументов и применяются стандартные или выбранные пользователем значения по умолчанию для параметров и квалификаторов команд.

Командная строка обычно состоит из имени команды, за которым следует один или несколько параметров, которые определяют объект операции, или как операция должна быть выполнена. Если пользователь вводит с терминала неполную команду, интерпретатор команд подсказывает ему необходимые параметры.

Интерпретатор команд высвечивает на экране знак денежной единицы (**¤**) для указания готовности приема команды и знак денежной единицы с последующим знаком подчеркивания (**_**) для указания соответствующего параметра.

В основном командный язык используется программистами при разработке программного обеспечения, операторами - для управления системой, администраторами - для назначения привилегий пользователям.

Разработчик прикладных программ может предусмотреть собственный командный интерфейс для пользователей системы.

Для того, чтобы не вводить несколько раз повторяющуюся последовательность команд, пользователи могут создавать командные процедуры. Командная процедура представляет собой файл, содержащий полные командные строки (включая знак **¤**). Пользователь может выдавать запросы интерпретатору

команд для чтения и обработки командных строк из файла процедуры точно также, как если бы они были последовательно набраны на терминале. Для выполнения командной процедуры пользователь просто задает имя файла командной процедуры, перед которым стоит знак "а".

Также пользователь может создать символ для выполнения командного файла. Пользователь может приравнять уникальную последовательность символов файлу командной процедуры.

Имеются команды для остановки и продолжения выполнения пакетных заданий, передачи управления и указания обработки ошибок.

3.8.1. Перечень команд диалогового командного языка (DCL)

3.8.1.1. Информационные и управляющие команды

ASSIGN

- назначает логическое имя данной строке символов (эквивалентное имя) и сохраняет эту пару имен в групповой или системной таблице логических имен процесса. Обычно используется для создания логического имени устройства.

DEASSIGN

- разрушает соответствие между логическим именем и его эквивалентным именем (смотри команды ASSIGN и ALLOCATE) или уничтожает символ (смотри команду

00152-01 31 01

DEFINE).

DEFINE

- создает эквивалентное имя для данного логического имени. Аналогична команде ASSIGN за исключением синтаксиса.

HELP

- выводит информацию, помогающую пользователю в выборе нужного квалификатора команды.

LOGIN

- пользователь начинает интерактивный сеанс работы с системой, нажимая УС/У, УС/У или возврат каретки, если используется терминал с твердой копией. Затем система напоминает, что надо ввести имя пользователя и пароль, и проверяет их достоверность.

LOGOUT

- завершает интерактивный сеанс и освобождает все ресурсы, распределенные пользователю.

MCR

- означает, что данная или следующая командная строка будет интерпретироваться командным интерпретатором ОС РВ.

REQUEST

- выводит сообщение на терминал оператора и дополнительно запрашивает ответ.

SET

- указывает, получает ли интерпретатор команд управление при введении УС/У; изменяет принятые по умолчанию значения имени устройства и каталога пользователя. Определяет характеристики по умолчанию для устройства на магнитной ленте; определяет, будет ли интерпретатор команд обрабатывать ошибки при выполнении команд в командных процедурах; изменяет характеристики выполнения текущего процесса; изменяет защиту файла; изменяет текущее состояние или атрибуты файла, помещенного в очередь на печать или на пакетную обработку; изменяет характеристики терминала; указывает, будут или нет командные строки выводиться на терминал или печататься в протоколе пакетного задания; переопределяет значение по умолчанию размера рабочего набора для текущего процесса.

SHOW

- выводит любой из следующих видов информации: текущий день, дату, время, текущее значение по умолчанию для имени устройства и каталога, состояние устройств в системе; назначенные устройствам логические имена, текущие характеристики и состояние накопителя на магнитной ленте; имя, номер и состояние локального узла сети и список доступных удаленных узлов; характеристики, принятые по умолчанию для системного печатающего устройства; состояние текущего процесса: защита текущего файла, назначаемая всем новым фай-

лам, создаваемым во время терминального сеанса или пакетным заданием; текущее состояние входов очередей заданий на пакетное выполнение и печать; текущая дисковая квота; состояние образа, выполняющегося в текущий момент; текущее значение локальных и глобальных символов; список и информацию о процессах в системе; текущие характеристики терминала; результаты трансляции логических имен; квота и лимит для рабочего набора текущего процесса.

Примечание. УС/Х означает одновременное нажатие клавиши "УС" и клавиши "Х", где Х, например, Y, S, C.

3.8.1.2. Команды управления пакетом и командными процедурами

DECK

- отмечает начало входного потока для команды или программы.

␣EOD

- означает конец данных входного потока, следующих за командой ␣DATA.

␣EOJ

- отмечает конец пакетного задания, введенного с устройства считывания перфокарт. Выполняет такие же функции, как команда LOGOUT.

EXIT

- заканчивает командную процедуру.

#GOTO

- передает управление на строку с указанной меткой.

#IF

- передает управление на строку с указанной меткой, если результат логического сравнения символических переменных истина.

#INQUIRE

- требует интерактивного присвоения значения символному имени.

#JOB

- отмечает начало пакетного командного файла и предусматривает учетную информацию (такую как лимит времени).

#ON

- передает управление на строку с указанной меткой, если ошибка данного или более высокого уровня возникла при выполнении командной процедуры.

#PASSWORD

- определяет пароль, связанный с именем пользователя, указанным в карте JOB пакетного задания.

#SUBMIT

- помещает данный пакетный командный файл или командную процедуру в очередь на пакетную обработку.

Примечание. Имена команд с предшествующим знаком "#"

используются только в пакетном командном файле или командной процедуре. Все остальные команды могут быть введены интерактивно.

3.8.1.3. Команды управления томами и устройствами

ALLOCATE

- резервирует устройство для монопольного использования и разрешает пользователю назначить логическое имя устройству.

DEALLOCATE

- освобождает зарезервированное устройство.

DISMOUNT

- разрушает логическую связь устройства с заданием пользователя.

INITIALIZE

- записывает файл каталога и другую информацию о структуре тома на диск или магнитную ленту для подготовки их к работе.

MOUNT

- делает устройство доступным для пользователя и дополнительно связывает логическое имя с томом или набором томов.

3.8.1.4. Команды для работы с файлами

ANALYZE/OBJECT

- обеспечивает описание содержимого объектного файла или выполнимого файла образа.

APPEND

- объединяет содержимое последовательных файлов в данный или вновь созданный файл.

CLOSE

- закрывает файл, открытый для ввода или вывода командой OPEN, и освобождает логическое имя, определенное при создании файла.

COPY

- копирует содержимое файла или файлов, создавая другой файл или файлы.

CREATE

- создает новый файл из данных, последовательно вводимых во входном потоке (с терминала или в пакете). Создает файл каталога на томе.

DELETE

- удаляет один или несколько файлов из массовой памяти на дисковом томе.

DELETE/ENTRY

- удаляет одно или несколько заданий из очереди на печать.

DELETE/SYMBOL

- удаляет один или несколько символов, находящихся в таблице локальных или глобальных символов.

DIFFERENCE

- сравнивает два файла и выдает различия между ними.

DIRECTORY

- выдает информацию (размер, защита, время создания, и т.д.) о файле или группе файлов.

DUMP

- выполняет вывод на печать содержимого файла, игнорируя любые символы управления форматом, которые могут встретиться в записях.

EDIT

- открывает текстовый файл и принимает команды для вставки, удаления или модификации данных в файле.

OPEN

- открывает файл для чтения или записи на командном уровне.

PRINT

- посылает содержимое некоторого файла или файлов на ацпу.

PURGE

- удаляет все, кроме последней, версии для данного файла или файлов, дополнительно сохраняет две или несколько последних версий.

READ

- считывает отдельную запись из указанного файла.

RENAME

- изменяет имена одного или нескольких существующих файлов.

SORT

- создает файл, переупорядочивая записи в данном файле по содержанию ключевых полей внутри записей.

TYPE

- выводит содержимое некоторого файла или файлов на дисплей с логическим именем SYS=OUTPUT: по умолчанию - это обычно терминал пользователя.

UNLOCK

- разрешает доступ к файлу, который был неправильно закрыт.

WRITE

- заносит запись в указанный файл.

3.8.1.5. Команды для разработки программ и управления их выполнением

BASIC

- транслирует исходные модули на языке БЕЙСИК, создает объектный модуль.

BLISS

- вызывает транслятор языка программирования БЛИСС.

CANCEL

- отменяет запросы на активизацию для указанного процесса.

COBOL

- транслирует указанные исходные модули, написанные на языке программирования КОБОЛ.

CONTINUE

- продолжает выполнение программы, прерванное УС/С или УС/У.

CORAL

- вызывает транслятор языка программирования КОРАЛ.

DEBUG

- начинает сеанс интерактивной отладки после прерывания выполнения программного образа с помощью ввода УС/С или УС/У.

DEPOSIT

- заменяет содержимое ячейки виртуальной памяти указанными данными.

EXAMINE

- отображает содержимое ячейки виртуальной памяти.

FORTRAN

- вызывает транслятор языка программирования ФОРТРАН для трансляции одной или нескольких исходных программ.

LIBRARY

- создает, удаляет или сохраняет библиотеки объектных модулей, разделяемых образов или исходных модулей макрокоманд.

LINK

- компоует модули для создания образов.

MACRO

- ассемблирует указанные исходные модули на языке МАКРОАССЕМБЛЕР, создает объектный модуль.

PL/1

- вызывает транслятор языка программирования ПЛ/1 для трансляции одной или нескольких исходных программ.

PASCAL

- вызывает транслятор языка программирования ПАСКАЛЬ для трансляции одной или нескольких исходных программ.

RUN

- выполняет программный образ в контексте текущего процесса или создает отделенный процесс и выполняет образ в контексте этого процесса.

STOP

- завершает программу после ввода УС/С или УС/У.

SYNCHRONIZE

- помещает процесс, выполняющий командную процедуру, в состояние ожидания до завершения выполнения указанного пакетного задания.

WAIT

- помещает текущий процесс в состояние ожидания до истечения указанного периода времени.

Подробное описание диалогового командного языка содержится в [3].

3.9. Режим совместимости

Режим совместимости позволяет обеспечить выполнение на ВК СМ 1700 большинства программ пользователя, написанных для 16-разрядных СМ ЭВМ с магистральной архитектурой. Предполагается, что все программное обеспечение в режиме совместимости использует средства системного обслуживания мос вп для обработки ввода-вывода, обработки прерываний, исключительных ситуаций и управления памятью. Процессор СМ 1700 реализует базовый набор инструкций 16-разрядных СМ ЭВМ, и практически все непривилегированные программы могут исполняться на ВК СМ 1700 без изменений. Существуют некоторые ограничения на условия выполнения, которые МОС ВП может предоставить программам 16-разрядных СМ ЭВМ.

Существует однозначное соответствие между виртуальными адресами в режиме совместимости и первыми 64 Кбайтами виртуального адресного пространства, доступного процессам в основном режиме. Так же как и в 16-разрядных СМ ЭВМ, программа в режиме совместимости ограничена для обращения только этими адресами. МОС ВП обеспечивает большинство из механизмов управления памятью 16-разрядных СМ ЭВМ. Например, режим совместимости автоматически поддерживает защиту сег-

ментов памяти, но для 512-байтных сегментов, а не для 64-байтных.

Все общие регистры и режимы адресации 16-разрядных СМ ЭВМ доступны в режиме совместимости. Регистры от R0 до R6 в режиме совместимости являются младшими 16-ю битами регистров от R0 до R6 основного режима. Регистр R7 (счетчик команд) в режиме совместимости является младшими битами регистра R15 основного режима. Регистры от R8 до R14 основного режима не используются в режиме совместимости. Заметим, что регистр R6 режима совместимости действует как указатель стека для временного хранения локальных данных программы, но этот локальный стек программы размещается в области программы, а не в области управления.

Младшие 16-битов длинного слова состояния процессора СМ 1700 (PSL) эквивалентны слову состояния процессора для 16-разрядных СМ ЭВМ.

Все прерывания и исключительные ситуации, которые происходят, когда процессор работает в режиме совместимости, заставляют процессор перейти в основной режим.

Набор команд режима совместимости такой же, как и для 16-разрядных СМ ЭВМ со следующими исключениями:

- привилегированные команды (HALT, WAIT, RESET, SPL и MARK) являются недопустимыми;
- команды прерываний (BPT, IOT, EMT и TRAP) заставляют процессор перейти в основной режим, где или прерывание может быть обслужено, или команда имитирована;
- команды MEPI, MTRI, MFPD, MTPD выполняются на ЭВМ СМ

00152-01 31 01

1700 точно так же как они выполнялись бы на 16-разрядных СМ ЭВМ в режиме пользователя. Эти команды игнорируются на уровне предварительного доступа и действуют как команды PUSH и POPR, обращающиеся к текущему стеку;

- команды с плавающей запятой эмулируются через программное обеспечение.

Процессор СМ 1700 может выполнять поток инструкций 16-разрядных СМ ЭВМ режима пользователя в контексте процесса. МОС ВП заменяет обращения к системным директивам ОС РВ обращениями к функционально эквивалентным программам системного обслуживания.

МОС ВП поддерживает сервисные процедуры системы управления данными для программ режима совместимости. Файлы программ и данных могут, следовательно, передаваться между операционными системами МОС ВП и ОС РВ.

МОС ВП также поддерживает команды программы связи с оператором (MCR) ОС РВ, которые или вводятся непосредственно с терминала, или представлены как косвенные командные файлы.

Для выполнения в режиме совместимости образ задачи должен удовлетворять следующим требованиям:

- образ не должен выполнять привилегированные инструкции 16-разрядных СМ ЭВМ;
- образ не должен зависеть от 32-разрядной организации памяти;
- образ не должен непосредственно управлять

вводом-выводом на физическом уровне;

- образ не должен использовать средства ОС РВ, недоступные в МОС ВП.
- образ не должен использовать сети.

МОС ВП представляет средства разработки программ (например, макроассемблер и построитель задач), модифицирующие программы для выполнения в режиме совместимости.

Для выполнения большинства директив ОС РВ в основном режиме выполняются функционально эквивалентные программы системного обслуживания. В большинстве случаев программы системного обслуживания дублируют функции. Например:

- директивы разрешения/блокировки контрольной точки интерпретируются как вызов программы системного обслуживания для установки режима обмена процессов;
- директива передать/принять преобразуется в вызов программ системного обслуживания для чтения/записи информации почтового ящика. Образы основного режима и режима совместимости могут совместно использовать почтовые ящики;
- директивы флагов событий в большинстве идентичны. Образы основного режима и режима совместимости могут совместно использовать флаги общих событий при условии, что они принадлежат одной группе;
- директива назначения логического номера устройства интерпретируется как назначение канала для соответствующего устройства.

В некоторых случаях МОС ВП не может имитировать функ-

цию, но она делает, что возможно для продолжения работы программы. Например:

- образу задачи разрешается объявить существенное событие, но директива игнорируется;
- директива установки приоритета игнорируется, поскольку пределы приоритетов различны. Для работы в заданном приоритете образ должен быть запущен в контексте процесса, задающего этот приоритет.

В большинстве случаев характеристики программных сред ОС РВ и МОС ВП соответствуют друг другу. Например, задачи могут приостанавливаться, принимать АСТ и запросы на планирование активизации. Подпрограммы АСТ могут быть объявлены как программы обработки трассировки прерываний, прерываний по неверным инструкциям и т.д. [4].

3.10. Наборы инструкций

Инструкции могут выполняться процессором в одном из двух режимов: основном и в режиме совместимости. В основном режиме процессор исполняет многочисленные инструкции ВК СМ 1700, распознает различные типы данных, использует шестнадцать 32-разрядных общих регистров. В режиме совместимости процессор выполняет непривилегированные инструкции 16-ти разрядных СМ ЭВМ, распознает целые данные и использует восемь 16-разрядных общих регистров. Процесс пользователя может выполнять образы обоих режимов.

Набор инструкций основного режима содержит 244 инструкции. Каждая инструкция состоит из кода операции, за

которым следуют указатели операндов (от 0 до 6), количество и тип которых зависит от кода операции. Каждый операнд может адресоваться одним из нескольких способов.

В инструкциях основного режима могут быть использованы 16 32-разрядных общих регистров.

Операнды инструкций часто хранятся в общих регистрах или доступ к этим операндам происходит через общие регистры. Шестнадцать 32-разрядных программируемых общих регистров обозначают как R0-R15 (в десятичной системе). Регистры могут быть использованы для временного хранения, для накопления, как базовые или индексные регистры. Базовый регистр содержит адрес базы структуры данных программного обеспечения, например, таблицы, а индексный регистр содержит логическое смещение в структуре данных.

В регистре данные хранятся в том же формате, как и в памяти. Если квадрослово или величина с плавающей запятой с двойной точностью хранятся в регистре, то в действительности они хранятся в двух соседних регистрах.

Некоторые регистры имеют специальное назначение в зависимости от выполняемой инструкции. Регистры R12-R15 имеют специальное назначение для многих инструкций и поэтому имеют специальное обозначение. Регистрами специального назначения являются:

- счетчик команд (PC, или R15), содержащий адрес следующего байта в потоке инструкций;
- указатель стека (SP, или R14), который содержит адрес вершины стека, используемый для вызова подп-

рограмм или процедур;

- указатель фрагмента (FP, или R13), содержащий адрес базы структуры данных программного обеспечения, хранящейся во фрагменте списка. Используется для вызова процедур;
- указатель аргумента (AP, или R12), который содержит адрес базы структуры данных программного обеспечения, называемой списком аргументов. Предназначен для вызова процедур.

Несмотря на большое число инструкций, их изучение не представляет особых трудностей. Многие инструкции соответствуют непосредственно операторам языка программирования высокого уровня, а мнемоника языка макроассемблер ассоциируется с функциями инструкций. Для выбора подходящей инструкции необходимо знать операции, типы данных и режимы адресации.

3.10.1. Типы данных

Тип данных определяет количество байтов памяти, к которым производится обращение как к единому элементу и которые интерпретируются как единый элемент. Инструкции основного режима процессора распознают следующие типы данных:

- 1) целое;
- 2) число с плавающей запятой;
- 3) десятичное упакованного формата;

00152-01 31 01

- 4) символьная строка;
- 5) битовое поле переменной длины;
- 6) числовая строка;
- 7) очередь.

В табл.2 содержится краткое описание возможных типов данных.

Таблица 2

Тип данного	Размер	Диапазон значений (десятичный)
целое	! байт	! со знаком от -128 до +127
	! (8 битов)	! без знака от 0 до 255
	! слово	! со знаком от -32768 до +32767
	! (16 битов)	! без знака от 0 до 65535
	! длинное слово	! со знаком от -2^{31} до $+2^{31}-1$
	! (32 бита)	! без знака от 0 до $2^{32}-1$
	! квадрослово	! со знаком от -2^{63} до $+2^{63}-1$
	! (64 бита)	! без знака от 0 до $2^{64}-1$
	! октаслово	! со знаком от -2^{127}
	! (128 битов)	! до $+2^{127}-1$
	! без знака от 0 до $2^{128}-1$	
число с плавающей запятой F_формата	! 32 бита	! от $2.9 \cdot 10^{(-38)}$
		! до $1.7 \cdot 10^{38}$
		! точность 7 десятичных цифр
		!
число с плавающей запятой	! 64 бита	! от $2.9 \cdot 10^{(-38)}$

Продолжение табл. 2

Тип данного	Размер	Диапазон значений (десятичный)
вещей запятой		до $1.7 \cdot 10^{38}$
D_формата		точность 16 десятичных цифр
число с плавающей запятой	64 бита	от $0.56 \cdot 10^{-308}$
G_формата		до $0.9 \cdot 10^{308}$
число с плавающей запятой	128 битов	от $0.84 \cdot 10^{-4932}$
H_формата		до $0.59 \cdot 10^{4932}$
		точность 33 десятичных цифр
упакованная десятичная строка	от 0 до 16 байтов (31 цифра)	число; в байте содержится 2 цифры; знак в младшей половине последнего байта
символьная строка	от 0 до 65535 байтов	один символ на байт
битовое поле переменной длины	от 0 до 32 битов	зависит от интерпретации

Продолжение табл. 2

Тип данного	Размер	Диапазон значений (десятичный)
числовая	от 0 до 31	от $-10^{*31}-1$ до $+10^{*31}-1$
строка	байта	
очередь	≥ 2 длинных слова (элемент очереди)	элементы очереди произвольно смещены в памяти

Целые данные хранятся как двоичная величина в целом байте, слове, длинном слове, а в некоторых случаях в квадрослове или октаслове. Байт - это 8 битов, слово - 2 байта, длинное слово - 4 байта, квадрослово - 8 байтов, октаслово - 16 байтов.

Процессор может интерпретировать целое число как значение со знаком или значение без знака. Знак определяется старшим по порядку битом.

Числа с плавающей запятой хранятся, используя порядок со знаком и двоичную нормализованную мантиссу. Нормализованный бит не отображается. Предусмотрены 4 типа формата данных с плавающей запятой. В дополнение к двум стандартным форматам чисел с плавающей запятой, используемым в 16-разрядных СМ ЭВМ (F₁ и D₁), в ЭВМ СМ 1700 используется 2 расширенных формата (G₁ и H₁).

Данные десятичного упакованного формата хранятся в цепочке байтов. Каждый байт делится на два 4-битных поля. В каждом поле хранится по одной десятичной цифре. Первая, или старшая по порядку цифра хранится в старшем поле первого байта, вторая цифра - в младшем поле первого байта, третья цифра - в старшем поле второго байта и т.д. Знак числа хранится в младшем поле последнего байта.

Символьные данные - это последовательность байтов, содержащих любые двоичные данные, например, коды КОИ-8. Первый символ строки хранится в первом байте, второй символ - во втором байте и т.д. Строка символов, содержащая коды КОИ-8 для десятичных цифр, называется числовой строкой.

Адрес любого элемента данных - это адрес первого байта, начиная с которого располагается данный элемент.

Целые, десятичные упакованные, символьные и данные с плавающей запятой могут быть сохранены в памяти, начиная с границы произвольного байта. Битовое поле не обязательно должно начинаться с границы байта. Это последовательность смежных битов (0-32), начало которой определяется относительно адреса данного байта или регистра. Инструкции основного режима могут интерпретировать битовое поле как целое со знаком или без знака.

Данные типа "очередь" состоят из циклических двухсвязных списков. Элемент очереди определяется его адресом. Каждый элемент очереди связывается с соседними элементами через два длинных слова. Первое длинное слово указывает на следующий элемент очереди, а второе длинное слово - на пре-

дыдущий элемент очереди. Процессор CM 1700 поддерживает два типа очередей в соответствии с типом связи: абсолютные и относительные.

3.10.2. Режимы адресации

Операнд инструкции может быть размещен в основной памяти, в общем регистре или в потоке инструкций. Способ, с помощью которого определяется местоположение операнда, называется режимом адресации операнда.

Существует 7 основных режимов адресации, которые используют общие регистры для идентификации размещения операнда. Эти режимы включают:

1) регистровый режим, при котором регистр содержит операнд;

2) косвенно-регистровый режим, при котором регистр содержит адрес операнда;

3) режим с автоуменьшением, при котором содержимое регистра сначала уменьшается на размер операнда, а затем используется как адрес операнда. Размер операнда (в байтах) определяется типом данных и зависит от инструкции;

4) режим с автоувеличением, при котором содержимое регистра используется как адрес операнда, а затем увеличивается на размер операнда. Если используется РС, то режим называется непосредственным;

5) косвенный режим с автоувеличением, при котором содержимое регистра используется как адрес длинного слова в памяти, содержащего адрес операнда, и затем увеличивается

на 4 (размер представления адреса в байтах). Если используется регистр РС, то режим называется абсолютным;

6) режим со смещением, при котором значение, хранимое в регистре, используется как базовый адрес. Байт(в), слово(W) или длинное слово(L) определяют константу, которая добавляется к базовому адресу, а их сумма является адресом операнда;

7) косвенный режим со смещением, при котором значение, хранимое в регистре, используется как базовый адрес таблицы адресов. Байт, слово или длинное слово определяют константу, которая добавляется к базовому адресу для получения адреса, по которому находится адрес операнда.

Эти основные семь режимов, исключая регистровый режим, могут быть модифицированы с помощью индексного регистра. Когда индексный регистр используется с основным режимом адресации для идентификации операнда, режим адресации называется основным режимом с приставкой индексный. Кроме 7 основных режимов адресации, которые используют регистры, процессор различает еще 6 индексных режимов адресации. В индексном режиме адресации один регистр используется для вычисления базового адреса структуры данных, а другой - для вычисления индексного смещения в структуре данных.

Процессор обеспечивает также литеральный режим адресации, при котором 6-битовое поле инструкции интерпретируется как целая константа или константа с плавающей запятой.

Разнообразные режимы адресации дают возможность программисту на языке макроассемблера писать, а трансляторам

языков программирования высокого уровня создавать очень компактные программы.

Режимы адресации приведены в табл.3.

Таблица 3

Режим адресации	Синтаксис макроассемблера	Индексация
литеральный (непосредственный)	S# константа I# константа	нет
регистровый	1) RN	нет
косвенно- регистровый	(RN)	да
с автоуменьшением	-(RN)	да
с автоувеличением	(RN)+	да
косвенный с автоувеличением (абсолютный)	@(RN)+ @#адрес	да
со смещением	B# адрес смещения (RN) W# адрес смещения (RN) L# адрес смещения (RN)	да

Продолжение табл. 3

Режим адресации	!	Синтаксис макроассемблера	!	Индексация
косвенный	!	а В# адрес смещения (RN)	!	да
со смещением	!	а W# адрес смещения (RN)	!	
	!	а L# адрес смещения (RN)	!	

1) N - принимает значения от 0 до 15.

3.10.3. Инструкции основного режима

Набор инструкций основного режима базируется более чем на 200 различных кодах операций. Коды операций могут быть сгруппированы в классы на основе их функций и использования.

Инструкции, используемые для обработки основных типов данных, включают:

- логические и целочисленные инструкции;
- инструкции для данных с плавающей запятой;
- инструкции для десятичных чисел в упакованном формате;
- инструкции для символьных строк;
- инструкции для битовых полей;
- инструкции для работы с очередями.

Инструкции, используемые для работы со специальными

видами данных, включают:

- инструкции для работы с адресами;
- инструкции для работы с общими регистрами;
- инструкции для работы с индексами.

Инструкции для управления работой программ и вызова процедур:

- инструкции перехода;
- инструкции вызова подпрограмм;
- инструкции вызова процедур.

Табл.4 содержит список основных инструкций ЭВМ СМ 1700 в порядке их классификации.

Таблица 4

Основные инструкции СМ 1700

Обозначение!	Содержание
	! инструкции для работы с целыми и данными с плавающей запятой
1) !	2)
MOV_	! переслать (B,W,L,F,D,G,H,Q,O)
MNEG_	! переслать с отрицанием (B,W,L,F,D,G,H)
MCOM_	! переслать в дополнительном коде (B,W,L)
MOVZ_	! переслать с расширением нулями (BW,BL,WL)
CLR_	! очистить (B,W,L=F,Q=D=G,O=H)
CVT_	! преобразовать формат (B,W,L,F,D,G,H) в (B,W,L,F,D,G,H) кроме (BB,WW,LL,FF,DD,GG,HH,DG,GD)
CVTR_L	! преобразовать с округлением (F,D,G,H) в

Обозначение!	Содержание
	! длинное слово
CMP_	! сравнить (B,W,L,F,D,G,H)
TST_	! проверить (B,W,L,F,D,G,H)
BIS_2	! установить бит (B,W,L); 2 операнда
BIS_3	! установить бит (B,W,L); 3 операнда
BIC_2	! очистить бит (B,W,L); 2 операнда
BIC_3	! очистить бит (B,W,L); 3 операнда
BIT_	! проверить бит (B,W,L)
XOR_2	! исключающее или (B,W,L); 2 операнда
XOR_3	! исключающее или (B,W,L); 3 операнда
ROTL	! циклический сдвиг длинного слова
	! арифметические инструкции для работы с целыми
	! и данными с плавающей запятой
	!
INC_	! увеличить (B,W,L) на 1
DEC_	! уменьшить (B,W,L) на 1
ASH_	! арифметический сдвиг (L,Q)
ADD_2	! сложить (B,W,L,F,D,G,H); 2 операнда
ADD_3	! сложить (B,W,L,F,D,G,H); 3 операнда
ADWC	! сложить с переносом
ADAWI	! сложить выравненные слова с блокировкой
SUB_2	! вычесть (B,W,L,F,D,G,H); 2 операнда

Обозначение!	Содержание
SUB_3	! вычесть (B,W,L,F,D,G,H); 3 операнда
SBWC	! вычесть с переносом
MUL_2	! умножить (B,W,L,F,D,G,H); 2 операнда
MUL_3	! умножить (B,W,L,F,D,G,H); 3 операнда
EMUL	! расширенное умножение
DIV_2	! разделить (B,W,L,F,D,G,H); 2 операнда
DIV_3	! разделить (B,W,L,F,D,G,H); 3 операнда
EDIV	! расширенное деление
EMOD_	! расширенное умножение с выделением целой ! части (F,D,G,H)
POLY_	! вычислить значение многочлена (F,D,G,H)
	! инструкции для работы с упакованными ! десятичными данными !
MOVDP	! переслать десятичные упакованные данные
CMPP3	! сравнить; 3 операнда
CMPP4	! сравнить; 4 операнда
ASHP	! арифметический сдвиг
ADDP4	! сложить; 4 операнда
ADDP6	! сложить; 6 операндов
SUBP4	! вычесть; 4 операнда
SUBP6	! вычесть; 6 операндов

Обозначение!	Содержание
MULP	! умножить
DIVP	! разделить
CVTLP	! преобразовать длинное слово в упакованный ! формат
CVTPL	! преобразовать упакованные данные в ! длинное слово
CVTTP	! преобразовать формат с остатком в ! упакованный формат
CVTPS	! преобразовать упакованный формат в формат ! с ведущими разделителями
CVTSP	! преобразовать формат с ведущими ! разделителями в упакованный формат
EDITPC	! редактирование данных упакованного формата ! для получения символьной строки
	! инструкции для работы с символьными ! строками !
MOVCS	! переслать символ; 3 операнда
MOVCS	! переслать символ; 5 операндов
MOVTC	! переслать символ с трансляцией
MOVTC	! переслать с трансляцией и выходом ! по символу

Обозначение!	Содержание
CMPC3	! сравнить симаолы; 3 операнда
CMPC5	! сравнить символы; 5 операндов
LOCC	! найти символ
SKPC	! пропустить символ
SCANC	! посмотреть символы
SPANC	! найти символ данного типа
MATCHC	! сравнить символы на совпадение
	! инструкции для работы с битовыми полями
	! переменной длины
	!
EXTV	! выделить поле
EXTZV	! выделить поле с расширением нулями
INSV	! включить поле
CMPV	! сравнить поле
CMPZV	! сравнить поле, расширенное нулями
FFS	! найти первый установленный бит
FFC	! найти первый сброшенный бит
	! инструкции для работы с очередями
	!
INSQUE	! занести элемент в очередь
INSQHI	! занести элемент в начало очереди

Обозначение!	Содержание
	! с блокировкой памяти
INSQTI	! занести элемент в конец очереди с ! блокировкой памяти
REMQUE	! удалить элемент из очереди
REMQHI	! удалить элемент из начала очереди ! с блокировкой памяти
REMQTI	! удалить элемент из конца очереди ! с блокировкой памяти
	! инструкция для работы с индексами !
INDEX	! вычислить индекс
	! инструкции для работы с адресами !
MOVA_	! переслать адрес (B,W,L=F,Q=D=G,O=H)
PUSHA_	! заслать адрес (B,W,L=F,Q=D=G,O=H) ! в стек
	! инструкции для работы с общими регистрами !
PUSHL	! заслать длинное слово в стек
PUSHR	! заслать содержимое регистров в стек

Обозначение!	Содержание
POPR	! выбрать содержимое регистров из стека
MOVPSL	! переслать из PSL
BISPSW	! установить бит в PSW
BICPSW	! очистить бит в PSW
	! инструкции безусловного перехода
	!
BR_	! безусловный переход со смещением (B,W)
JMP	! абсолютный безусловный переход
	! инструкции перехода по коду условия
	!
BLSS	! переход по меньше
BLSSU	! беззнаковый переход по меньше
BLEQ	! переход по меньше или равно
BLEQU	! беззнаковый переход по меньше или равно
BEQL	! переход по равно
BEQLU	! беззнаковый переход по равно
BNEQ	! переход по неравенству
BNEQU	! беззнаковый переход по неравенству
BGTR	! переход по больше
BGTRU	! беззнаковый переход по больше
BGEQ	! переход по больше или равно

Обозначение!	Содержание
BGEQU	! беззнаковый переход по больше или равно
BCC	! переход по отсутствию переноса
BCS	! переход по наличию переноса
BVS	! переход по переполнению
BVC	! переход по отсутствию переполнения
	! инструкции перехода по значению бита
BLB_	! переход по состоянию младшего бита
	! (установлен - S, сброшен - C)
BB_	! переход по состоянию бита (установлен - S,
	! сброшен - C)
BBS_	! переход, если бит установлен (и установка
	! - S или сброс - C бита)
BBC_	! переход, если бит сброшен (и установка - S
	! или сброс - C бита)
BBSSI	! переход, если бит установлен, и установка
	! с блокировкой
BBCCI	! переход, если бит сброшен, и очистка с
	! блокировкой
	! инструкции циклического и многофункционального
	! перехода
	!
ACB_	! сложить со сравнением и переход (B, W, L)

Обозначение!	Содержание
	! F, D, G, H)
AOBLEQ	! прибавить единицу и переход, если меньше ! или равно
AOBLSS	! прибавить единицу и переход, если меньше
SOBGEQ	! отнять единицу и переход, если больше ! или равно
SOBGTR	! отнять единицу и переход, если больше
CASE_	! многофункциональный переход (B, W, L)
	! инструкции вызова и возврата из подпрограммы !
BSB_	! переход к подпрограмме со смещением (B, W)
JSB	! абсолютный переход к подпрограмме
RSB	! возврат из подпрограммы
	! инструкции вызова и возврата из процедуры !
CALLG	! вызов процедуры с общим списком аргументов
CALLS	! вызов процедуры со списком аргументов ! в стеке
RET	! возврат из процедуры ! !

Обозначение!	Содержание
	! инструкции вызова и возврата из защищенных процедур
	!
CHM_	! изменить режим (ядра - K, выполнения - E, супервизора - S, пользователя - U)
REI	! возврат из исключительной ситуации или прерывания
PROBER	! проверить доступ по чтению
PROBEW	! проверить доступ по записи
SVPCTX	! сохранить контекст процесса
LDPCTX	! загрузить контекст процесса
MTPR	! переслать в регистр процесса
MFPR	! переслать из регистра процесса
	! специальные инструкции
	!
CRC	! циклический контроль по избыточности
BPT	! останов по ошибке
XFC	! вызов расширенных функций
NOP	! отсутствие операции
HALT	! останов

00152-01 31 01

1) символ подчеркивания (), следующий за именем инструкции, означает, что данная инструкция будет выполняться над любым типом данных, записанным в скобках.

2) В - байт; W - слово; L - длинное слово; Q - квадратное слово; O - октаслово; F - число с плавающей запятой F_формата; D - число с плавающей запятой D_формата; G - число с плавающей запятой G_формата; H - число с плавающей запятой H_формата.

Более подробно набор инструкций рассматривается в [5].

3.11. Системы программирования

МОС ВП включает язык МАКРОАССЕМБЛЕР СМ 1700 для программирования с использованием набора машинных команд и большое количество различных систем программирования, предназначенных для подготовки, отладки, трансляция и исполнения программ, написанных на языках высокого уровня ФОРТРАН, КОБОЛ, БЕЙСИК, ПЛ/1, ПАСКАЛЬ, КОРАЛ, БЛИСС, МОДУЛА-2, СИ. Под управлением МОС ВП работает система программирования ДИАМС/МОС ВП.

Использование систем программирования позволяет существенно облегчить труд программистов и повысить его производительность, что обеспечит более широкое применение вычислительного комплекса СМ 1700 и других программно-совместимых с ним ЭВМ в различных отраслях народного хозяйства.

3.11.1. Система программирования на языке МАКРОАССЕМБЛЕР СМ 1700

Язык МАКРОАССЕМБЛЕР СМ 1700, являясь машинно-ориентированным языком, отражает новые широкие возможности архитектуры ВК СМ 1700, сохраняя при этом общие черты, свойственные языку программирования типа АССЕМБЛЕР в предшествующих моделях СМ ЭВМ.

Язык МАКРОАССЕМБЛЕР СМ 1700 характеризуется следующими свойствами:

1) поддерживает разнообразные типы данных, в том числе и такие новые типы данных, как битовое поле переменной длины и очередь;

2) предоставляет программисту различные режимы адресации, которые обеспечивают гибкость в размещении и обработке структур данных;

3) предусматривает широкий систематизированный набор инструкций, позволяющий создавать эффективные программы;

4) имеет расширенный набор директив, включая директивы для:

- определения макрокоманд;
- создания программных секций;
- условной трансляции;
- определения блоков повторения;
- работы с библиотекой макрокоманд;
- управления распечаткой листинга трансляции.

Язык МАКРОАССЕМБЛЕР СМ 1700 является мощным инструментом для разработки системного программного обеспечения.

3.11.2. Система программирования ФОРТРАН

Система программирования ФОРТРАН предоставляет программисту следующие возможности:

- использовать язык программирования высокого уровня ФОРТРАН;
- обращаться к системным сервисным средствам и процедурам исполнительной библиотеки;
- программа, написанная на языке программирования ФОРТРАН, может вызывать внешние подпрограммы, написанные на любом языке программирования высокого уровня, поддерживаемом в системе;
- обрабатывать как последовательные и относительные, так и индексные файлы;
- использовать разделяемый транслятор.

Транслятор языка программирования ФОРТРАН производит разделяемые оптимизированные объектные программы. Транслятор использует преимущества большого виртуального адресного пространства МОС ВП для операций над числами с плавающей запятой и символьными строками. Обработка операций ввода-вывода поддерживается системными средствами управления данными. Объектные модули ФОРТРАНА могут быть скомпонованы с объектными модулями МАКРОАССЕМБЛЕРА и программами исполнительной библиотеки, которая является общей для всех программ основного режима, для обеспечения стандартных библиотечных функций.

3.11.3. Система программирования КОБОЛ

Система программирования КОБОЛ предназначена для создания высокоэффективных программ, выполняемых в основном режиме, и предоставляет программисту следующие средства:

- возможность работать с символьными данными;
- выполнение операций сортировки и слияния в исходных модулях;
- поддержку всех типов организации файлов;
- структурное программирование;
- поддержку всех типов данных, включая упакованные десятичные числа и числа с плавающей запятой;
- программы, написанные на КОБОЛЕ, могут вызывать внешние подпрограммы, написанные на КОБОЛЕ или других языках программирования высокого уровня, поддерживаемых в системе;
- возможность писать разделяемые программы для использования в других системах программирования основного режима ВК СМ 1700;
- использовать для записи программ различные форматы.

3.11.4. Система программирования БЕЙСИК

Система программирования БЕЙСИК предоставляет пользователям следующие возможности:

- использование средств системы управления данными для обработки файлов и записей;
- работа с символьными строками;

- редактирование программ;
- режим непосредственного исполнения программ, расширяющий возможности работы в интерактивном режиме.

3.11.5. Система программирования ПЛ/1

Система программирования ПЛ/1 обладает следующими особенностями:

- поддерживает все типы данных;
- использует широкие возможности ввода-вывода, включая поддержку ISAM;
- управляет распределением памяти со стороны пользователя;
- обрабатывает исключительные ситуации;
- поддерживает стандартный интерфейс вк СМ 1700 для вызовов, включая доступ к сервисным средствам МОС ВП и исполнительной библиотеке;
- имеет быстродействующий оптимизирующий транслятор основного режима;

3.11.6. Система программирования ПАСКАЛЬ

Язык программирования ПАСКАЛЬ получил особое распространение как язык общего назначения.

Он имеет следующие особенности:

- разбиение программы на блоки, используя составной оператор BEGIN...END;
- структурирование данных, включая возможность описа-

ния и использования указателей, записей, файлов и массивов;

- процедуры и функции для управления вводом-выводом и работы с данными.

Система программирования ПАСКАЛЬ представляет пользователю многие возможности, общие для других систем программирования основного режима, включая:

- отдельную компиляцию модулей;
- стандартный интерфейс вызова подпрограмм, написанных на других языках программирования;
- доступ к сервисным средствам МОС ВП.

3.11.7. Система программирования КОРАЛ

Система программирования КОРАЛ обеспечивает разработку задач, ориентированных на использование в системах реального времени. Транслятор языка КОРАЛ работает в режиме совместимости, создает объектные программы для работы в основном режиме ВК СМ 1700 и предлагает пользователям ряд возможностей, включая:

- несколько числовых типов данных;
- генерацию повторно входимых программ на уровне процедур;
- средства оптимизации программ;
- механизм сообщений об ошибках.

3.11.8. Система программирования БЛИСС

Язык программирования БЛИСС является инструментальным системным языком программирования высокого уровня, который используется в основном режиме под управлением МОС ВП. Он разработан специально для создания языковых трансляторов, утилит и системного программного обеспечения.

Язык программирования БЛИСС содержит многие средства языков программирования высокого уровня (цикл DO, утверждение IF-THEN-ELSE, автоматический стек и механизмы для определения и вызова подпрограмм), а также обеспечивает гибкость и доступ к аппаратуре, что обычно доступно только из языка АССЕМБЛЕРА.

Система программирования БЛИСС предоставляет программисту следующие возможности:

- выполнение программ на машинах различной архитектуры без или с незначительной их модификацией;
- конструирование сложных выражений, в которых несколько различных видов операций могут выполняться в одном операторе программы;
- эксплуатацию конструкций языка программирования высокого уровня.

3.11.9. Система программирования МОДУЛА-2

Система программирования МОДУЛА-2 обеспечивает:

- статический контроль типов данных;
- модульное программирование;
- возможность раздельной трансляции;
- возможность выделения модулей со специализированными интерфейсами (модулей определения и модулей реализации).

3.11.10. Система программирования СИ

Система программирования СИ обеспечивает разработку высокоэффективных (оптимальных по памяти и быстродействию) системных и прикладных программ, обладающих свойствами мобильности (переносимости). При этом система программирования характеризуется:

- экономичностью языковых конструкций;
- полным набором типовых структур управления;
- наличием средств описания элементарных и агрегированных данных;
- средствами статистического и динамического распределения памяти;
- развитым аппаратом арифметических и логических операций, включая битовые логические операции и операции сдвига;
- возможностью раздельной трансляции модулей, рекурсивным вызовом процедур, наличием интерфейсов с раз-

ноязыковыми модулями;

- наличием средств трансляционной макробработки текста программ.

3.11.11. Система программирования ДИАМС/МОС ВП

Диалоговая многопользовательская система, работающая под управлением операционной системы МОС ВП (ДИАМС/МОС ВП), предназначена для создания сложных территориально-распределенных автоматизированных систем информационного обеспечения, требующих оперативного хранения и обработки больших объемов данных, организованных в иерархические древовидные структуры.

Система основана на языке программирования ДИАМС. ДИАМС/МОС ВП предоставляет пользователю возможность создавать программы и массивы по тем же правилам, которые приняты в системах ДИАМС-2 и ДИАМС-3, реализованных на ЭВМ СМ 1420. Создание и выполнение программ пользователь ведет в режиме интерпретации. По сравнению с системой ДИАМС-3 ДИАМС/МОС ВП имеет улучшенные характеристики функционирования и расширенные возможности применения.

3.12. Средства разработки программ

МОС ВП предоставляет пользователю широкий набор средств для разработки программ, включающий в себя, помимо трансляторов, редакторы текстов, компоновщик, программы общей исполнительной библиотеки, программу библиотечарь,

символический отладчик, программу изменения образа задания и ряд других служебных программ.

МОС ВП обеспечивает работу нескольких текстовых редакторов, в том числе универсального построчно/символьного текстового редактора. При работе в символьном режиме универсальный редактор EDT позволяет пользователю быстро вставить или удалить любое количество текста внутри уже существующего текста.

Универсальный текстовый редактор запоминает произведенные изменения и может воспроизвести их по требованию, если в процессе редактирования произойдет сбой [6].

Компоновщик объединяет объектные модули вместе с другими необходимыми данными в выполняемые образы, которые могут быть непосредственно выполнены под управлением МОС ВП. Основными функциями компоновщика являются: разрешение символьных ссылок, распределение виртуальной памяти и инициализация образа. Компоновщик облегчает модульную разработку программ. Сложные программы могут быть разбиты на небольшие модули, которые проще написать и отладить. Различные модули одной программы могут быть написаны на различных языках программирования. При компоновке эти модули объединяются в единый образ. В результате компоновки могут быть созданы карта образа и файл таблицы символов.

Компоновщик позволяет также создавать разделяемые образы, которые могут использоваться для разделения одной физической копии процедур и/или данных несколькими процессами пользователей [7].

Общая исполнительная библиотека обеспечивает:

- получение записи с устройства, выполнение строковых операций, преобразование типов данных для ввода-вывода, определение системной даты и времени;
- выполнение математических операций, таких как арифметические, алгебраические и тригонометрические функции;
- выделение и освобождение виртуальной памяти, локальных флагов событий, номеров логических устройств для систем программирования бейсик и фортран;
- выполнение функций управления экраном, которые не зависят от типа терминала;
- выполнение операций, связанных с обработкой прерываний;
- выполнение синтаксического анализа строк;
- возможность использования процедур перекрестных ссылок, которые получают данные о перекрестных ссылках, представляют их в компактном виде и форматируют для вывода, причем они могут быть вызваны из программ, написанных на макроассемблере или на любом языке программирования высокого уровня, поддерживаемом системой.

Исполнительная библиотека обеспечивает выполнение программ, написанных на языках программирования БЕЙСИК, КОБОЛ, ФОРТРАН и ПАСКАЛЬ. Все процедуры имеют стандартные правила вызова и обработки условий.

Программисты, использующие эти языки программирования,

могут не заботиться о механике вызова процедур исполнительной библиотеки или об обращениях к базовым функциям МОС ВП. Необходимые функции таких обращений встроены в трансляторы. Пользователь имеет возможность добавлять процедуры в исполнительную библиотеку, а также может написать новые базовые системные функции [8].

Программа библиотекарь позволяет программисту создавать и поддерживать собственные пользовательские библиотеки. Эти библиотеки представляют собой индексные файлы и содержат объектные модули часто используемых макрокоманд, текстовые файлы и модули с различной вспомогательной информацией. Типы библиотек различаются по типам хранящихся в них данных [9].

Символический отладчик работает в интерактивном режиме и может использоваться для отладки программ, разработанных с помощью большинства систем программирования, поддерживаемых МОС ВП.

Символический отладчик позволяет программистам производить трассировку программ, выводить на дисплей и модифицировать содержимое ячеек памяти, используя при этом символические имена, которые были использованы в исходных текстах программ, задавать точки останова и осуществлять пошаговое выполнение программы [10].

Программа корректировки образа задачи позволяет модифицировать программы без повторной трансляции и компоновки. Изменения вносятся непосредственно в файлы загрузочных, или исполняемых модулей. Доступ к адресам программы возможен

посредством тех же символических имен, которые встречаются в исходном тексте. Кроме того, программа производит автоматическое перемещение вставляемых частей программы, которые превосходят по размеру заменяемые части [11].

3.13. Обеспечение надежности функционирования МОС ВП

МОС ВП является операционной системой повышенной надежности. Надежность функционирования обеспечивается следующими программными и аппаратными средствами:

- защитой памяти заданий и системы от несанкционированного доступа;
- авторизацией доступа к системе и к файлам;
- корректировкой ошибок устройств;
- возможностью диагностирования устройств автономно и под управлением МОС ВП;
- возможностью сбора и распечатки статистики об ошибках периферийных устройств;
- устойчивостью к ошибкам оператора;
- регистрацией и анализом состояния системы при аварийных ситуациях;
- возможностью сохранения и восстановления носителей данных;
- выводом на терминал сообщений об ошибках оператора и заданий.

Более подробная информация об обеспечении надежности функционирования МОС ВП содержится в [12], [13], [14].

4. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ

Подход к освоению операционной системы МОС ВП должен определяться теми целями и задачами, которые ставит перед собой пользователь.

Пользователи операционной системы МОС ВП делятся на 4 категории:

- 1) программист;
- 2) оператор;
- 3) системный программист;
- 4) администратор системы.

Последние две категории пользователей обычно совпадают, т.к. Системный программист выполняет и функции администратора.

Для каждой категории пользователей разработана соответствующая документация, изучение которой облегчает и ускоряет процесс освоения МОС ВП.

Перечень устройств, поддерживаемых операционной системой МОС ВП

Перечень устройств, поддерживаемых операционной системой МОС ВП, указан в таблице.

Таблица 1

Наименование устройства	!Символическое! ! обозначение !	Возможный тип устройства

комбинированное устройство! ввода-вывода на кассетной ! магнитной ленте !	DD	! TU58, см 5218 ! !
устройство внешней памяти ! на с'емном магнитном диске! емкостью 10 Мбайт, !	DL	! RL02 ! !
подключаемое на общую шину! устройство внешней памяти ! на с'емном магнитном диске! емкостью 10 Мбайт, !	DQ	! RL02 ! !
подключаемое через IDC ! устройство внешней памяти ! (емкость 121 Мбайт), !	DQ	! R80 ! !

Продолжение табл. 1

Наименование устройства	!Символическое! ! обозначение !	Возможный тип устройства
подключаемое через IDC	!	!
устройство внешней памяти на с'емном магнитном диске!	DM	! RK06, RK07, ! CM 5408
емкостью 16 Мбайт	!	!
устройство внешней памяти	DU	! RA81, RA80,
на магнитном диске "вин- честерского" типа емкостью!	!	! CM 5504
121 Мбайт, подключаемое	!	!
через UDA	!	!
устройство внешней памяти	DJ	! RA60
на магнитном диске	!	!
емкостью 456 Мбайт,	!	!
подключаемое через UDA	!	!
устройства внешней памяти	DR	! RM03/05/80, RP07
на магнитных дисках,	!	!
подключаемые через UDA	!	!
устройство внешней памяти	DB	! RP05
на магнитном диске	!	!
емкостью 80 Мбайт	!	!
устройство внешней памяти	!	! SC21/SC31,
на магнитном диске емко-	!	! RM03_EMULEX
стью 80 Мбайт, подключен-	!	! CM 5412

Продолжение табл. 1

Наименование устройства	!Символическое! ! обозначение !	Возможный тип устройства
ное к контроллеру CM 5110	!	!
устройство внешней памяти на гибком магнитном диске	DX	! RX01
устройство внешней памяти на гибком магнитном диске двойной плотности	DY	! RX02
устройство алфавитно-цифровое параллельной печати с расширенным набором символов	!	!
устройство внешней памяти на магнитных лентах с плотностью 6250/1600	LP	! LP26
устройство внешней памяти на магнитных лентах с плотностью 1600/800	!	!
устройство внешней памяти на магнитных лентах с плотностью 1600	MM	! TU78
устройство печати	MT	! TU45, TU77, TE16
консольное с клавиатурой	MS	! TU80, TS11, CM 5309
	!	!
	MU	! TU81
	OP	! LA120, CM 6380
	!	!

Продолжение табл. 1

Наименование устройства	!Символическое! ! обозначение !	! Возможный тип устройства
удаленный терминал	! RT	! VT100, VT220, ! VT240, ! CM 7224 (BTA- ! 2000-17), ! CM 7225 (BTA- ! 2000-18)
интерактивный терминал	! TT	! VT100, VT200, ! VT240, ! CM7225 (BTA ! 2000-18)
синхронный адаптер	! XS	! DUP11
модуль связи	! XM	! DMC11
терминал, подключенный через DMF32A	! TX	! VT100, ! BTA-200015M
синхронный порт DMF32	! XG	! DMF32
ацпу, подключенное через DMF32P	! LC	! LP26 !
графическая станция	! VB	! VS100
графическая станция	! VS	! VS11, ! CM7317

Основные термины МОС ВП

Приложение содержит три раздела, в которых описаны основные термины, используемые в документации по МОС ВП:

1) краткий список основных терминов на русском языке в алфавитном порядке с английским переводом;

2) список терминов на английском языке в алфавитном порядке с кратким объяснением термина;

3) сокращения и условные обозначения МОС ВП.

1. КРАТКИЙ СПИСОК ТЕРМИНОВ НА РУССКОМ ЯЗЫКЕ

Абсолютное время - ABSOLUTE TIME

Абсолютный индексный режим - ABSOLUTE INDEXED MODE

Абсолютный режим - ABSOLUTE MODE

Аварийное завершение - ABORT

Аварийное завершение по разединению - DISCONNECT ABORT

Аварийный отказ - CRASH

Аварийный сигнал - ALARM

Аварийный сигнал защиты - SECURITY ALARM

Автоматический захват записи - AUTOMATIC RECORD LOCKING

Адаптер общей шины - UBA, UNIBUS ADAPTER

Адрес - ADDRESS

00152-01 31 01

- Адрес записи в файле - RECORD'S FILE ADDRESS (RFA)
- Адрес подчиненной станции - TRIBUTARY ADDRESS
- Адрес точки входа - TRANSFER ADDRESS
- Адрес узла - NODE ADDRESS
- Адресация рассылки - BROADCAST ADDRESSING
- Адресное пространство - ADDRESS SPACE
- Адресное пространство процесса - PROCESS ADDRESS SPACE
- Адресное пространство системы - SYSTEM ADDRESS SPACE
- Адресный тип доступа - ADDRESS ACCESS TYPE
- Активизатор образа - IMAGE ACTIVATOR
- Активный элемент - ACTIVE COMPONENT
- Альтернативный ключ, вторичный ключ - ALTERNATE KEY
- Аппаратный блок управления процессом, аппаратный рсв -
HARDWARE PROCESS CONTROL BLOCK
- Аппаратный адрес - HARDWARE ADDRESS
- Аппаратный контекст - HARDWARE CONTEXT
- Аппаратура передачи данных - DATA CIRCUIT-TERMINATING
EQUIPMENT (DCE)
- AST по захвату - BLOCKING AST
- Асинхронная операция над записью - ASYNCHRONOUS RECORD
OPERATION
- Асинхронное системное прерывание - ASYNCHRONOUS SYSTEM
TRAP (AST)
- Ассемблер - ASSEMBLER
- Ассоциативная двусторонняя кэш-память - TWO-WAY ASSOCIATIVE
CACHE
- Атрибут - ATTRIBUTE

База данных - DATABASE

База данных ввода-вывода - I/O DATABASE

База данных конфигурации - CONFIGURATION DATABASE

База данных прав по идентификаторам - RIGHTS DATABASE

Базовый адрес операнда - BASE OPERAND ADDRESS

Базовый приоритет - BASE PRIORITY

Базовый регистр - BASE REGISTER

Базовый регистр блока управления процессом - PCBW

Базовый регистр управления системой - SYSTEM CONTROL

BLOCK BASE REGISTER (SCBB)

Базовый регистр области программы - PROGRAM REGION BASE

REGISTER (POBR)

Базовый регистр области управления - CONTROL REGION BASE

REGISTER (P1BR)

Базовый регистр системы - SYSTEM BASE REGISTER (SBR)

Байт - BYTE

Бакет - BUCKET

Балансный набор - BALANCE SET

Библиотекарь - LIBRARIAN

Библиотечный файл - LIBRARY FILE

Битовая строка - BIT STRING

Битовое поле переменной длины - VARIABLE-LENGTH BIT FIELD

(VBF)

Блок - BLOCK

Блок данных прерывания - INTERRUPT DATA BLOCK (IDB)

Блок данных устройства - DEVICE DATA BLOCK (ddb)

Блок дополнительных атрибутов - EXTENDED ATTRIBUTE BLOCK
(XAB)

Блок доступа к записям - RECORD ACCESS BLOCK (RAB)

Блок доступа к файлу - FILE ACCESS BLOCK (FAB)

Блок запросов канала - CHANNEL REQUEST BLOCK (CRB)

Блок значения захвата - LOCK VALUE BLOCK

Блок имени файла - NAME BLOCK (NAM)

Блок начальной загрузки - BOOTSTRAP BLOCK

Блок соединений сети - NETWORK CONNECT BLOCK (NCB)

Блок состояния ввода-вывода - I/O STATUS BLOCK (IOSB)

Блок управления адаптером - ADAPTER CONTROL BLOCK (ADP)

Блок управления выходом - ECB

Блок управления каналом - CHANNEL CONTROL BLOCK (CCB)

Блок управления окном - WCB

Блок управления процессом - PROCESS CONTROL BLOCK (PCB)

Блок управления системой - SYSTEM CONTROL BLOCK (SCB)

Блок управления томом - VCB

Блок управления устройством - UNIT CONTROL BLOCK (UCB)

Блок управления файлом - FILE CONTROL BLOCK (FCB)

Блокирование записей - RECORD BLOKING

Блокировка страницы в памяти - LOCKING A PAGE IN MEMORY

Блокировка страницы в рабочем наборе - LOCKING A PAGE IN
THE WORKING SET

Блокировка страницы по вводу-выводу - I/O LOCKDOWN

Буквенно-цифровой символ - ALPHANUMERIC CHARACTER

Буквенно-цифровой UIC - ALPHANUMERIC UIC

Буфер - BUFFER

Буфер для склеек - PASTE BUFFER
Буфер замены - SUBSTITUTE BUFFER
Буфер инструкций - INSTRUCTION BUFFER
Буфер поиска строки - STRING SEARCH BUFFER
Буфер символов - CHARACTER BUFFER
Буфер строки - LINE BUFFER
Буфер трансляции - TRANSLATION BUFFER
Буферизованный ввод-вывод - BUFFERED I/O
Буферизованный путь данных - BUFFERED DATA PATH

Ввод-вывод блоков - BLOCK I/O
Вектор - VECTOR
Вектор исключительной ситуации - EXCEPTION VECTOR
Вектор прерывания - INTERRUPT VECTOR
Взаимное исключение - MUTEX
Взаимосвязанные задачи - COOPERATING TASKS
Видеотерминал - VIDEO TERMINAL
Виртуальная память - VIRTUAL MEMORY
Виртуальная цепь - VIRTUAL CIRCUIT
Виртуальное адресное пространство - VIRTUAL ADDRESS SPACE
Виртуальный адрес - VIRTUAL ADDRESS
Виртуальный диск - VIRTUAL DISK
Владелец - OWNER
Внешняя память - SECONDARY STORAGE
Внутреннее соединение общей шины - UBI
Внутренняя диагностическая проверка - BUGCHECK
Внутренняя синхронная шина - SYNCHRONOUS BACKPLANE

INTERCONNECT (SBI)

Внутренняя шина процессора - BACKPLANE INTERCONNECT

Возвращаемый код состояния - RETURN STATUS CODE

Все - WORLD

Вспомогательная память - BACKING STORE

Вспомогательный управляющий процесс - ANCILLARY CONTROL
PROCESS (ACP)

Вторичный вектор - SECONDARY VECTOR

Вторичный ключ - ALTERNATE KEY

Вторичный пароль - SECONDARY PASSWORD

Вход в систему - LOGGING IN

Вход в систему по доверенности - PROXY LOGIN

Входной поток - INPUT STREAM

Выделенный маршрутный узел - DESIGNATED ROUTER

Вызов - CALL

Выполняемый образ - EXECUTABLE IMAGE

Выход - EXIT

Выход из системы - LOGGING OUT

Выходной файл - OUTPUT FILE

Главный каталог файлов - MFD

Главный узел - HOST NODE

Глобальная секция - GLOBAL SECTION

Глобальный - GLOBAL

Глобальный символ - GLOBAL SYMBOL

Группа - GROUP

Групповая метка - LABEL GROUP

Дамп смежного узла - UPLINE DUMP

Данные - DATA

Датаграмма - DATAGRAM

Двусторонняя замкнутая группа пользователей - BILATERAL
CLOSED USER GROUP (BCUG)

Декодирование - DECRYPTION

Дельта-время, интервал времени - DELTA TIME

Дескриптор - DESCRIPTOR

Дескриптор символьной строки - CHARACTER STRING DESCRIPTOR

Десятичное переполнение - DV

Держатель идентификатора - HOLDER

Диагностика - DIAGNOSTIC

Диалоговый командный язык - DIALOG COMMAND LANGUAGE
(DCL)

Динамический доступ - DYNAMIC ACCESS

Диспечер исключительных ситуаций - EXCEPTION DISPATCHER

Диспечер заданий - JOB CONTROLLER

Дистрибутивный носитель - DISTRIBUTION KIT

Дисциплина обслуживания FIFO - FIFO

Дисциплина обслуживания LIFO - LIFO

Длина записи - RECORD LENGTH

Длина маршрута - PATH LENGTH

Длинное слово - LONGWORD

Длинное слово состояния процессора - PROCESSOR STATUS
LONGWORD (PSL)

Дозавершение ввода-вывода, довывод - I/O RUNDOWN

Должно быть равно нулю - MBZ

Дополнительный код - TWO'S COMPLEMENT

Дорожка - TRACK

Доступный узел - REACHABLE NODE

Драйвер - DRIVER

Драйвер ввода-вывода - I/O DRIVER

Драйвер устройства - DEVICE DRIVER

Дублирование, дублировать - BACKUP

Ждать - WAIT

Журнальный файл - JOURNAL FILE

Заблокированный пароль - LOCKED PASSWORD

Завершение - EXIT

Завершение образа - IMAGE EXIT

Зависание системы - HANGING

Заголовок процесса - PROCESS HEADER

Заголовок файла - FILE HEADER

Загрузка - BOOT

Задание - JOB

Задача - TASK

Задержанное эхо - DEFFERED ECHO

Закрепить устройство - ALLOCATE A DEVICE

Замкнутый узел - LOOP NODE

Замкнутая группа пользователей - CLOSED USER GROUP (CUG)

Запись - RECORD

Запись переменной длины с постоянной управляющей частью -

00152-01 31 01

VARIABLE WITH FIXED-LENGTH CONTROL RECORD FORMAT

Запрос на внешнее соединение - OUTBOUND CONNECTION

Запрос на ожидание прерывания - WAIT FOR INTERRUPT REQUEST

Запросы на соединение - INBOUND CONNECTION

Захват бакетов - BUCKET LOCKING

Захват записи - RECORD LOCKING

Захваченный пароль - LOCKED PASSWORD

Защита - PROTECTION

Защитное действие - EVASIVE ACTION

Значение исключительного состояния - CONDITION VALUE

Идентификатор - IDENTIFIER

Идентификатор метки - LABEL IDENTIFIER

Идентификатор процесса - PROCESS IDENTIFICATION (PID)

Идентификатор файла - FILE IDENTIFIER (FID)

Известный элемент - KNOWN COMPONENT

Имя каталога - DIRECTORY NAME

Имя образа - IMAGE NAME

Имя пользователя - USER NAME

Имя процесса - PROCESS NAME

Имя узла - NODE NAME

Имя устройства - DEVICE NAME

Имя устройства загрузки - BOOT NAME

Имя файла - FILE NAME

Индекс - INDEX

Индексная организация файла - INDEXED FILE ORGANIZATION

Индексный регистр - INDEX REGISTER

00152-01 31 01

Индексный режим адресации - INDEXED ADDRESSING MODE

Индексный режим с автоувеличением - AUTOINCREMENT INDEXED
MODE

Индексный режим с автоуменьшением - AUTODECREMENT INDEXED
MODE

Индексный режим со смещением - DISPLACEMENT INDEXED MODE

Индексный файл - INDEX FILE

Инструкции вызова - CALL INSTRUCTIONS

Интерактивная система - INTERACTIVE SYSTEM

Интерпретатор команд - COMMAND INTERPRETER

Информационный блок задания - JOB INFORMATION BLOCK (JIB)

Исполнительная библиотека - RUN-TIME PROCEDURE LIBRARY

Исключительная ситуация - EXCEPTION

Исполнительный адрес - EFFECTIVE ADDRESS

Исходная задача - SOURCE TASK

Исходный файл - SOURCE FILE

Кадр - FRAME

Кадр вызова - CALL FRAME

Кадр стека - STACK FRAME

Канал - CHANNEL

Канал ввода-вывода процесса - PROCESS I/O CHANNEL

Канал данных контроллера - CONTROLLER DATA CHANNEL

Канал параллельного доступа - MULTIACCESS CHANNEL

Карта памяти индексного файла - INDEX FILE BIT MAP

Каталог - DIRECTORY

Каталог файлов пользователя - USER FILE DIRECTORY (UFD)

Квадрослово - QUADWORD

Квалификатор - QUALIFIER

Квота - QUOTA

Класс распределения - ALLOCATION CLASS

Класс событий - EVENT CLASS

Кластер - CLUSTER

Кластер общих флагов событий - COMMON EVENT FLAG CLUSTER

Кластер флагов событий - EVENT FLAG CLUSTER

Ключ, символ, переключатель - KEY, SWITCH

Ключевое слово - KEYWORD

Код двоично - EBCDIC

Код идентификации пользователя - USER IDENTIFICATION CODE
(UIC)

Код КОИ-8 - AMERICAN STANDARD CODE FOR INFORMATION
INTERCHANGE (ASCII)

Код операции - OPCODE

Код с использованием ошибок - ECC

Код состояния - STATUS CODE

Код функции - FUNCTION CODE

Код функции ввода-вывода - I/O FUNCTION CODE

Кодирование - ENCRYPTION

Коды условий - CONDITION CODES

Команда - COMMAND

Командная процедура - COMMAND PROCEDURE

Командная строка - COMMAND STRING

Командный узел - COMMAND NODE

Командный уровень - COMMAND LEVEL

Командный файл - COMMAND FILE

Коммутация пакетов - PACKET SWITCHING

Коммутируемая виртуальная цепь - SWITCHED VIRTUAL CIRCUIT
(SVC)

Компоновка - LINKING, BINDING

Компоновщик - LINKER

Конечный узел - END NODE

Консоль - CONSOLE

Консоль оператора - OPERATOR'S CONSOLE

Консольный регистрационный приемник - LOGGING CONSOLE

Консольный терминал - CONSOLE TERMINAL

Конспект - SUMMARY

Контекст - CONTEXT

Контекст процесса - PROCESS CONTEXT

Контекстная индексация - CONTEXT INDEXING

Контроль - AUDITING

Контроль защиты по усмотрению - DISCRETIONARY CONTROLS

Контроль по циклической избыточности - CRC

Копирование файлов с сохранением атрибутов - REORGANIZATION

Копия по ссылке - COPY-ON-REFERENCE

Косвенно-регистрационный режим - REGISTER DEFERRED MODE

Косвенно-регистрационный режим с индексацией - REGISTER DEFERRED INDEXED MODE

Косвенный индексный режим с автоувеличением -

AUTOINCREMENT DEFERRED INDEXED MODE

Косвенный индексный режим со смещением - DISPLACEMENT

DEFERRED INDEXED MODE

Косвенный командный файл - INDIRECT COMMAND FILE

Косвенный режим с автоувеличением - AUTOINCREMENT DEFERRED
MODE

Косвенный режим со смещением - DISPLACEMENT DEFERRED MODE

Курсор - CURSOR

КЭШ-память - CACHE MEMORY

КЭШ-память прямого отображения - DIRECT MAPPING CACHE

Лексическая функция - LEXICAL FUNCTION

Лимит - LIMIT

Линия - LINE

Литеральный режим - LITERAL MODE

Личная секция - PRIVATE SECTION

Ловушка - TRAP

Логическая запись - LOGICAL RECORD

Логическое имя - LOGICAL NAME

Логический канал - LOGICAL LINK

Локальность - LOCALITY

Локальность программы - PROGRAM LOCALITY

Локальный диск - LOCAL DISK

Локальный символ - LOCAL SYMBOL

Локальный узел - LOCAL NODE

Максимальное число транзитных узлов - MAXIMUM VISITS

Малая клавиатура - KEYPAD

Малый процесс - SMALL PROCESS

Маркер страницы - PAGE MARKER

Маршрутизация - ROUTING

Маршрутизация по регионам - AREA ROUTING

Маршрутный узел - ROUTER

Маршрутный узел региона - AREA ROUTER

Маршрутный узел 1 уровня - LEVEL 1 ROUTER

Маршрутный узел 2 уровня - LEVEL 2 ROUTER

Маска входа - ENTRY MASK

Межблочный промежуток - INTERRECORD GAP (IRG)

Метка - LABEL

Метод "полного уровня" - HIGHWATER MARKING

Методы доступа к записям - RECORD ACCESS MODE

Многоязычный набор символов - MULTINATIONAL CHARACTER
SET

Многоточечная адресация - MULTICAST ADDRESSING

Многоточечный групповой адрес - MULTICAST GROUP ADDRESS

Модем - MODEM

Модификатор функции - FUNCTION MODIFIER

Модификатор функции ввода-вывода - I/O FUNCTION
MODIFIER

Модуль - MODULE

Монтировать том - MOUNT A VOLUME

Набор меток - LABEL SET

Набор томов - VOLUME SET

Назначить канал - ASSIGN A CHANNEL

Назначенный ресурс - DEDICATED

Наложение в канале, конфликтная ситуация - COLLISION

Накопитель - DRIVE

Нарушение доступа - ACCESS VIOLATION

Нарушение защиты - BREACH

Неавтоматическое освобождение записей - MANUAL RECORD
UNLOCKING

Немаршрутный узел - NONROUTING NODE

Непечатаемый символ - NONPRINTING CHARACTER

Непосредственный режим - IMMEDIATE MODE

Непостоянная база данных - VOLATILE DATABASE

Непрерывная область - CONTINUOUS AREA

Непрерывный - CONTINUOUS

Непривилегированный - NONPRIVILEGED

Номер версии - VERSION NUMBER

Номер виртуального блока - VIRTUAL BLOCK NUMBER (VBN)

Номер виртуальной страницы - VIRTUAL PAGE NUMBER (VPN)

Номер группы - GROUP NUMBER

Номер логического блока - LOGICAL BLOCK NUMBER (LBN)

Номер логического канала - LOGICAL CHANNEL NUMBER (LCN)

Номер метки - LABEL NUMBER

Номер пользователя - USER NUMBER

Номер программиста - PROGRAMMER NUMBER

Номер проекта - PROJECT NUMBER

Номер строки - LINE NUMBER

Номер физического блока - PHYSICAL BLOCK NUMBER

Номер физической страницы - PAGE FRAME NUMBER (PFN)

Номер члена - MEMBER NUMBER

Область, регион - AREA

Область программы - PROGRAM REGION (PO)

Область системы - SYSTEM REGION

Область управления - CONTROL REGION (P1)

Область управления фиксированной длины - FIXED-LENGTH
CONTROL AREA

Обмен рабочих наборов - SWAPPING

Обмен страниц - PAGING

Обнуленная по запросу страница - DEMAND ZERO PAGE

Образ - IMAGE

Образ основного режима - NATIVE IMAGE

Обратная запись - WRITE BACK

Обратное отслеживание - TRACEBACK

Обслуживающая программа, утилита - UTILITY

Обслуживающее устройство - SERVED DEVICE

Общий идентификатор - GENERAL IDENTIFIER

Общий регистр - GENERAL REGISTER

Объект, объектный - OBJECT

Объектный модуль - OBJECT MODULE

Ограничивающий учетный раздел - CAPTIVE ACCOUNTS,
TIED ACCOUNT, TURNKEY ACCOUNT

Окно - WINDOW

Окно отображения - MAPPING WINDOW

Оконечное оборудование данных - DATA TERMINAL EQUIPMENT
(DTE)

Октаслово (увосьмеренное слово) - OCTAWORD

Оператор присваивания - ASSIGNMENT STATEMENT

Операционная система - OPERATING SYSTEM

Опознавание - AUTHENTICATION

Организация файла - FILE ORGANIZATION

Основная память - MAIN MEMORY

Основной пароль - PRIMARY PASSWORD

Основной режим - NATIVE MODE

Основной текстовый буфер - MAIN TEXT BUFFER

Открытые учетные разделы - OPEN ACCOUNTS

Отладчик - DEBUGGER

Относительная организация файла - RELATIVE FILE
ORGANIZATION

Относительный набор тома - RVN

Относительный номер записи - RELATIVE RECORD NUMBER

Отображение номера физической страницы - PAGE FRAME NUMBER
MAPPING (PFN MAPPING)

Отсоединенный процесс - DETACHED PROCESS

Отсутствие информации - NULL

Очередь - QUEUE

Очередь заданий - JOB QUEUE

Очередь к устройству - DEVICE QUEUE

Очередь процессов по состоянию - STATE QUEUE

Очередь спула - SPOOL QUEUE

Память - MEMORY

Пакет - PACKET

Пакет запроса ввода-вывода - I/O REQUEST PACKET (IRP)

Пакетная обработка - BATCH PROCESSING

Параметр - PARAMETER
Параметр команды - COMMAND PARAMETER
Пароль - PASSWORD
Пароль пользователя - USER PASSWORD
Первая часть (инструкции) выполнена - FPD
Первичный вектор - PRIMARY VECTOR
Первичный ключ - PRIMARY KEY
Перевод строки - LINE FEED
Перевод формата - FORM FEED
Перезаписываемая диагностическая управляющая память - WDCS
Перезаписываемая управляющая память - WCS
Переключатель - SWITCH
Переключение метода доступа к записям - RECORD ACCESS MODE
SWITCHING
Переключение контекста - CONTEXT SWITCHING
Периферийные устройства - PERIPHERAL DEVICES
Планировщик памяти - SWAPPER
Плотность - DENSITY
Плотность записи - DENSITY
Подкаталог - SUBDIRECTORY
Подключение к вектору прерывания - CONNECT-TO-INTERRUPT
Подпрограмма - SUBROUTINE
Подпроцесс - SUBPROCESS
Подсказка - PROMPT
Подсостояние - SUBSTATE
Подчиненная станция - TRIBUTARY
Подчиненный терминал - SLAVE TERMINAL

Позиционно-зависимая программа - POSITION-DEPENDENT CODE

Позиционно-независимая программа - POSITION-UNDEPENDENT
CODE

Поле - FIELD

Полностью ассоциативная КЭШ-память - FULLY ASSOCIATIVE
CACHE

Полоса пропускания - BANDWIDTH

Попытка незаконного доступа - BREAKIN ATTEMPT

Последним пришел - первым обслужен - LIFO

Последовательная организация файла - SEQUENTIAL FILE
ORGANIZATION

Последовательность AP2 - ESCAPE SEQUENCE

Последовательность "рукопожатия" - HANDSHAKING SEQUENCE

Последовательный метод доступа - SEQUENTIAL ACCESS MODE

Поставить запрос в очередь ввода-вывода - QIO

Постоянная база данных - PERMANENT DATABASE

Постоянная виртуальная цепь - PERMANENT VIRTUAL CIRCUIT
(PVC)

Построчно-печатающее устройство - LINE PRINTER

Потеря из-за переполнения - CONGESTION LOSS

Поток - STREAM

Почтовый ящик - MAILBOX

Предварительный ввод - TYPE-AHEAD

Прерывание - INTERRUPT

Прерывание устройства - DEVICE INTERRUPT

Привилегии - PRIVILEGES

Привилегии образа - IMAGE PRIVILEGES

00152-01 31 01

Приаилегии пользователя - USER PRIVILEGES
Привилегии процесса - PROCESS PRIVILEGES
Привилегированный - PRIVILEGED
Приемная задача - TARGET TASK
Приемный узел - SINK NODE
Приоритет - PRIORITY
Приоритет в очереди - QUEUE PRIORITY
Приоритет программы - SOFTWARE PRIORITY
Приоритет процесса - PROCESS PRIORITY
Приостанов - SUSPENSION
Приращение, увеличение - INCREMENT
Пробудить - WAKE
Программа - PROGRAM
Программа драйвера запуска ввода-вывода - DRIVER START
I/O ROUTINE
Программа инициализации устройства - UNIT INITIALIZATION
ROUTINE
Программа обработки выхода - EXIT HANDLER
Программа обработки исключительной ситуации - ESR
Программа обработки исключительного состояния - CONDITION
HANDLER
Программа обмена страниц - PAGER
Программа обслуживания прерывания - INTERRUPT SERVICE
ROUTINE (ISR)
Программа связи с оператором - MONITOR CONSOLE ROUTINE
(MCR)
Программа системного обслуживания, системное обслуживание -

00152-01 31 01

SYSTEM SERVICES

Программа "троянский конь" - TROJAN HORSE PROGRAM
Программная защита терминала - SECURE TERMINAL SERVER
Программная секция - PROGRAM SECTION (PSECT)
Программное прерывание - SOFTWARE INTERRUPT
Программный блок управления процессом (программный PCB) -
SOFTWARE PROCESS CONTROL BLOCK (SOFTWARE PCB)
Программный контекст - SOFTWARE CONTEXT
Программы FDT - FDT ROUTINE
Продолжить, возобновить - RESUME
Прозрачный - TRANSPARENT
Произвольный доступ - RANDOM ACCESS
Произвольный доступ по адресу записи в файле - RANDOM
ACCESS BY RECORD'S FILE ADDRESS
Произвольный доступ по ключу - RANDOM ACCESS BY KEY
Произвольный доступ по относительному номеру записи -
RANDOM ACCESS BY RELATIVE RECORD NUMBER
"прокрутка" информации - SCROLLING
Прологовая таблица драйвера - DRIVER PROLOQUE TABLE (DPT)
Просмотр остатка данных на диске - DISK SCAVENGING
Пространство ввода-вывода - I/O SPACE
Пространство процесса - PROCESS SPACE
Пространство системы - SYSTEM SPACE
Пространство физических адресов - PHYSICAL ADDRESS SPACE
Протокол - PROTOKOL
Протокол доступа к каналу - LINK ACCESS PROTOCOL (LAP)
Протокол обслуживания в сети - NSP

Процедура - PROCEDURE

Процедура обмена страниц - PAGER

Процесс - PROCESS

Процесс-владелец - OWNER PROCESS

Процесс-прототип - SHELL PROCESS

Процесс обмена рабочих наборов - WORKING SET SWAPPER

Процесс реального времени - REAL-TIME PROCESS

Процесс сообщений оператору - OPERATOR COMMUNICATION
MANAGER (OPCOM)

Прямой ввод-вывод - DIRECT I/O

Прямой доступ к памяти - DMA

Прямой путь данных - DIRECT DATA PATH

Псевдоустройство - PSEUDO DEVICE

Пустая строка - NULL STRING

Пустой процесс - NULL PROCESS

Путь, маршрут - PATH

Рабочий набор - WORKING SET

Разблокированная запись - UNBLOCKED RECORDS

Разброс/сбор данных - SCATTER/GATHER

Развернуть стек вызова - UNWIND THE CALL STACK

Разделение времени - TIME-SHARING

Разделение канала передачи данных - DATA LINK MAPPING (DLM)

Разделение файла - FILE SHARING

Разделитель, ограничитель - DELIMITER

Разделяемая квота - POOLED QUOTA

Разделяемый образ - SHAREABLE IMAGE

00152-01 31 01

Размер кластера при страничном отказе - PAGE FAULT CLUSTER
SIZE

Разрешение исключительной ситуации - EXCEPTION ENABLES

Разрешение ловушки - TRAP ENABLE

Разрешение передачи - POLLING

Распределяемая запись - WRITE ALLOCATE

Расширение - EXTENSION

Расширение имени файла - FILE NAME EXTENSION

Расщепление бакета - BUCKET SPLIT

Реверсирование видеоизображения - REVERSE VIDEO

Регистр - REGISTER

Регистр длины области программы - PROGRAM REGION LENGTH
REGISTER (POLR)

Регистр длины области управления - CONTROL REGION LENGTH
REGISTER (P1LR)

Регистр длины системы - SYSTEM LENGHT REGISTER (SLR)

Регистр идентификации системы - SYSTEM IDENTIFICATION
REGISTER

Регистр команд и состояния - CONTROL/STATUS REGISTER (CSR)

Регистр конфигурации - CONFIGURATION REGISTER

Регистр процессора - PROCESSOR REGISTER

Регистр устройства - DEVICE REGISTER

Регистратор ошибок - ERROR LOGGER

Регистрационный монитор - LOGGING MONITOR

Регистрация - LOGGING

Регистрирующая запись - LOG

Регистровый режим - REGISTER MODE

Регистры возврата значений - VALUE RETURN REGISTERS

Редактор - EDITOR

Редактирующие символы, режим редактирования - CHARACTER
EDITING

Реентерабельный код - REENTRANT CODE

Режим адресации - ADDRESSING MODE

Режим доступа - ACCESS MODE

Режим захвата - LOCK MODE

Режим изменения - CHANGE MODE

Режим исключительной ситуации по ошибке - FAILURE EXCEPTION
MODE

Режим обмена - SWAP MODE

Режим ожидания ресурса - RESOURCE WAIT MODE

Режим пересылки - MOVE MODE

Режим перехода - BRANCH MODE

Режим пользователя - USER MODE

Режим с автоувеличением - AUTOINCREMENT MODE

Режим совместимости - COMPATIBILITY MODE

Режим со смещением - DISPLACEMENT MODE

Режим супервизора - SUPERVISOR MODE

Режим указания - LOCATE MODE

Режим управления - EXECUTIVE MODE

Режим ядра - KERNEL MODE

Ресурс - RESOURCE

Сбой - FAULT

Сегмент ввода-вывода образа - IMAGE I/O SEGMENT

00152-01 31 01

Сегмент ввода-вывода процесса - PROCESS I/O SEGMENT

Секция - SECTION

Секция образа - IMAGE SECTION (ISECT)

Секция процесса - PROCESS SECTION

Секция файла - FILE SECTION

Сетевая задача - NETWORK TASK

Сетевой командный терминал - REMOTE COMMAND TERMINAL

Сетевой объект - NETWORK OBJECT

Сеть - NETWORK

Сеть с коммутацией пакетов данных - PACKET SWITCHING DATA
NETWORK (PSDN)

Сжатие рабочего набора - SHRINK WORKING SET

Сигнал - SIGNAL

Сигнал о наложении в канале - COLLISION DETECT

Сигнал передачи - CARRIER SENSE

Сигнал передачи при параллельном доступе с определением
наложения в канале - CARRIER SENSE, MULTIPLE
ACCESS WITH COLLISION
DETECT (CSMA/CD)

Сильная ссылка - STRONG REFERENCE

Сильное опеределение - STRONG DEFINITION

Симбионт - SYMBIONT

Символ - CHARACTER

Символ обращения - WILDCARD CHARACTER

Символ "А" - "A" CHARACTER

Символ определения функций клавиатуры - GOLD KEY

Символ продолжения - CONTINUATION CHARACTER

00152-01 31 01

Символьная строка - CHARACTER STRING

Символьный код кои-8 - AMERICAN STANDARD CODE FOR
INFORMATION INTERCHANGE (ASCII)

Символический отладчик - SYMBOLIC DEBUGGER

Символическое имя - SYMBOL

Символы редактирования, символьное редактирование -
CHARACTER EDITING

Синтаксис - SYNTAX

Синтаксический Анализ - PARSING

Синхронная операция над записью - SYNCHRONOUS RECORD
OPERATION

Синхронное раз'единение - SYNCHRONOUS DISCONNECT

Система - SYSTEM

Система управления данными (СУД-32) - RECORD MANAGEMENT
SERVICES

Система управления файлами - FCS

Системная динамическая память - SYSTEM DYNAMIC MEMORY

Системная очередь - SYSTEM QUEUE

Системное виртуальное пространство - SYSTEM VIRTUAL SPACE

Системное обслуживание, программа системного обслуживания -
SYSTEM SERVICES

Системное устройство - SYSTEM DEVICE

Системный буферизованный ввод-вывод - SYSTEM BUFFERED I/O

Системный виртуальный адрес - SYSTEM VIRTUAL ADDRESS (SVA)

Системный образ - SYSTEM IMAGE

Системный отладчик - XDELTA

Системный пароль - SYSTEM PASSWORD

Системный программист - SYSTEM PROGRAMMER

Системный процесс - SYSTEM PROCESS

Системо-определенный идентификатор - SYSTEM-DEFINED
IDENTIFIER

Сквозная запись - WRITE THROUGH

Сквозное хранение - STORE THROUGH

Скрытое устройство - CONCEALED DEVICE

Слабая ссылка - WEAK REFERENCE

Слабое определение - WEAK DEFINITION

Слово - WORD

Слово состояния процессора - PROCESSOR STATUS WORD (PSW)

Слоты для заголовков процессов - PROCESS HEADER SLOTS

Смежный узел - ADJACENT NODE

Смещение - OFFSET

Собственный блок тома - HOME BLOCK

Событие - EVENT

Соединительный узел - CONNECTOR NODE

Сокращенный пакет запросов - SRP

Сообщение прерывания - INTERRUPT MESSAGE

Сортировка - SORTING

Сортирующая последовательность - COLLATING SEQUENCE

Состояние - STATUS

Состояние неконтролируемой работы магнитной ленты - RUNAWAY
TAPE CONDITION

Сохранение - BACKUP

Спецификатор базового операнда - BASE OPERAND SPECIFIER

Спецификатор задачи - TASK SPECIFIER

00152-01 31 01

Спецификатор операнда - OPERAND SPECIFIER

Спецификация диапазона - RANGE SPECIFICATION

Спецификация узла - NODE SPECIFICATION

Спецификация файла - FILE SPECIFICATION

Список поиска - SEARCH LIST

Список прав по идентификаторам - RIGHTS LIST

Список управления доступом - ACCESS CONTROL LIST (ACL)

"спячка" - HIBERNATION

Спул - SPOOLING

Средства межпроцессных связей - INTERPROCESS COMMUNICATION
FACILITY

Средства проверки функционирования системы - USER

ENVIRONMENT TEST PACKAGE (UETP)

Средство сборки/разборки пакетов - PACKET ASSEMBLY/

DISASSEMBLY (PAD) FACILITY

Срочное прерывание - URGENT INTERRUPT

Стек - STACK

Стек вызова - CALL STACK

Стек прерывания - INTERRUPT STACK (IS)

Стирание при выделении области - ERASE-ON-ALLOCATE

Стирание при удалении - ERASE-ON-DELETE

Стоимость, вес - COST

Стоимость маршрута - PATH COST

Страница - PAGE

Страничный отказ - PAGE FAULT

Строка - STRING

Строка поиска - SEARCH STRING

Строка со счетчиком - COUNTED STRING

Структура данных - DATA STRUCTURE

Супервизорный режим - SUPERVISOR MODE

Сцепленная запись - SPANNED RECORD

Счетчики - COUNTERS

Счетчик инструкций - PROGRAM COUNTER (PC)

Таблица выбора функций - FUNCTION DECISION TABLE (FDT)

Таблица точек входа драйвера - DRIVER DISPATCH TABLE
(DDT)

Таблица глобальных символов - GLOBAL SYMBOL TABLE (GST)

Таблица глобальных страниц - GLOBAL PAGE TABLE

Таблица логических имен - LOGICAL NAME TABLE

Таблица пролога драйвера - DRIVER PROLOQUE TABLE (DPT)

Таблица символов отладчика - DST

Таблица страниц области программы - POPT

Таблица страниц области управления - P1PT

Таблица страниц системы - SYSTEM PAGE TABLE (SPT)

Таблицы страниц процесса - PROCESS PAGE TABLE

Тайм-аут - TIMEOUT

Таймер - TIMER

Текстовый буфер - TEXT BUFFER

Текущий режим доступа - CURRENT ACCESS MODE

Телезагрузка задачи - DOWNLINE TASK LOAD

Телезагрузка системы - DOWNLINE SYSTEM LOAD

Терминал - TERMINAL

00152-01 31 01

Терминал оператора защиты - SECURITY OPERATOR TERMINAL

Терминал с твердой копией - HARD-COPY TERMINAL

Тетрада - NIBBLE

Тип данных - DATE TYPE

Тип доступа - ACCESS TYPE

Тип доступа на запись - WRITE ACCESS TYPE

Тип доступа на модификацию - MODIFY ACCESS TYPE

Тип доступа на чтение - READ ACCESS TYPE

Тип доступа перехода - BRANCH ACCESS TYPE

Тип объекта - OBJECT TYPE

Тип события - EVENT TYPE

Тип спецификатора операнда - OPERAND SPECIFIER TYPE

Тип устройства - GENERIC DEVISE NAME

Тип файла - FILE TYPE

Том - VOLUME

Точка входа - ENTRY POINT

Транзитный участок - HOP

Трафарет стирания - ERASURE PATTERN

Уведомление о состоянии сети - NETWORK STATUS NOTIFICATIONS

Удаленное устройство - REMOTE DEVICE

Удаленное DTE - REMOTE DTE

Удаленный узел - REMOTE NODE

Узел - NODE

Узел. исполнитель - EXECUTOR NODE

Указатель аргумента - ARGUMENT POINTER (AP)

Указатель кадра - FRAME POINTER (FP)

Указатель стека - STACK POINTER (SP)
Указатель стека прерываний - INTERRUPT STACK POINTER (ISP)
Умолчание, подразумеваемый по умолчанию - DEFAULT
Универсальный символ - UNIVERSAL SYMBOL
Упакованная десятичная строка - PACKED DECIMAL STRING
Упакованный формат десятичных чисел - PACKED DECIMAL
Управление доступом - ACCESS CONTROL
Управление памятью - MEMORY MANAGEMENT
Управление симбионтами - SYMBIONT MANAGER
Управление учетом - ACCOUNTING MANAGER
Управляющая программа - EXECUTIVE
Управляющая программа совместимости - APPLICATION MIGRATION
EXECUTE (AME)
Управляющая станция - CONTROL STATION
Управляющий ключ - CONTROL KEY
Уровень асинхронного системного прерывания - ASYNCHRONOUS
SYSTEM TRAP LEVEL (ASTLVL)
Уровень приоритета прерываний - INTERRUPT PRIORITY LEVEL
(IPL)
Условие, исключительное состояние - CONDITION
Устройство - DEVICE
Устройство большой емкости - MASS STORAGE DEVICE
Устройство загрузки - LOAD DEVICE
Устройство управления, контроллер - DEVICE CONTROLLER
Учет, учетный раздел - ACCOUNT
Учетное имя - ACCOUNT NAME

Файл - FILE

Файл авторизации - AUTHORIZATION FILE

Файл авторизации пользователя - USER AUTHORIZATION FILE

Файл входа в систему - LOGIN FILE

Файл регистрации - LOGGING FILE

Файловая система - FILE SYSTEM

Файловая структура - FILE STRUCTURE (FILES-11)

Файловая структура уровня 1 - FILES-SM STRUCTURE LEVEL 1,

ON-DISK STRUCTURE

LEVEL 1 (ODS-1)

Файловая структура уровня 2 - FILES-SM STRUCTURE LEVEL 2,

ON-DISK STRUCTURE

LEVEL 2 (ODS-1)

Физическая память - PHYSICAL MEMORY

Физический адрес - PHYSICAL ADDRESS

Физический канал - PHYSICAL LINK

Фиксированная управляющая область - FIXED CONTROL AREA

Фиксированные номера строк - FIXED LINE NUMBERS

Флаг - FLAG

Флаг события - EVENT FLAG

Форк-блок - FORK BLOCK

Форк-диспетчер - FORK DISPATCHER

Форк-очередь - FORK QUEUE

Форк-процесс - FORK PROCESS

Форк-уровень драйвера - DRIVER FORK LEVEL

Формат записи - RECORD FORMAT

Формат записи переменной длины - VARIABLE-LENGTH RECORD

00152-01 31 01

FORMAT

Формат записи фиксированной длины - FIXED-LENGTH RECORD

FORMAT

Формат потока - STREAM RECORD FORMAT

Функции виртуального ввода-вывода - VIRTUAL I/O FUNCTION

Функции логического уровня ввода-вывода - LOGICAL I/O
FUNCTION

Функции физического уровня ввода-вывода - PHYSICAL I/O
FUNCTION

Функция ввода-вывода - I/O FUNCTION

Характеристики - CHARACTERISTICS

Центральный процессор - CENTRAL PROCESSING UNIT (CPU)

Цепь - CIRCUIT

Цепь в многоточечной системе - MULTIPOINT CIRCUIT

Цепь рассылки - BROADCAST CIRCUIT

Цепь точка-точка - POINT-TO-POINT CIRCUIT

Цилиндр - CYLINDER

Циклическое планирование, циклическая диспетчеризация -
ROUND ROBIN

Чередующаяся адресация - INTERLEAVING

Числовая строка - NUMERIC STRING

Число с плавающей запятой - FLOATING (POINT) DATA

Число с плавающей запятой D_формата - DOUBLE FLOATING DATA,
D_FLOATING POINT DATA

00152-01 31 01

Число с плавающей запятой G_формата - G_FLOATING DATA

Число с плавающей запятой F_формата - F_FLOATING POINT
DATA

Число с плавающей запятой H_формата - H_FLOATING POINT
DATA

Чистый код - PURE CODE

Чтение, запись, выполнение, удаление - RWED

Чужой том - FOREIGN VOLUME

Шаг квантования - QUANTUM

Ширина экрана - SCREEN WIDTH

Эквивалентное имя - EQUIVALENCE NAME

Экстент - EXTENT

Элемент, компонент - COMPONENT

Элемент сети, состояние - STATE

Элемент списка управления доступом - ACCESS CONTROL LIST
ENTRY (ACE)

Элемент таблицы страниц - PAGE TABLE ENTRY (PTE)

Эхо - ECHO

Язык ассемблера - ASSEMBLY LANGUAGE

Язык описания файлов - FILE DEFINITION LANGUAGE (FDL)

Ячейка для записи - RECORD CELL

2. СПИСОК ТЕРМИНОВ МОС ВП НА АНГЛИЙСКОМ ЯЗЫКЕ

"A" CHARACTER (символ "A")

- подмножество символов кода КОИ-8, которое допускается для использования в стандартных метках магнитных лент.

ABORT (аварийное завершение)

- исключительная ситуация, возникающая во время выполнения инструкции и иногда оставляющая регистры и память в неопределенном состоянии, которое может препятствовать повторному выполнению инструкции.

ABSOLUTE INDEXED MODE (абсолютный индексный режим)

- индексный режим адресации, в котором спецификатор базового операнда адресуется в абсолютном режиме.

ABSOLUTE MODE (абсолютный режим)

- косвенный режим с автоувеличением, где в качестве регистра используется счетчик инструкций (РС).

ABSOLUTE TIME (абсолютное время)

- значения, выражающие определенную дату (месяц, день, год) и время дня. Значения абсолютного времени представлены в системе положительными числами.

ACCESS CONTROL (управление доступом)

- проверка запросов на соединение, вход в систему или доступ к файлам с целью определения возможности приема таких запросов. Имя пользователя и пароль - это наиболее общие средства управления доступом.

ACCESS CONTROL LIST (ACL) (список управления доступом)

- список, определяющий виды доступа, которые необходи-

мо либо предоставить, либо нет пользователям некоторого объекта. Список управления доступом можно создать для таких объектов, как файлы, устройства и почтовые ящики. Каждый список управления доступом состоит из одного или нескольких элементов, называемых элементами списка управления доступом.

ACCESS CONTROL LIST ENTRY (ACE) (элемент списка управления доступом)

- элемент в списке управления доступом, определяющий права доступа к некоторому объекту. Списки управления доступом для каждого объекта могут содержать большое число элементов и ограничены лишь полным объемом памяти и соображениями, связанными с характеристиками функционирования.

ACCESS MODE (режим доступа)

- любой из четырех режимов доступа к процессору, в котором функционирует программное обеспечение. В порядке убывания привилегий и защиты - это следующие режимы доступа: ядра, управления, супервизора и пользователя. Когда процессор находится в режиме ядра, выполняющаяся программа имеет полный контроль над системой и несет за нее ответственность. В любом другом режиме процессор не может выполнять привилегированные инструкции. Поле текущего режима доступа содержится в длинном слове состояния процессора. См. также RECORD ACCESS MODE (метод доступа к записям).

00152-01 31 01

ACCESS TYPE (тип доступа)

- 1) способ обращения процессора к операндам инструкций. Существуют следующие типы доступа: на чтение, на запись, на модификация и переход.
- 2) способ обращения процедуры к своим аргументам.

ACCESS VIOLATION (нарушение доступа)

- попытка обратиться по адресу, который не отображен в виртуальной памяти или недоступен при текущем режиме доступа.

ACCOUNT (учет, учетный раздел, учетная информация)

- ключ в систему и единица системного учета. Каждый пользователь системы, включая саму систему, имеет учетный раздел. Администратор системы создает учетные разделы в главном каталоге файлов и приписывает учетный номер каждому пользователю. Отдельные пользователи имеют также учетное имя. Вход в систему осуществляется под конкретным учетным именем или номером. Этот номер информирует систему о расположении файлов пользователя и о том, какие типы доступа к другим файлам и системным средствам пользователю следует дать. Учетные разделы пользователей бывают как привилегированными, так и непривилегированными. Привилегированные пользователи имеют доступ ко всем командам и частям системы. Непривилегированные пользователи ограничены рамками обычных действий, не угрожающих целостности операционной системы.

00152-01 31 01

ACCOUNTING MANAGER (управление учетом)

- функция системного процесса, вызванного диспетчером заданий, которая заносит учетные записи в системный учетный файл регистрации с целью отслеживания активности задания и завершения пользовательских процессов, пакетных заданий, заданий печати и т.д.

ACCOUNT NAME (учетное имя)

- строка, которая определяет конкретный учетный раздел, служащий для накопления данных об использовании ресурсов заданий. Это имя представляет собой учетную величину расходов пользователя, а не его UIC.

ACTIVE COMPONENT (активный элемент)

- элемент сети с рабочим состоянием, отличным от OFF (выключено). Ключевое слово "активный" может быть использовано с командами SHOW или LIST программы NCP для отображения информации об активных линиях, цепях, узлах и регистрации.

ADAPTER CONTROL BLOCK (ADP) (блок управления адаптером)

- структура в базе данных ввода-вывода, которая описывает адаптер общей шины.

ADDRESS (адрес)

- число, используемое операционной системой и пользовательскими программами для идентификации ячейки памяти. См. также VIRTUAL ADDRESS (виртуальный адрес) и PHYSICAL ADDRESS (физический адрес).

ADDRESS ACCESS TYPE (адресный тип доступа)

- тип доступа, при котором указанный в инструкции опе-

ранд непосредственно ей недоступен. Фактическим операндом инструкции является адрес указанного операнда. Контекст вычисления адреса задается типом операнда.

ADDRESS SPACE (адресное пространство)

- множество всех возможных адресов, доступных процессу. Виртуальное адресное пространство связано с множеством всех возможных виртуальных адресов. Физическое адресное пространство связано с множеством всех возможных физических адресов.

ADDRESSING MODE (режим адресации)

- способ определения операнда; существует семь основных режимов адресации: регистровый, косвенно-регистровый, с автоувеличением, косвенный с автоувеличением, с автоуменьшением, со смещением и косвенный со смещением. Кроме того, существуют шесть индексных режимов адресации, использующих два общих регистра, и литеральный режим адресации. Три режима адресации через РС имеют названия: непосредственный (для регистрового режима, отличного от режима, использующего РС), абсолютный (для косвенного режима с автоувеличением, использующего РС) и режим адресации для команд перехода.

ADJACENT NODE (смежный узел)

- узел сети, удаленный от локального узла и соединенный с ним единственной физической линией.

ALARM (аварийный сигнал)

- см. SECURITY ALARM (аварийный сигнал защиты).

ALLOCATE A DEVICE (закрепить устройство)

- зарезервировать конкретное устройство для монопольного использования. Устройство может быть закреплено за некоторым пользовательским процессом только тогда, когда оно не закреплено ни за каким другим процессом.

ALLOCATION CLASS (класс распределения)

- уникальное число, в интервале от 0 до 255, приписанное администратором системы паре главных вычислительных машин и двухканальным устройствам, которые эти машины делают доступными для других узлов кластера. Класс распределения дает пользователям способ обращения к двухканальным устройствам через любую из главных вычислительных машин. Если при таком способе одна из главных вычислительных машин недоступна, то пользователь может получить доступ к устройству, определенному этим классом, через другую главную вычислительную машину данного класса.

ALPHANUMERIC CHARACTER (буквенно-цифровой символ)

- большая или малая буква, знак #, подчеркивание или десятичная цифра.

ALPHANUMERIC UIC (буквенно-цифровой код идентификации пользователя)

- формат кода идентификации пользователя (UIC), который определяет пользовательскую группу и члена груп-

00152-01 31 01

пы не в числовой, а в буквенно-цифровой форме.

ALTERNATE KEY (альтернативный ключ, вторичный ключ)

- необязательный ключ в записях данных индексного файла; используется СУД-32 для построения альтернативного индекса. См. (KEY INDEXED FILE) (ключ индексного файла) и PRIMARY KEY (первичный ключ).

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII)
(код КОИ-8)

- набор 8-битовых двоичных чисел, представляющих символы алфавита, пунктуации, цифры и другие специальные символы, используемые в представлении текстов и в протоколе связей.

ANCILLARY CONTROL PROCESS (ACP) (вспомогательный управляющий процесс)

процесс, который действует как интерфейс между пользовательской программой и драйвером ввода-вывода. АСР выполняет функции управления файлами и каталогами. Тремя примерами АСР являются: АСР файловой структуры, АСР магнитной ленты и АСР сети.

APPLICATION MIGRATION EXECUTE (AME) (управляющая программа совместимости)

- часть МОС ВП, которая позволяет задачам ОС РВ выполняться под управлением МОС ВП в режиме совместимости с ОС РВ.

AREA (область, регион)

- 1) зона индексного файла, поддерживаемая СУД-32,

00152-01 31 01

которая позволяет пользователю задать размещение и/или специфичные размеры бакета для конкретных частей файла. Область содержит произвольное число бакетов, в файле может быть от 1 до 255 областей.

- 2) группа узлов в сети, которая может функционировать независимо, как подсеть.

AREA ROUTER (маршрутный узел региона)

- маршрутный узел 1 уровня.

AREA ROUTING (маршрутизация по регионам)

- метод группирования узлов некоторой сети в регионы с целью маршрутизации. Маршрутизация в сети со множеством регионов является иерархической: первый уровень - это маршрутизация внутри региона (маршрутизация 1 уровня) и второй, более высокий уровень - между регионами (маршрутизация 2 уровня). Маршрутизация по регионам позволяет поддерживать конфигурацию очень больших сетей.

ARGUMENT POINTER (AP) (указатель аргумента)

- общий регистр 12 (R12). По соглашению, AP содержит адрес базы списка аргументов для процедур, инициированных с использованием инструкций CALL.

ASSEMBLER (АССЕМБЛЕР)

- языковой процессор, который транслирует исходную программу, содержащую директивы языка АССЕМБЛЕРА и машинные инструкции, в объектный модуль.

ASSEMBLY LANGUAGE (язык АССЕМБЛЕРА)

- машинно-ориентированный язык программирования. Язы-

00152-01 31 01

ком АССЕМБЛЕРА для вычислительного комплекса СМ 1700 является язык МАКРОАССЕМБЛЕРА.

ASSIGN A CHANNEL (назначить канал)

- установить необходимую программную связь между пользовательским процессом и устройством для того, чтобы разрешить данному процессу взаимодействовать с этим устройством.

ASSIGNMENT STATEMENT (оператор присваивания)

- определение символического имени с целью его использования вместо символьных строк или числового значения. Символические имена могут определять синонимы для команд системы или могут использоваться в качестве переменных в командных процедурах.

ASYNCHRONOUS RECORD OPERATION (асинхронная операция над записью)

- режим обработки записи, в котором пользовательская программа может продолжить свое выполнение, не ожидая конца обслуживания выданного ей запроса на поиск или занесение записи.

ASYNCHRONOUS SYSTEM TRAP (AST) (асинхронное системное прерывание)

- программно-моделируемое прерывание с передачей управления программе, определенной пользователем. AST позволяет процессу пользователя получить сведения о происшедшем конкретном событии асинхронно по отношению к другому пользовательскому процессу. Если

00152-01 31 01

первый процесс определил программу AST для некоторого события, то система прерывает второй процесс и выполняет программу AST, когда это событие происходит. При завершении программы AST, система возобновляет выполнение второго процесса с того места, где он был прерван.

ASYNCHRONOUS SYSTEM TRAP LEVEL (ASTLVL) (уровень асинхронного системного прерывания)

- значение, хранящееся во внутреннем регистре процессора (ASTLVL), определяющее наиболее привилегированный режим доступа, до которого задерживается AST. AST не произойдет до тех пор, пока привилегия текущего режима доступа не понизится (увеличится в числовом значении) до значения большего или равного ASTLVL. Таким образом, AST для некоторого режима доступа не будет обслужено, пока процессор работает в более привилегированном режиме.

ATTRIBUTE (атрибут)

- в контексте защиты, атрибут - это поле информации в базе данных прав по идентификаторам, определяющее характеристики, которые могут быть предоставлены держателю идентификатора.

AUDITING (контроль)

- действие по регистрации появления некоторого события, связанного с защитой.

AUTHENTICATION (опознавание)

- действие по идентификации пользователей в начале их

00152-01 31 01

работы в системе. Мос вп (и большинство других операционных систем) использует пароль в качестве основного средства опознавания.

AUTHORIZATION FILE (файл авторизации)

- см. USER AUTORIZATION FILE (файл авторизации пользователя).

AUTODECREMENT INDEXED MODE (индексный режим с автоуменьшением)

- индексный режим адресации, при котором спецификатор базового операнда использует адресацию режима с автоуменьшением.

AUTODECREMENT MODE (режим с автоуменьшением)

- режим адресации, в котором содержимое выбранного регистра уменьшается, и результат используется как адрес фактического операнда для инструкции. Содержимое регистра уменьшается в соответствии с контекстом типа данных этого регистра: на 1 - для байта, на 2 - для слова, на 4 - для длинного слова и числа с плавающей запятой F_формата, на 8 - для квадрослова и числа с плавающей запятой D_формата.

AUTOINCREMENT DEFERRED INDEXED MODE (косвенный индексный режим с автоувеличением)

- индексный режим адресации, в котором спецификатор базового операнда использует адресацию косвенного режима с автоувеличением.

AUTOINCREMENT DEFERRED MODE (косвенный режим с автоувеличением)

- режим адресации, при котором указанный регистр содержит адрес длинного слова с адресом фактического операнда. Содержимое регистра увеличивается на 4 (число байтов в длинном слове). Если в качестве регистра используется *rs*, то этот режим называется абсолютным режимом.

AUTOINCREMENT INDEXED MODE (индексный режим с автоувеличением)

- индексный режим адресации, в котором спецификатор базового операнда использует адресацию режима с автоувеличением.

AUTOINCREMENT MODE (режим с автоувеличением)

- режим адресации, в котором содержимое специфицированного регистра используется как адрес операнда, а затем содержимое регистра увеличивается на размер этого операнда.

AUTOMATIC RECORD LOCKING (автоматический захват записи)

- возможность СУД-32, которая позволяет пользователю захватывать в данное время только одну запись в указанном разделяемом файле. Захват происходит при каждом исполнении макрокоманд *PFIND* и *PGET* (если не установлен бит *NLK* в поле обработки записи). Захват снимается при обращении к следующей записи, удалении или корректировке текущей записи, откреплении потока записей или закрытии файла. Во время восстановления

00152-01 31 01

блока СУД-32 сохраняет захват модифицированной записи.

BACKING STORE (вспомогательная память)

- область адресов во внешней памяти, где содержатся данные.

BACKPLANE INTERCONNECT (внутренняя шина процессора)

- внутрипроцессорная шина, используемая адаптером общей шины, центральным процессором и контроллером памяти для передачи данных и прерываний.

BALANCE SET (балансный набор)

- множество всех рабочих наборов процессов, размещенных на данный момент в физической памяти. Требования на память, поступающие от тех процессов, рабочие наборы которых находятся в балансном наборе, уравниваются доступной памятью. Балансный набор обслуживается системным процессом, называемым планировщиком памяти.

BANDWIDTH (плоска пропускания)

- диапазон частот, предназначенный для канала или системы, т.е. отличие в герцах между самой высокой и самой низкой частотами диапазона.

BASE OPERAND ADDRESS (базовый адрес операнда)

- адрес базы некоторой таблицы или массива при адресации индексного типа.

BASE OPERAND SPECIFIER (спецификатор базового операнда)

- регистр, используемый для вычисления базового адреса операнда некоторой таблицы или массива при адресации

индексного типа.

BASE PRIORITY (базовый приоритет)

- приоритет, который система назначает процессу при его создании. Обычно базовый приоритет берется из файла авторизации пользователя. Планировщик никогда не устанавливает приоритет процесса ниже его базового приоритета. Базовый приоритет может быть изменен только системным администратором или самим процессом. Базовый приоритет выполняющегося процесса может быть изменен любым пользователем, имеющим привилегию ALTPRI.

BASE REGISTER (базовый регистр)

- общий регистр, используемый для хранения адреса первого элемента в списке, таблице, массиве или другой структуре данных.

BATCH PROCESSING (пакетная обработка)

- режим обработки, при котором все команды, предназначенные для выполнения операционной системой (и, возможно, данные, которые должны использоваться в качестве входных данных команд), помещаются в некоторый файл или набиваются на перфокартах и представляются системе для обработки. См. Для сравнения термин COMMAND PROCEDURE (командная процедура).

BILATERAL CLOSED USER GROUP (BCUG) (двусторонняя замкнутая группа пользователей)

- необязательное средство в сети с коммутацией пакетов данных, которое ограничивает пару DTE при связи друг

с другом.

BINDING (компоновка)

- см. LINKING (компоновка).

BIT STRING (битовая строка)

- см. VARIABLE-LENGTH BIT FIELD (битовое поле переменной длины).

BLOCK (блок)

- 1) наименьшая логически адресуемая единица данных, которую определенное устройство может передать за одну операцию ввода-вывода (512 последовательных байтов для большинства дисковых устройств).
- 2) произвольное число непрерывных байтов, используемых для хранения логически связанных данных о состоянии, управлении и другой обрабатываемой информации.

BLOCK I/O (ввод-вывод блоков)

- способ доступа к данным, при котором программа обрабатывает составляющие файл блоки (физические записи), а не его логические записи; допускается прямой доступ к блокам в файле без учета организации файла или формата записей.

BLOCKING AST (AST по захвату)

- AST, которое может быть запрошено процессом, использующим программы системного обслуживания управления захватом. AST по захвату передается запрашивающему процессу, если он препятствует другому процессу получить доступ к ресурсу.

BOOT (загрузка)

- сокращение термина "начальная загрузка". Подпрограмма или аппаратная процедура, позволяющая устройству самостоятельно осуществлять установку в заданное состояние, например, стандартная программа, несколько первых инструкций которой позволяют осуществлять занесение оставшейся части этой программы в память с устройства ввода. Загрузкой называется и занесение в память копии операционной системы.

BOOT NAME (имя устройства загрузки)

- сокращенное имя устройства, используемого для загрузки программ. Например, DUO является таким именем для устройства DUA0.

BOOTSTRAP BLOCK (блок начальной загрузки)

- блок в индексном файле на системном диске. Он может содержать программу, которая загружает операционную систему в память.

BRANCH ACCESS TYPE (тип доступа перехода)

- атрибут инструкции, который указывает, что процессор не обращается к адресу операнда, поскольку операнд - это смещение перехода; размер смещения перехода задается типом данных операнда.

BRANCH MODE (режим перехода)

- режим адресации, при котором спецификатор операнда инструкции является смещением со знаком размером в байт или слово. Смещение добавляется к содержимому скорректированного РС, который представляет собой

адрес первого байта за смещением, и результат дает адрес перехода.

BREACH (нарушение защиты)

- нарушение в защите системы, приводящее к доступу пользователя или программы к некоторому объекту.

BREAKIN ATTEMPT (попытка незаконного доступа)

- попытка неавторизованного в системе пользователя получить доступ к системе. Поскольку первый доступ к системе осуществляется через вход в систему, то попытка незаконного доступа относится к попыткам входа в систему недопозволенным способом.

BROADCAST ADDRESSING (адресация рассылки)

- специальный тип многопунктовой адресации, при которой получать сообщение должны все узлы.

BROADCAST CIRCUIT (цепь рассылки)

- цепь, на которой объединено множество узлов и в которой существует метод передачи сообщения, принимаемого многими получателями.

BUCKET (бакет)

- структура памяти, содержащая от 1 до 32 блоков и используемая для построения и обработки файлов с относительной или индексной организацией. Бакет содержит одну или несколько записей или ячеек для записей. Бакет представляет собой единицу непрерывной передачи информации между буфером СУД-32 и диском.

BUCKET LOKING (захват бакетов)

- средство, которое препятствует доступу к любой записи в бакете более чем одному пользователю до тех пор, пока этот пользователь не освободит бакет.

BUCKET SPLIT (расщепление бакета)

- результат внесения записей в полный бакет. Для минимизации расщепления бакета СУД-32 пытается сохранить половину записей в первоначальном бакете и передать оставшиеся записи во вновь созданный бакет.

BUFFER (буфер)

- внутренняя область памяти, используемая для временного хранения записей при выполнении операций ввода или вывода.

BUFFERED DATA PATCH (буферизованный путь данных)

- путь данных в адаптере общей шины, который передает 32-х или 64-х битные данные за одну передачу.

BUFFERED I/O (буферизованный ввод-вывод)

- см. SYSTEM BUFFERED I/O (системный буферизованный ввод-вывод).

BUGCHECK (внутренняя диагностическая проверка)

- программа операционной системы для внутренней диагностической проверки. Система регистрирует ошибку и дает аварийный отказ.

BYTE (байт)

- восемь смежных битов, начинающихся на границе адресуемого байта. Биты нумеруются справа налево от 0 до 7. БИТ 0 - младший. Если байт интерпретируется ариф-

00152-01 31 01

метически, то он представляет собой целое число в дополнительном коде, вес битов увеличивается от 0 к 6. Бит 7 является знаковым. значение целого числа со знаком находится в диапазоне от -128 до 127 (десятичное). Если байт интерпретируется как целое без знака, то вес битов увеличивается от 0 к 7, и значение целого без знака находится в диапазоне от 0 до 255 (десятичное). Байт можно использовать для хранения одного символа в коде КОИ-8.

CACHE MEMORY (кэш-память)

- небольшая высокоскоростная память, размещаемая между медленной основной памятью и процессором. Кэш-память увеличивает эффективную скорость обмена с памятью и скорость процессора. Она содержит копии данных, использованных процессором последними, и выбирает несколько байтов данных из памяти, предполагая, что процессор будет обращаться к следующей по порядку серии байтов.

CALL (вызов)

- передача управления указанной программе.

CALL FRAME (кадр вызова)

- см. STACK FRAME (кадр стека).

CALL INSTRUCTIONS (инструкции вызова)

- инструкции процессора CALLG (вызов процедуры общего вида) и CALLS (вызов процедуры стекового вида).

CALL STACK (стек вызова)

- стек и обычная стековая структура, используемые при

00152-01 31 01

вызове процедуры. Каждый режим доступа имеет по одному стеку вызова для каждого контекста процесса, и один стек вызова имеется в контексте обработки прерываний.

CAPTIVE ACCOUNTS (ограничивающий учетный раздел)

- тип учетной информации МОС ВП, которая лимитирует активность пользователя. Обычно пользователь ограничен в применении некоторых командных процедур и/или команд. Пользователя может быть не разрешено применение ключа УС/У. (Этот тип учетной информации является синонимом TURNKEY ACCOUNT - ограничивающий учетный раздел или TIED ACCOUNT - ограничивающий учетный раздел).

CARRIER SENSE (сигнал передачи)

- сигнал, выдаваемый на физическом уровне систем "трал" и "магистр" для указания того, что одна или более станций (узлов) осуществляют в данный момент передачу информации в магистраль ETHERNET.

CARRIER SENSE, MULTIPLE ACCESS WITH COLLISION DETECT (CSMA/CD) (сигнал передачи при параллельном доступе с определением наложения в канале)

- процедура управления каналом, используемая магистралью ETHERNET. Позволяет множеству станций обращаться к каналу рассылки произвольным образом; исключает конфликтные ситуации при помощи сигнала передачи и задержки, разрешает конфликтные ситуации при помощи выявления наложения в канале и повторной

передачи данных.

CENTRAL PROCESSING UNIT (CPU) (центральный процессор)

- центральный процессор представляет собой аппаратуру, которая осуществляет управление, счет, назначение трактов передачи информации, а также выполнение программ, короче говоря, центральный процессор является той частью эвм, которая фактически производит вычисления.

CHANNEL (канал)

- 1) логический путь, соединяющий процесс пользователя с физическим устройством. Процесс пользователя просит операционную систему назначить канал к устройству для связи с этим устройством. См. Также **CONTROLLER DATA CHANNEL (канал данных контроллера)**;
- 2) средства передачи по сети с коммутацией пакетов данных. Каждый канал определяется уникальным эталонным номером, называемым номером логического канала (LCN).

CHANNEL CONTROL BLOCK (CCB) (блок управления каналом)

- структура в базе данных ввода-вывода, созданная с помощью программы системного обслуживания "назначить канал ввода-вывода" для описания того устройства, которому назначен канал.

CHANNEL REQUEST BLOCK (CRB) (блок запросов канала)

- структура в базе данных ввода-вывода, которая описывает действия конкретного контроллера. Блок запросов канала некоторого контроллера содержит указатели на

00152-01 31 01

очередь ожидания драйверов, готовых обратиться к устройству через этот контроллер;

CHARACTER (символ)

- символ, представленный в коде КОИ-8. См. Также ALPHANUMERIC CHARACTER (буквенно-цифровой символ).

CHARACTER BUFFER (буфер символов)

- временная область памяти, которая используется для хранения последнего символа, удаленного операцией стирания символов редактора EDT.

CHARACTER EDITING (символы редактирования, символьное редактирование)

- редактирование текста на символьном уровне; режим EDT, введенный командой CHANGE.

CHARACTER STRING (символьная строка)

- непрерывный набор байтов. Символьная строка определяется двумя атрибутами: адресом и длиной. Адрес строки является адресом байта, содержащего первый символ строки. Последующие символы хранятся в байтах с возрастающими адресами. Длина - это число символов в строке.

CHARACTER STRING DESCRIPTOR (дескриптор символьной строки)

- структура данных размером в квадрослово, используемая для передачи символьных данных (строк). Первое слово содержит длину символьной строки. Второе слово может содержать информацию о типе. Оставшееся длинное слово содержит адрес строки.

CHARACTERISTICS (характеристики)

- тип отображения для команд SHOW и LIST программы NCP. Относится к статической информации об элементе, содержащемся или в непостоянных или в постоянных базах данных. Такая информация может включать параметры, определенные для данного элемента командами SET или DEFINE.

CIRCUIT (цепь)

- виртуальные пути связи между узлами или оборудованием DTE. Цепи выполняют действия на физических линиях и представляют собой среду, на которой происходят все операции ввода-вывода. Цепи X.25 являются виртуальными цепями.

CLOSED USER GROUP (CUG) (замкнутая группа пользователей)

- необязательное средство в сети с коммутацией пакетов данных, которое заключает два или несколько DTE в одну группу для связи друг с другом. Базовый CUG, кроме того, препятствует связи этих DTE с другими DTE вне группы. Дополнения к средствам базового CUG делают возможной связь одного или нескольких DTE с другими DTE вне этой группы.

CLUSTER (кластер)

- 1) набор смежных блоков, который является основной единицей распределения пространства на дисковом томе с файловой структурой.
- 2) набор страниц, загружаемый в память за одну операцию обмена страниц.

00152-01 31 01

- 3) кластер флагов событий.
- 4) конфигурация процессоров ВК СМ 1700.

COLLATING SEQUENCE (сортирующая последовательность)

- порядок, приписанный символам в некотором наборе символов (например, в коде КОИ-8, в многонациональном наборе символов или в коде дкой), и используемый в целях задания последовательности символов.

COLLISION (наложение в канале; конфликтная ситуация)

- наложение множества передаваемых данных в физическом канале сети, приводящее к искажению данных и необходимости повторной передачи.

COLLISION DETECT (сигнал о наложении в канале)

- сигнал, выдаваемый на физическом уровне систем "трал" или "магистр" в уровень их канала данных для указания того, что одна или несколько станций (узлов) конкурируют с передачей информации локальной станцией.

COMMAND (команда)

- инструкция, вводимая пользователем с терминала или включаемая в командную процедуру.

COMMAND FILE (командный файл)

- см. COMMAND PROCEDURE (командная процедура).

COMMAND INTERPRETER (интерпретатор команд)

- системная программа, состоящая из множества процедур, которая выполняется в режиме супервизора в контексте процесса с целью синтаксической проверки и анализа команд, вводимых пользователем с терминала

или представленных в командном файле.

COMMAND LEVEL (командный уровень)

- определяет входной поток для интерпретатора команд. Первоначальный входной поток всегда имеет нулевой командный уровень. Каждое последующее выполнение некоторой процедуры изменяет командный уровень.

COMMAND NODE (командный узел)

- узел, с которого вводится команда программы NCP.

COMMAND PARAMETER (параметр команды)

- ограниченный пробелами позиционный операнд команды, например, спецификация файла, возможность или константа.

COMMAND PROCEDURE (командная процедура)

- файл, содержащий команды и данные, которые могут поступать к интерпретатору вместо команд, вводимых поочередно с терминала. Таким образом, командные процедуры предоставляют средство для автоматического выполнения команд в операционной системе. Кроме того, они позволяют пользователям применять такие методы программирования, как циклы, счётчики, метки и подстановки символов с целью построения сложных последовательностей команд, которые могут быть изменены при взаимодействии с пользователем.

COMMAND STRING (командная строка)

- строка или набор последовательных строк, содержащих команду и, необязательно, информацию, модифицирующую команду. Полная командная строка состоит из команды,

ее необязательных квалификаторов, параметров (например, спецификации файлов), и их квалификаторов. Командная строка обычно заканчивается нажатием клавиши <BK>.

COMMON EVENT FLAG CLUSTER (кластер общих флагов событий)

- набор из 32-х флагов событий, который позволяет взаимодействующим процессам посылать друг другу уведомления о событиях. Кластеры общих флагов событий создаются по мере необходимости. Процесс может связываться не более чем с двумя кластерами общих флагов событий.

COMPATIBILITY MODE (режим совместимости)

- режим работы, который позволяет центральному процессору выполнять непривилегированные инструкции 16-разрядных см эвм с магистральной архитектурой. Мос вп обеспечивает режим совместимости, предоставляя для образа задачи ос рв среду функционирования этой операционной системы. Процедуры режима совместимости МОС ВП перехватывают вызовы управляющей программы ос рв и преобразуют их в соответствующие функции МОС ВП.

COMPONENT (элемент, компонент)

- контролируемый и управляемый элемент сети. Элементы включают линии, цепи, узлы, модули, регистраторы и объекты. Элементы образуют часть синтаксиса команд программы НСР.

CONCEALED DEVICE (скрытое устройство)

- устройство ввода-вывода, которое имеет связанное с ним логическое имя; пользователь поэтому видит не имя устройства, а это логическое имя, распечатываемое в большинстве системных сообщений.

CONDITION (условие, исключительное состояние)

- состояние ошибки, существующее при возникновении исключительной ситуации. См. EXCEPTION (исключительная ситуация) и CONDITION HANDLER (программа обработки исключительного состояния).

CONDITION CODES (коды условий)

- четыре бита в слове состояния процессора, которые отображают результаты последних выполненных инструкций.

CONDITION HANDLER (программа обработки исключительного состояния)

- программа, выполняемая МОС ВП при возникновении исключительной ситуации. При ее возникновении МОС ВП ищет программу обработки исключительного состояния и, если находит, сразу же инициирует ее. Эта программа может выполнить некоторое действие, чтобы изменить состояние, которое вызвано исключительной ситуацией, и продолжить выполнение процесса, приведшего к исключительной ситуации. Программы обработки исключительных состояний выполняются в контексте процесса в режиме доступа программы, которая вызвала исключительную ситуацию.

CONDITION VALUE (значение исключительного состояния)

- 32-битная величина, которая однозначно определяет исключительную ситуацию, вызвавшую данное исключительное состояние.

CONFIGURATION DATABASE (база данных конфигурации)

- база данных, содержащая файлы, которые дают информацию об элементах сети. Эти файлы содержат информацию о локальном узле и обо всех узлах, модулях, цепях, линиях, регистраторах и объектах в сети. См. Также PERMANENT DATABASE (постоянная база данных) и VOLATILE DATABASE (непостоянная база данных).

CONFIGURATION REGISTER (регистр конфигурации)

- регистр команд и состояния для некоторого адаптера, например, адаптера общей шины. Он находится в пространстве ввода-вывода адаптера.

CONGESTION LOSS (потеря из-за переполнения)

- исключительное состояние, при котором пакеты данных, передаваемые по сети, теряются, когда маршрутизирующий уровень системы "трал" не в состоянии поместить их в буфер.

CONNECT-TO-INTERRUPT (подключение к вектору прерывания)

- функция, подключающая процесс к вектору прерывания устройства. Для подключения к вектору прерывания процесс должен создать карту пространства ввода-вывода программы, содержащей этот вектор. См. PAGE FRAME NUMBER MAPPING (отображение номера физической страницы).

CONNECTOR NODE (соединительный узел)

- узел, который работает как шлюз интерфейса X.25 с тем, чтобы дать возможность множеству узлов МДС ВП обращаться к сети с коммутацией пакетов данных.

CONSOLE (консоль)

- блок ручного управления, объединенный в одно целое с центральным процессором. Консоль включает последовательный интерфейс шины, подсоединенный к терминалу с твердой копией, который позволяет оператору запускать и останавливать систему, управлять функционированием системы и выполнять диагностические работы.

CONSOLE TERMINAL (консольный терминал)

- терминал с твердой копией, подсоединенный к консоли центрального процессора.

CONTEXT (контекст)

- среда работы. См. Также PROCESS CONTEXT (контекст процесса), HARDWARE CONTEXT (аппаратный контекст) и SOFTWARE CONTEXT (программный контекст).

CONTEXT INDEXING (контекстная индексация)

- способность автоматически индексировать структуру данных, поскольку размер данных известен. Используется для определения фактора смещения.

CONTEXT SWITCHING (переключение контекста)

- прерывание выполняющейся работы и переключение на другую работу. Переключение контекста происходит как назначение на выполнение одного процесса после другого. Операционная система сохраняет аппаратный кон-

00152-01 31 01

текст прерванного процесса в его аппаратном блоке рса, используя инструкцию "сохранить контекст процесса"; загружает в аппаратный контекст аппаратный РСВ другого процесса, используя инструкцию "загрузить контекст процесса"; назначает этот процесс на выполнение.

CONTIGUOUS AREA (непрерывная область)

- группа физически смежных блоков.

CONTINUATION CHARACTER (символ продолжения)

- знак минус в конце командной строки, разрешающий пользователю продолжить командную строку на следующую строку после нажатия клавиши <BK>.

CONTROL KEY (управляющий ключ)

- символ клавиатуры, который вызывает управляющее действие. Управляющий ключ - это, обычно, комбинация символа УС и алфавитного символа, например УС/У.

CONTROL REGION (P1) (область управления)

- половина пространства процесса со старшими адресами (область P1). Виртуальные адреса области управления относятся к связанной с процессом информации, которая используется системой для управления процессом. Эта информация содержит, в частности, стек ядра, стек режима управления, стек супервизора, постоянные каналы ввода-вывода, векторы исключительных ситуаций и динамически используемые системные процедуры (такие, как процедуры интерпретатора команд). Как правило, стек пользователя также находится в области

управления.

CONTROL REGION BASE REGISTER (P1BR) (базовый регистр области управления)

- регистр процессора или его эквивалент в аппаратном блоке управления процессом, который содержит базовый виртуальный адрес таблицы страниц области управления процессом.

CONTROL REGION LENGTH REGISTER (P1LR) (регистр длины области управления)

- регистр процессора или его эквивалент в аппаратном блоке управления процессом, который содержит число несуществующих элементов таблицы страниц для виртуальных страниц в области управления процессом.

CONTROL STATION (управляющая станция)

- узел сети на управляющем конце многоточечной цепи. Управляющая станция осуществляет управление подчиненными станциями этой цепи.

CONTROL/STATUS REGISTER (CSR) (регистр команд и состояния)

- регистр команд и состояния для некоторого устройства или контроллера. Размещается в пространстве ввода-вывода процессора.

CONTROLLER DATA CHANNEL (канал данных контроллера)

- логический путь, к которому должен быть дан доступ драйверу некоторого устройства мультиплексного контроллера, прежде чем драйвер активизирует устройство.

00152-01 31 01

COOPERATING TASKS (взаимосвязанные задачи)

- две задачи, которые взаимодействуют друг с другом в режиме позадачной связи. В частности, взаимосвязанные задачи должны принять соглашение относительно посылаемых данных пользователей, а именно, как они будут передавать и принимать сообщения, чтобы обеспечить одну передачу информации на каждый прием, и какая задача будет раз'единять линию.

COPY-ON-REFERENCE (копия по ссылке)

- метод, используемый при управлении памятью для разделения данных, когда некоторый процесс обращается к ним; в этом случае данные перед обращением копируются и становятся личными. Копия по ссылке допускает разделение начальных значений глобальной секции, страницы которой доступны по чтению и записи, но содержат предварительно инициализированные данные, доступные многим процессам.

COST (стоимость, вес)

- целое значение, приписанное цепи, соединяющей соседние узлы. В соответствии с алгоритмом маршрутизации пакеты данных пересылаются по путям с наименьшей стоимостью.

COUNTED STRING (строка со счетчиком)

- символично-строковая структура данных, состоящая из байта длины, за которым следует строка. Хотя строка со счетчиком не используется как аргумент процедуры, она является удобной для представления в памяти.

COUNTERS (счетчики)

- статистика о характеристиках работы и ошибках таких элементов сети, как линии или узлы.

CRASH (аварийный отказ)

- реакция системы на неустойчивое состояние, в частности на разрушение системы. Система перестает функционировать, а не продолжает работу с возможным нанесением ущерба самой себе. На терминале пользователя нет индикации об аварийном отказе системы, однако, если он произошел, то пользователь не в состоянии работать на терминале. См. Также HANGING (зависание системы).

CURRENT ACCESS MODE (текущий режим доступа)

- режим доступа к процессору выполняющейся в данный момент программы. Этот режим отражен в поле текущего режима в длинном слове состояния процессора.

CURSOR (курсор)

- светящийся индикатор, используемый на видеотерминале для указания того места на экране, где появится следующий символ.

CYLINDER (цилиндр)

- дорожки одного и того же радиуса на всех записывающих поверхностях диска.

DATA (данные)

- общий термин, относящийся к любому представлению информации в виде, годном для передачи, интерпретации или обработки.

00152-01 31 01

DATA CIRCUIT-TERMINATING EQUIPMENT (DCE) (аппаратура передачи данных)

- термин мкктт для интерфейса X.25, относящийся к аппаратуре сети, которая устанавливает, поддерживает и завершает некоторое соединение, управляет преобразованием сигналов и кодированием данных между окончечным оборудованием (DTE) и сетью, коммутирует обмен в сети, к которой присоединено оборудование DTE. Если этот термин используется не вместе с X.25, то он является синонимом термина MODEM (модем).

DATA LINK MAPPING (DLM) (разделение канала передачи данных)

- способность применения интерфейса X.25 виртуальной цепи в качестве канала данных систем "трал" и "магистр".

DATA TERMINAL EQUIPMENT (DTE) (оконечное оборудование данных)

- термин интерфейса X.25, относящийся к пользовательскому оборудованию (эвм или терминалу), подсоединенному к аппаратуре DCE для передачи и/или приема данных в сети с коммутацией пакетов данных.

DATA STRUCTURE (структура данных)

- любые таблица, список, массив, очередь или дерево с корректно определенным форматом и соглашениями по доступу для обращения к ним одного или нескольких образов.

DATA TYPE (тип данных)

- обычно это способ, по которому группируются и интерпретируются биты. По отношению к инструкциям процессора тип данных операнда определяет размер операнда и значение битов в операнде.

DATABASE (база данных)

- 1) все базы данных, описанные системой управления базами данных.
- 2) набор относительных структур данных.

DATAGRAM (датаграмма)

- единица пересылки данных по сети, которая обрабатывается независимо от всех других единиц данных в сети. При добавлении заголовка маршрута датаграмма становится пакетом.

DEBUGGER (отладчик)

- см. SYMBOLIC DEBUGGER (символический отладчик).

DECRYPTION (декодирование)

- процесс, восстанавливающий закодированную информацию к первоначальному незакодированному виду.

DEDICATED (назначенный ресурс)

- системный ресурс (устройство ввода-вывода, образ или вся система), прикрепленный к одной прикладной или целевой программе.

DEFAULT (умолчание, подразумеваемый по умолчанию)

- значение или действие, автоматически включаемые в команду, если пользователь не указал обратное. В большинстве случаев, подразумеваемые по умолчанию

00152-01 31 01

значения будут так называемыми "нормальными" или "предполагаемыми".

DEFERRED ECHO (задержанное эхо)

- относится к тому факту, что вывод эха на терминал не происходит до тех пор, пока процесс не будет готов принять ранее введенные символы.

DELIMITER (разделитель, ограничитель)

- символ, который определяет, завершает или организует элементы символьной строки, оператора или программы.

DELTA TIME (дельта-время, интервал времени)

- значение времени, выражающее смещение от текущей даты и времени. Дельта-время всегда выражается в системе отрицательным числом, абсолютное значение которого используется в качестве смещения от текущего времени.

DEMAND ZERO PAGE (обнуленная по запросу страница)

- страница (обычно в стеке образа или в буферной области), которая инициализируется и полностью обнуляется в тот момент, когда в результате страничного отказа она динамически создается в памяти.

DENSITY (плотность записи)

- число бит на одном миллиметре магнитной ленты. Типичными значениями плотности являются 32 бита на миллиметр и 64 бита на миллиметр.

DESCRIPTOR (дескриптор)

- структура данных, используемая в последовательностях вызовов для передачи типов аргументов, адресов и

00152-01 31 01

другой дополнительной информации. См. CHARACTER STRING DESCRIPTOR (дескриптор символьной строки).

DESIGNATED ROUTER (выделенный маршрутный узел)

- маршрутный узел в магистрали ETHERNET, выделенный для выполнения функций маршрутного обслуживания в интересах конечных узлов.

DETACHED PROCESS (отсоединенный процесс)

- процесс, который не имеет владельца; процесс-предок дерева подпроцессов. Отсоединенный процесс создается диспетчером задания при входе пользователя в систему, при инициализации пакетного задания или при запросе на соединение логического канала. Диспетчер задания не является владельцем создаваемых им процессов пользователей, поэтому эти процессы называются отсоединенными.

DEVICE (устройство)

- общее имя любой периферийной аппаратуры, подключенной к процессору, которая способна получать, хранить или передавать данные. Построчно-печатающие устройства и терминалы являются примерами устройств, ориентированных на записи. Устройства магнитных лент и дисковые устройства являются примерами устройств большой емкости. Терминальные линейные интерфейсы и межпроцессорные каналы являются примерами устройств связи. Устройства не обязательно представляют собой аппаратуру. См. PSEUDO DEVICE (псевдоустройство).

00152-01 31 01

DEVICE CONTROLLER (устройство управления, контроллер)

- электронные схемы, связанные с каждым физическим устройством системы, служащие в качестве интерфейса между процессором и аппаратурой устройства.

DEVICE DATA BLOCK (DDB) (блок данных устройства)

- структура базы данных ввода-вывода, которая определяет общее имя устройства/контроллера и имя драйвера для набора устройств, присоединенных к одному контроллеру.

DEVICE DRIVER (драйвер устройства)

- программа, связанная с каждым физическим устройством системы и служащая в качестве интерфейса между операционной системой и контроллером.

DEVICE INTERRUPT (прерывание устройства)

- прерывание, полученное на уровнях приоритета прерываний от 16 до 23. Прерывания устройств могут быть затребованы только устройствами, контроллерами и памятью.

DEVICE NAME (имя устройства)

- поле в спецификации файла, определяющее устройство, на котором находится файл. Имена устройств включают также и мнемонику, которая определяет периферийное устройство ввода-вывода в запросе на передачу данных. Имя устройства состоит из мнемоники, за которой следует буква индентификатора контроллера (если нужно), затем номер устройства (если нужно) и заключительное двоеточие (:).

00152-01 31 01

DEVICE QUEUE (очередь к устройству)

- см. **SPOOL QUEUE** (очередь спула)

DEVICE REGISTER (регистр устройства)

- ячейка в логической схеме контроллера, используемая для запроса функций устройства (например, функции передачи информации ввода-вывода) и/или сообщения о состоянии.

D_FLOATING POINT DATA (число с плавающей запятой D-формата)

- см. **DOUBLE FLOATING DATA** (число с плавающей запятой D-формата).

DIAGNOSTIC (диагностика)

- программа, которая тестирует аппаратуру, встроенные программы, работу периферийного оборудования, логические схемы или память и сообщает обо всех обнаруженных ошибках.

DIALOG COMMAND LANGUAGE (DCL) (диалоговый командный язык)

- интерпретатор команд в МОС ВП. Противопоставляется программе **MONITOR CONTSOLE ROUTINE** (программа связи с оператором).

DIRECT DATA PATH (прямой путь данных)

- путь данных в адаптере общей шины, по которому за один раз передаются 16 битов данных.

DIRECT I/O (прямой ввод-вывод)

- операция ввода-вывода, во время выполнения которой система блокирует в физической памяти страницы, содержащие буфер. Передача данных происходит непосредственно из буфера процесса. Противопоставляется

00152-01 31 01.

понятию SYSTEM BUFFERED I/O (системный буферизованный ввод-вывод).

DIRECT MAPPING CACHE (кэш-память прямого отображения)

- организация кэш-памяти, в которой любой блок данных основной памяти может быть размещен только в одной возможной позиции кэш-памяти. За счет этого поиск любых данных в кэш-памяти требует лишь одного сравнения адреса. Противопоставляется понятию FULLY ASSOCIATIVE CACHE (полностью ассоциативная кэш-память).

DIRECTORY (каталог)

- файл, содержащий краткие каталоги набора файлов, хранящихся на диске или ленте. Каталог содержит имя, тип и номер версии каждого файла в данном наборе, а также уникальное число, которое определяет фактическое расположение файла и указывает на список его атрибутов. См. Также MASTER FILE DIRECTORY (главный каталог файлов) и SUBDIRECTORY (подкаталог).

DIRECTORY NAME (имя каталога)

- поле в спецификации файла, определяющее файл каталога, в котором содержится информация о файле. Имя каталога начинается с левой скобки ([или <) и кончается правой скобкой (] или >).

DISCONNECT ABORT (аварийное завершение по раз'единению)

- вид раз'единения, при котором задачи могут лишиться логической связи без отсоединения канала. Аварийное завершение по раз'единению служит для приемника

признаком того, что не все посланные сообщения были обязательно получены.

DISCRETIONARY CONTROLS (контроль защиты по усмотрению)

- средства управления защитой, применяемые по усмотрению пользователя, т.е. они не являются обязательными. Примером таких необязательных средств защиты являются списки управления доступом.

DISK SCAVENGING (просмотр остатка данных на диске)

- любой метод получения с диска информации, от которой отказался владелец. Хотя при помощи нормальных средств эта информация больше не доступна первоначальному владельцу, тем не менее, ее можно восстановить и использовать с помощью одного из методов просмотра данных.

DISPLACEMENT DEFERRED INDEXED MODE (косвенный индексный режим со смещением)

- индексный режим адресации, в которой указанный базовый операнд использует адресацию косвенного режима со смещением.

DISPLACEMENT DEFERRED MODE (косвенный режим со смещением)

- режим адресации, при котором расширение спецификатора является смещением в байтах, словах или длинных словах. Смещение расширяется со знаком до 32 битов и добавляется к базовому адресу, извлекаемому из указанного в инструкции регистра. Результатом является адрес длинного слова, которое содержит адрес фактического операнда. Если РС используется как регистр,

00152-01 31 01

то в качестве базового адреса используется скорректированное содержимое РС. Оно является адресом первого байта после расширения спецификатора.

DISPLACEMENT INDEXED MODE (индексный режим со смещением)

- индексный режим адресации, в котором спецификатор базового операнда использует адресацию режима со смещением.

DISPLACEMENT MODE (режим со смещением)

- режим адресации, при котором расширение спецификатора является смещением в байтах, словах или длинных словах. Смещение расширяется со знаком до 32 битов и прибавляется к базовому адресу, извлекаемому из указанного в инструкции регистра. В результате получается адрес фактического операнда. Если РС используется как регистр, то в качестве базового адреса используется скорректированное содержимое РС. Оно является адресом первого байта после расширения спецификатора.

DOUBLE FLOATING DATA (число с плавающей запятой D_формата)

- число с плавающей запятой двойной точности длиной в восемь байтов, имеющее диапазон от (+ или -) $2.9 \times 10^{+37}$ до (+ или -) $1.7 \times 10^{+38}$ и точность приблизительно шестнадцать десятичных цифр.

DOWNLINE SYSTEM LOAD (телезагрузка системы)

- функция системы "трал", позволяющая удаленному приемному узлу получать от другого узла образ файлов операционной системы.

DOWNLINE TASK LOAD (телезагрузка задачи)

- функция системы "трал", позволяющая удаленному приемному узлу получать от другого узла задачу малой ОС РВ.

DRIVE (накопитель)

- электро-механический блок системы устройства большой емкости, на котором смонтирован записывающий носитель данных (дисковая кассета, пакет дисков или бобина с магнитной лентой).

DRIVER (драйвер)

- набор команд и таблиц, под управлением которых на устройстве выполняется физический ввод-вывод.

DRIVER DISPATCH TABLE (DDT) (таблица точек входа драйвера)

- таблица в драйвере ввода-вывода, которая содержит адреса точек входа в стандартные программы драйвера, а также размеры диагностических программ и буферов регистрации ошибок для данного типа устройства.

DRIVER FORK LEVEL (форк-уровень драйвера)

- уровень приоритетов прерываний, на которых выполняется форк-процесс драйвера, т.е. IPL со значением от 8 до 11. Каждый блок управления устройством (UCB) содержит форк-уровень драйвера этого устройства.

DRIVER PROLOQUE TABLE (DPT) (прологовая таблица драйвера)

- таблица в драйвере, описывающая драйвер и тип устройства для той процедуры МОС ВП, которая загружает драйвер в систему.

00152-01 31 01

DRIVER START I/O ROUTINE (программа драйвера запуска ввода-вывода)

- см. **START I/O ROUTINE** (программа запуска ввода-вывода).

DYNAMIC ACCESS (динамический доступ)

- метод переключения программой одного режима доступа к записи на другой во время обработки файла.

EBCDIC (Код дкои)

- расширенный двоично-кодированный десятичный код обмена; набор восьмибитовых символов, используемых для представления данных.

ECC

- код с исправлением ошибок.

ECHO (эхо)

- характеристика управления терминалом, определяющая, что символы, набираемые пользователем на клавиатуре терминала, отображаются также и на экране или печатающем устройстве.

EDITOR (редактор)

- системный образ, используемый для создания и изменения текстовых файлов.

EFFECTIVE ADDRESS (исполнительный адрес)

- адрес, полученный после выполнения косвенной или индексной модификаций.

ENCRYPTION (кодирование)

- процесс кодирования информации, после которого ее содержимое не будет сразу понятно получателю копии

информации.

END NODE (конечный узел)

- узел, который получает адресованные ему пакеты и посылает пакеты на другие узлы, но не может маршрутизировать пакеты через другие узлы.

ENTRY MASK (маска входа)

- слово, биты которого задают сохраняемые и восстанавливаемые регистры при вызове подпрограммы или процедуры, использующей инструкции CALL и RETURN.

ENTRY POINT (точка входа)

- ячейка, которую можно указать в качестве объекта вызова. Она содержит маску входа и биты разрешения исключительных ситуаций, называемые маской точки входа.

EQUIVALENCE NAME (эквивалентное имя)

- строка, связанная с логическим именем в таблице логических имен. Эквивалентным именем может быть, например, имя устройства, другое логическое имя или логическое имя, связанное с частью спецификации некоторого файла. См. Также SEARCH LIST (список поиска)

ERASE-ON-ALLOCATE (стирание при выделении области)

- метод, использующий трафарет стирания при каждом выделении новой области для экстенда файла. Новая область уничтожается по трафарету стирания таким образом, что последующие попытки прочесть эту область могут дать только трафарет стирания, а не

некоторые ценные остатки данных.

ERASURE PATTERN (трафарет стирания)

- символьная строка, которую можно использовать для наложения данных на магнитном носителе с целью стирания информации, хранящейся в данной области.

ERASE-ON-DELETE (стирание при удалении)

- метод, в котором применяется трафарет стирания при удалении или чистке файла.

ERROR LOGGER (регистратор ошибок)

- системный процесс, который освобождает буфера регистрации ошибок и записывает сообщения об ошибках в файл ошибок. Ошибки, регистрируемые системой, включают системные ошибки памяти, ошибки устройств и тайм-ауты устройств, а также прерывания с неправильными адресами векторов.

ESCAPE SEQUENCE (последовательность AP2)

- переход от обычного режима работы к режиму, не являющемуся таковым. Символ AP2 представляет собой код, служащий признаком перехода к режиму AP2. Последовательность AP2 является набором комбинаций символов, начинающихся с символа AP2, которые терминал передает без интерпретации программе обработки данной последовательности.

EVASIVE ACTION (защитное действие)

- действие МОС ВП, выполняемое в ответ на предполагаемые попытки незаконного доступа. МОС ВП имеет ряд критериев, которые используются системой для выявле-

ния таких попыток. Если МОС ВП подозревает, что неавторизованный пользователь пытается войти в систему, то обычно ее защитное действие состоит в блокировке на ограниченное время всех попыток входа.

EVENT (событие)

- 1) изменение состояния процесса или указание о некотором действии, касающемся отдельного процесса или взаимодействующих процессов; об этом сообщается планировщику, который изменяет порядок выполнения процессов. События могут быть синхронными по отношению к выполняющемуся процессу (запрос ожидания) или асинхронными (завершение ввода-вывода). Обмен рабочих наборов, запрос на пробуждение и страничный отказ являются событиями.
- 2) сетевое или учитывающее специфику вычислительной системы событие, запись о котором содержится в регистрирующем элементе.

EVENT CLASS (класс событий)

- конкретная классификация событий. В класс также входит идентификация событий с учетом специфики системы.

EVENT FLAG (флаг события)

- бит в кластере флагов событий, который может быть установлен в 0 или 1 для указания того, что произошло связанное с ним событие. Флаги событий используются для синхронизации действий процесса или нескольких процессов.

EVENT FLAG CLUSTER (кластер флагов событий)

- набор из 32 флагов событий, который используется для извещения о событиях. Для каждого процесса определены четыре кластера: два - локальных для процесса и два - общих. Восемь локальных флагов событий процесса зарезервированы для использования системой.

EVENT TYPE (тип события)

- частный вид события, которое уникально в некотором классе событий.

EXCEPTION (исключительная ситуация)

- событие, обнаруженное аппаратурой или программами (отличное от прерывания и инструкций перехода, или вызова), которое изменяет нормальный порядок выполнения инструкций. Исключительная ситуация возникает только в результате выполнения инструкции или набора инструкций (в то время как прерывание вызывается некоторым, независимым от выполняющейся инструкции действием). Существуют три типа аппаратных исключительных ситуаций: ловушки, сбой и аварийные завершения. Их примерами являются попытки выполнить привилегированные или зарезервированные инструкции, ловушки трассирования, сбой режима совместимости, выполнение инструкций останова, арифметические ловушки, в частности, переполнение, потеря значимости и деление на ноль.

EXCEPTION DISPATCHER (диспетчер исключительных ситуаций)

- процедура операционной системы, которая ищет прог-

00152-01 31 01

рамму обработки состояния, когда оно возникает по исключительной ситуации. Если для исключительной ситуации или состояния никакая программа обработки не найдена, то образ, вызвавший эту ситуацию, завершается.

EXCEPTION ENABLES (разрешение исключительной ситуации)

- см. TRAP ENABLES (разрешение прерывания).

EXCEPTION VECTOR (вектор исключительной ситуации)

- см. VECTOR (вектор).

EXECUTABLE IMAGE (выполняемый образ)

- образ, который может быть выполнен в рамках процесса. Для этого образ считывается из некоторого файла.

EXECUTIVE (управляющая программа)

- общее имя для набора процедур операционной системы, которые выполняют основные функции управления и контроля.

EXECUTIVE MODE (режим управления)

- второй по привилегированности режим доступа к процессору. В этом режиме работает СУД-32 и ряд процедур системного обслуживания.

EXECUTOR NODE (узел исполнитель)

- узел, на котором фактически выполняется команда программы NCP.

EXIT (выход, завершение)

- действие, совершаемое как при нормальном, так и при ненормальном окончании выполнения образа. К таким действиям относится, например, отсоединение каналов

00152-01 31 01

ввода-вывода и открепление кластеров общих флагов событий. При этом вызываются определенные пользователем или системой программы обработки выхода.

EXIT HANDLER (программа обработки выхода)

- программа, выполняемая при завершении образа. Программа обработки выхода позволяет процедуре, не находящейся в стеке вызова, получить управление и очистить свои базы данных до фактического завершения образа.

EXTENDED ATTRIBUTE BLOCK (ХАВ) (блок дополнительных атрибутов)

- пользовательская структура данных СУД-32, которая помимо атрибутов, определенных в блоке доступа к файлу (ФАВ), содержит такие дополнительные атрибуты, как типы границ (выровненный на цилиндре номер логического блока, номер виртуального блока) и информацию о защите файла.

EXTENSION (расширение)

- размер области, выделяемой в конце файла всякий раз, когда последующая запись превышает зарезервированную длину файла.

EXTENT (экстент)

- непрерывная область на диске, содержащая файл или часть файла. Состоит из одного или более кластеров.

FAILURE EXCEPTION MODE (режим исключительной ситуации по ошибке)

- режим выполнения, выбранный процессом, который дает указание системе объявить состояние исключительной ситуации, если в результате вызова программы системного обслуживания произойдет ошибка. Обычный для программ системного обслуживания режим заключается в возврате кода состояния ошибки, который должен быть проверен процессом.

FAULT (сбой)

- состояние аппаратной исключительной ситуации, которое возникает в середине инструкции и оставляет регистры и память в таком состоянии, что устранение сбоя и повторное выполнение инструкции даст правильный результат.

FDT ROUTINE (программы FDT)

- программы драйвера, вызываемые программой системного обслуживания очереди запросов ввода-вывода для выполнения зависимой от устройства предобработки запроса ввода-вывода.

F_FLOATING-POINT DATA (число с плавающей запятой F_формата)

- см. FLOATING (POINT) DATA (число с плавающей запятой).

FIELD (поле)

- набор последовательных байтов в логической записи. См. Также VARIABLE-LENGTH BIT FIELD (битовое поле переменной длины).

FIFO (дисциплина обслуживания FIFO)

- "первым пришел - первым вышел"; порядок, в котором

00152-01 31 01

выполняется обработка. Например, обработка очереди по дисциплине FIFO будет проводиться на основе "первым пришел - первым обслужен". См. Также LIFO (дисциплина обслуживания LIFO).

FILE (файл)

- набор элементов данных, расположенных в некоторой структуре, значимой для пользователя. Файл - это любые поименованные, находящиеся в памяти данные или программа (или то и другое вместе), к которым имеет доступ система. Доступ может быть двух типов: только по чтению, означающий, что файл не должен изменяться, а также по чтению и записи, означающий, что содержимое файла может быть изменено. См. Также VOLUME (том).

FILE ACCESS BLOCK (FAB) (блок доступа к файлу)

- структура данных пользователя СУД-32, которая описывает конкретный файл и относящуюся к нему информацию, необходимую для таких операций над файлами, как OPEN (открыть), CLOSE (закрыть) или CREATE (создать).

FILE DEFINITION LANGUAGE (FDL) (язык описания файлов)

- специальный язык, используемый для написания спецификаций файлов данных. Эти спецификации пишутся в текстовых файлах, называемых файлами FDL. Они используются обслуживающими и библиотечными программами СУД-32 для создания фактических файлов данных.

FILE HEADER (заголовок файла)

- блок в индексном файле, описывающий файл дисковой файловой структуры. Заголовок файла содержит информацию, необходимую файловой системе для поиска и использования файла. Часть этой информации отображается по команде DIRECTORY языка DCL. Для каждого дискового файла имеется по крайней мере один заголовок.

FILE IDENTIFIER (FID) (идентификатор файла)

- шестибайтовое значение, используемое для однозначного опознания файла на дисковом томе файловой структуры. В идентификаторе файла содержится номер файла, номер файловой последовательности и относительный номер тома.

FILE NAME (имя файла)

- поле, предшествующее типу файла в спецификации файла, которое содержит от 1 до 39 символов имени файла.

FILE NAME EXTENSION (расширение имени файла)

- см. FILE TYPE (тип файла).

FILE ORGANIZATION (организация файла)

- определенная структура файла, которая используется для физического упорядочивания данных, на носителе большой емкости. Организация файлов СУД-32 бывает последовательной, относительной и индексной.

FILE SECTION (секция файла)

- секция файла является частью файла, которая содержит

пользовательские данные и которая выделяется заголовком и граничными метками. Внутри некоторой секции файла не могут быть вставлены другие или множественные секции файла.

FILE SHARING (разделение файла)

- возможность доступа к данному относительному или индексному файлу, предоставляемая несколькими процессами.

FILE SPECIFICATION (спецификация файла)

- уникальное имя для файла на носителе большой емкости. Оно определяет узел, устройство, имя каталога, имя файла, тип файла и номер версии, под которой хранится файл.

FILE STRUCTURE (файловая структура)

- способ, по которому составляющие файл блоки размещаются на диске или магнитной ленте; этот способ обеспечивает физический метод доступа в соответствии со способом обработки данных файла.

FILE SYSTEM (файловая система)

- метод записи, ведения каталогов и обращения к файлам на некотором томе.

FILE TYPE (тип файла)

- поле в спецификации файла, состоящее из символа точки ".", за которой следует от 0 до 39 символов определения типа. По соглашению, это поле идентифицирует общий класс файлов, которые имеют одинаковые характеристики или применение. Таковы, например, компиля-

тор или ассемблер, файлы листинга, двоичные объектные файлы и т.д.

FILES-SM (файловая структура)

- имя дисковой структуры, используемой операционными системами ОС РВ и МОС ВП. См также FILES-SM STRUCTURE LEVEL 1 (файловая структура уровня 1) и FILES-11 STRUCTURE LEVEL 2 (файловая структура уровня 2).

FILES-SM STRUCTURE LEVEL 1 (файловая структура уровня 1)

- исходная файловая структура, используемая операционными системами МОС ВП и ОС РВ для дисковых томов. Мос вп поддерживает этот уровень для обеспечения совместимости операционных систем.

FILES-SM STRUCTURE LEVEL 2 (файловая структура уровня 2)

- структура дисковых файлов второго поколения, поддерживаемая системой МОС ВП.

FIXED-LENGTH CONTROL AREA (область управления фиксированной длины)

- область, находящаяся перед записью переменной длины и содержащая дополнительную информацию об этой записи; область может не иметь отношения к основному содержимому записи. Область управления фиксированной длины может быть использована, например, для хранения пронумерованных строк или информации, предназначенной для управления кареткой.

FIXED-LENGTH RECORD FORMAT (формат записи фиксированной

длины)

- формат файла, в котором все записи имеют одинаковую длину.

FIXED LINE NUMBERS (фиксированные номера строк)

- номера, закрепленные за строками текста в некотором файле. Редактор EDT сохраняет записи о номерах во время сеанса редактирования, в котором используются файлы; по окончании этого сеанса номера копируются в файл.

FLAG (флаг)

- бит, который может быть установлен в 1 для инициирования выполнения конкретной последовательности инструкций.

FLOATING (POINT) DATA (число с плавающей запятой)

- число с плавающей запятой одинарной точности, длиной 4 байта, имеющее диапазон от (+ или -) $2.9 \cdot 10^{(-37)}$ до (+ или -) $1.7 \cdot 10^{38}$ и точность примерно 7 десятичных знаков.

FOREIGN VOLUME (чужой том)

- любой том, отличный по формату от тома с файловой структурой.

FORK BLOCK (форк-блок)

- часть блока управления устройством, которая во время ожидания драйвером некоторого ресурса содержит контекст драйвера. Форк-блок того драйвера, который ждет ресурс процессора, помещен в форк-очередь.

FORK DISPATCHER (форк-диспетчер)

- программа МОС ВП, обслуживающая прерывания; она активизируется программным прерыванием на форк-уровне приоритета прерываний. Активизированный форк-диспетчер выбирает форк-процессы драйвера из форк-очереди до тех пор, пока для этого IPL в очереди есть процессы.

FORK PROCESS (форк-процесс)

- процесс с минимальным контекстом, который выполняет программный образ при ряде следующих ограничений: он работает на повышенном уровне приоритета прерываний (IPL) и в виртуальном адресном пространстве системы; он использует лишь регистры R0-R5 (другие регистры должны быть сохранены и восстановлены); ему разрешено обращаться только к статической памяти и модифицировать ее (эта память никогда не модифицируется программами более высокого уровня приоритета прерываний). МОС ВП использует программные прерывания и форк-процессы для синхронизации действий управляющей программы.

FORK QUEUE (форк-очередь)

- очередь форк-блоков драйвера, которая ждет своей активизации форк-диспетчером МОС ВП на отдельном уровне приоритета прерываний.

FORM FEED (перевод формата)

- перемещение положения курсора на начало новой страницы, (используется в редакторе EDT); один из подра-

00152-01 31 01

зумеваемых по умолчанию ограничителей слов; подразумеваемый по умолчанию ограничитель страницы.

FRAME (кадр)

- блок, который ограничен флагами и содержит заголовок; он используется на уровне канала в сетях с коммутацией пакетов данных для обмена между DTE и DCE пакетами данных, управляющей информацией и информацией об ошибках.

FRAME POINTER (FP) (указатель кадра)

- общий регистр 13 (R13). По соглашению, он содержит базовый адрес самого последнего кадра вызова в стеке.

FULLY ASSOCIATIVE CACHE (полностью ассоциативная кэш-память)

- организация кэш-памяти, при которой любой блок данных из главной памяти может быть помещен в любое место кэш-памяти. Для того, чтобы найти конкретный блок в кэш-памяти необходимо произвести сравнение адресов каждого блока. Такая кэш-память противоположна DIRECT MAPPING CACHE (кэш-памяти прямого отображения).

FUNCTION CODE (код функции)

- см. I/O FUNCTION CODE (код функции ввода-вывода).

FUNCTION DECISION TABLE (FDT) (таблица выбора функций)

- таблица драйвера, в которой перечисляются все действительные коды функций устройств, а также приводятся адреса программ предобработки ввода-вывода, связан-

00152-01 31 01

ных с каждой такой функцией.

FUNCTION MODIFIER (модификатор функции)

- см. I/O FUNCTION MODIFIER (модификатор функции ввода-вывода).

GENERAL IDENTIFIER (общий идентификатор)

- один из трех возможных типов идентификаторов, который определяет одну или несколько групп пользователей. Общий идентификатор является буквенно-цифровым.

GENERAL REGISTER (общий регистр)

- любой из шестнадцати 32-х битных регистров, используемых в качестве первичных операндов инструкций основного режима. Общие регистры включают 12 регистров общего назначения, которые могут быть использованы как накопители, счетчики и указатели ячеек основной памяти, а также четыре специальных регистра: указатель кадра (FP), указатель аргумента (AP), указатель стека (SP), счетчик инструкций (PC).

GENERIC DEVICE NAME (тип устройства)

- имя устройства, определяющее не конкретное устройство, а класс устройств; имя устройства, в котором опущен конкретный контроллер или номер устройства.

G_FLOATING DATA (число с плавающей запятой G_формата)

- число с плавающей запятой расширенного диапазона, длиной 8 байтов, лежащее в интервале от (+ или -) $0.56 \times 10^{**(-308)}$ до $9 \times 10^{**308}$ и имеющее точность примерно 15 десятичных цифр.

GLOBAL (глобальный)

- воздействующий на весь файл, на всю систему или на весь образ (в зависимости от контекста).

GLOBAL PAGE TABLE (таблица глобальных страниц)

- таблица страниц, содержащая элементы основной таблицы страниц для глобальных секций.

GLOBAL SECTION (глобальная секция)

- структура данных (например, глобальная общая область для ФОРТРАНА) или разделяемая секция образа, потенциально доступная для всех процессов в системе. Доступ к ней защищается привилегией и/или номером группы в коде идентификации пользователя.

GLOBAL SYMBOL (глобальный символ)

- 1) определяемый в модуле символ, который потенциально доступен для ссылки из другого модуля. Глобальные символы разрешаются (путем сопоставления ссылок и определений) компоновщиком. Противопоставляется понятию LOCAL SYMBOL (локальный символ).
- 2) символ командного языка, доступный со всех командных уровней.

GLOBAL SYMBOL TABLE (GST) (таблица глобальных символов)

- индекс сильно определенных глобальных символов, используемый в библиотеке для доступа к модулям, определяющим глобальные символы. Компоновщик будет, кроме того, помещать таблицы глобальных символов в образ. Например, компоновщик добавляет таблицу глобальных символов к выполняемым образам, которые

предназначены для работы с символическим отладчиком. Компоновщик добавляет таблицу глобальных символов и ко всем разделяемым образам.

GOLD KEY (символ определения функций клавиатуры)

- верхний левый символ на клавиатуре терминала, который позволяет менять функции клавиатуры.

GROUP (группа)

- 1) множество пользователей, которые имеют специальные привилегии на доступ к каталогам друг друга и к файлам внутри этих каталогов (если они не защищены особым образом). Так, в контексте "система/владелец/группа/все", понятие группы относится ко всем членам данной группы владельцев.
- 2) набор заданий (процессов и их подпроцессов) с правом доступа к общим флагам событий группы и к таблицам логических имен.

GROUP NUMBER (номер группы)

- первое число или его буквенно-цифровой эквивалент в коде идентификации пользователя.

HANDSHAKING SEQUENCE (последовательность "рукопожатия")

- обмен информацией при установлении связи по логическому каналу. Он выполняется для того, чтобы обеспечить успешное завершение такой связи.

HANGING (заисание системы)

- крайне медленная реакция системы. Если при интерактивной работе кажется, что команды с терминала никуда не поступают или ничего не выполняют, то говорят,

00152-01 31 01

что имеет место зависание системы.

HARD-COPY TERMINAL (терминал с твердой копией)

- терминал, который распечатывает выводимую информацию на бумаге. См. Также **TERMINAL** (терминал).

HARDWARE ADDRESS (аппаратный адрес)

- для устройства магистрали **ETHERNET** - это уникальный физический адрес, закрепленный изготовителем за конкретным контроллером связей **ETHERNET** (обычно, в постоянной памяти).

HARDWARE CONTEXT (аппаратный контекст)

- значения, которые во время выполнения процесса содержатся в следующих регистрах:
 - счетчике инструкций (**PC**);
 - длинном слове состояния процессора (**PSL**);
 - 14 общих регистрах (от **R0** до **R13**);
 - четырех регистрах процессора (**POBR**, **POLR**, **P1BR**, **P1LR**), описывающих виртуальное адресное пространство процесса;
 - указателе стека (**SP**) для режима доступа, в котором работает процессор;
 - содержимой, которое должно быть загружено в **SP** для каждого режима доступа, отличного от текущего режима.

Во время выполнения процесса его аппаратный контекст постоянно корректируется процессором. Если процесс не выполняется, то аппаратный контекст этого процесса хранится в его аппаратном блоке

PCB.

HARDWARE PROCESS CONTROL BLOCK (аппаратный блок управления процессом; аппаратный PCB)

- известная процессору структура данных, которая содержит сохраненный аппаратный контекст в то время, когда процесс не выполняется. Аппаратный PCB процесса размещается в его заголовке.

HIBERNATION ("спячка")

- состояние, в котором процесс неактивен, но известен системе со всеми его текущими состояниями. "спящий" процесс становится активным вновь при выдаче запроса на его пробуждение. Процесс может планировать запрос на пробуждение перед переходом в "спячку", или же запрос на его пробуждение может выдать другой процесс. Кроме того, "спящий" процесс может стать активным на время, достаточное для обслуживания любого AST, которое он возможно получит, пока "спит". Противопоставляется понятию SUSPENSION (приостанов).

HIGHWATER MARKING (метод "полного уровня")

- этот метод состоит в том, что система отслеживает самый последний экстенд, который был записан пользователем в выделенную для файла область. После этого система препятствует любым попыткам чтения информации из области, в которую запись не выполнялась, основываясь на предпосылке, что любая информация, существующая сверх записанного в данное время лимита

00152-01 31 01

та, является информацией, которую пользователь намеревается уничтожить. Мос вп решает задачи метода "полного уровня", в соответствии со стратегией стирания при выделении области для экстенда.

H_FLOATING DATA (число с плавающей запятой H_формата)

- число с плавающей запятой расширенного диапазона, состоящее из 16 байтов, заключенное в интервале от (+ или -) $0.84 \times 10^{+4932}$ до (+ или -) $0.59 \times 10^{+4932}$ и с точностью около 33 десятичных цифр.

HOLDER (держатель идентификатора)

- пользователь, владеющий конкретным идентификатором. Говорят, что пользователь является держателем идентификатора, если он владеет данным идентификатором. Местом, где система связывает пользователей и принадлежащие им идентификаторы, является база данных прав по идентификаторам.

HOME BLOCK (собственный блок тома)

- блок в индексном файле, который содержит определение тома, например, метку тома и код защиты.

HOP (транзитный участок)

- логическое расстояние между двумя узлами. Один транзитный участок - это расстояние между смежными узлами.

HOST NODE (главный узел)

- 1) в системах "трал" или "магистр" - это узел, который обеспечивает обслуживание других узлов (например, главный узел предоставляет файлы образа программы

00152-01 31 01

на телезагрузку).

- 2) узел, который делает некоторое устройство доступным для других узлов кластера.

IDENTIFIER (идентификатор)

- обозначение, которое представляет пользователя или группу пользователей в системе. Существуют три типа идентификаторов: идентификаторы UIC, системно-определенные идентификаторы и общие идентификаторы.

IMAGE (образ)

- образ состоит из процедур и данных, связанных компонентом вместе. Имеется три типа образов: выполняемые, разделяемые и системные.

IMAGE ACTIVATOR (активизатор образа)

- активизатор образа создает структуры данных управления памятью, необходимые, как для отображения виртуальных страниц образа в физические страницы, так и для выполнения обмена страниц.

IMAGE EXIT (завершение образа)

- см. EXIT (выход)

IMAGE I/O SEGMENT (сегмент ввода-вывода образа)

- часть области управления, которая содержит внутренние блоки доступа к файлу (IFAB) системы СУД-32, а также буфера ввода-вывода для образа, выполняющегося в данный момент в рамках некоторого процесса.

IMAGE NAME (имя образа)

- имя файла, в котором хранится образ.

00152-01 31 01

IMAGE PRIVILEGES (привилегии образа)

- привилегии, приписанные образу при его установке.
См. PROCESS PRIVILEGES (привилегии процесса).

IMAGE SECTION (ISECT) (секция образа)

- группа программных секций с одинаковыми атрибутами (доступны только для чтения, доступны для чтения и записи, абсолютные, перемещаемые и т.д.), ISECT является единицей выделения виртуальной памяти для образа.

IMMEDIATE MODE (непосредственный режим)

- режим адресации, при котором pc используется как регистр в режиме адресации с автоувеличением.

INBOUND CONNECTION (запросы на соединение)

- посылаемые задачей запросы на соединение по логическому каналу.

INCREMENT (приращение, увеличение)

- добавить одну величину к другой; добавленная величина.

INDEX (индекс)

- структура, позволяющая отыскать запись в индексном файле по значению ключа. См. KEY (INDEX FILES) (ключ индексных файлов).

INDEX FILE (индексный файл)

- файл на томе файловой структуры, он содержит информацию для доступа ко всем файлам тома, позволяет операционной системе опознать том и обратиться к нему.

INDEX FILE BIT MAP (карта памяти индексного файла)

- таблица в индексном файле на томе файловой структуры, в которой указываются используемые заголовки файлов.

INDEX REGISTER (индексный регистр)

- регистр, используемый для хранения адресного смещения.

INDEXED ADDRESSING MODE (индексный режим адресации)

- режим адресации, в котором для определения фактического операнда инструкции используются два регистра: индексный регистр и спецификатор базового операнда. Содержимое индексного регистра используется как индекс (смещение) в таблице или массиве. Спецификатор базового операнда дает базовый адрес массива (базовый адрес операнда). Адрес фактического операнда вычисляется умножением содержимого индексного регистра на размер (в байтах) фактического операнда и добавлением полученного произведения к базовому адресу операнда. Режимы адресации, вытекающие из адресации индексного режима, образуются добавлением слова "индексный" или "с индексацией" к режиму адресации спецификатора базового операнда: косвенно-регистровый с индексацией, индексный с автоувеличением, косвенный индексный с автоувеличением (или абсолютный индексный), индексный с автоуменьшением, индексный со смещением и косвенный индексный со смещением.

INDEXED FILE ORGANIZATION (индексная организация файла)

- организация файла, при которой файл содержит записи и индекс первичного ключа (и, необязательно, один или несколько индексов альтернативных ключей), используемые для обработки записей последовательно по индексу или произвольно по индексу.

INDIRECT COMMAND FILE (косвенный командный файл)

- см. COMMAND PROCEDURE (командная процедура)

INPUT STREAM (входной поток)

- источник команд и данных. Это или терминал пользователя, или пакетный поток, или командная процедура.

INSTRUCTION BUFFER (буфер инструкций)

- восьмибайтовый буфер процессора, используемый для хранения байтов декодируемой инструкции и для предварительного выбора инструкций из потока инструкций. Для поддержания этого восьмибайтового буфера в заполненном состоянии управляющие логические схемы непрерывно выбирают данные из памяти.

INTERACTIVE SYSTEM (интерактивная система)

- вычислительная система, в которой пользователь и система взаимодействуют непосредственно через терминал. Операционная система немедленно принимает запросы, вводимые пользователем с терминала, и выполняет их.

INTERLEAVING (чередующаяся адресация)

- назначение последовательных физических адресов памяти поочередно между двумя контроллерами памяти.

INTERPROCESS COMMUNICATION FACILITY (средства межпроцессных связей)

- кластер общих флагов событий, почтовый ящик или глобальная секция, используемые для передачи информации между двумя или несколькими процессами.

INTERRECORD GAP IRG (межблочный промежуток)

- пустое пространство, намеренно помещаемое между записями данных на записывающей поверхности магнитной ленты.

INTERRUPT (прерывание)

- 1) событие, отличное от исключительной ситуации, инструкций перехода, выбора или вызова, которое изменяет обычный порядок выполнения инструкций. Прерывания обычно являются внешними по отношению к процессу, выполняющемуся в тот момент, когда происходит прерывание. См. Также **DEVICE INTERRUPT** (прерывание устройства), **SOFTWARE INTERRUPT** (программное прерывание) и **URGENT INTERRUPT** (срочное прерывание).
- 2) пакет, посылаемый через сеть с коммутацией пакетов данных, который обходит обычные процедуры управления потоком, используемые пакетами данных.

INTERRUPT DATA BLOCK (IDB) (блок данных прерывания)

- структура в базе данных ввода-вывода, которая описывает характеристики отдельного контроллера и отмечает устройства, присоединенные к этому контроллеру.

INTERRUPT MESSAGE (сообщение прерывания)

- сгенерированное пользователем сообщение, которое

посылается помимо обычного обмена сообщениями данных при непрозрачной связи между задачами. Такое применение термина "прерывание" противоположно обычному, означающему программный или аппаратный механизм прерывания.

INTERRUPT PRIORITY LEVEL (IPL) (уровень приоритета прерываний)

- уровень прерывания, на котором генерируется аппаратное или программное прерывание. Существует 31 уровень возможных приоритетов прерываний: IPL 1 является самым низким, 31 - самым высоким. Эти уровни разрешают конфликтные ситуации при поступлении запросов на обслуживание процессором. Например, устройство не может прервать процессор, если в текущий момент он работает на уровне приоритета прерываний, не меньшем, чем уровень программы обслуживания прерывания от этого устройства.

INTERRUPT SERVICE ROUTINE (ISR) (программа обслуживания прерывания)

- программа, выполняемая при прерывании.

INTERRUPT STACK (IS) (стек прерывания)

- общесистемный стек, используемый при работе в контексте обслуживания прерываний. Процессор всегда находится или в контексте процесса, работая в одном из режимов - пользователя, управления. Супервизора или ядра - или же в общесистемном контексте обслужи-

вания прерываний, работая в режиме ядра. Это указывается битами стека прерываний и текущего режима в длинном слове состояния процессора. Контекст стека прерываний не переключается.

INTERRUPT STACK POINTER (ISP) (указатель стека прерываний)

- указатель общесистемного стека прерываний.

INTERRUPT VECTOR (вектор прерывания)

- см. VECTOR (вектор).

I/O DATABASE (база данных ввода-вывода)

- набор структур данных, который описывает запросы ввода-вывода, контроллеры, устройства, тома и драйверы устройств системы МДС ВП. Примерами таких структур являются таблица точек входа драйвера, прологовая таблица драйвера, таблица данных устройства, блок управления устройством, блок запросов канала, пакет запроса ввода-вывода, блок данных прерывания.

I/O DRIVER (драйвер ввода-вывода)

- см. DRIVER (драйвер).

I/O FUNCTION (функция ввода-вывода)

- операция ввода-вывода, интерпретируемая операционной системой и дающая в результате одну или несколько физических операций ввода-вывода.

I/O FUNCTION CODE (код функции ввода-вывода)

- 6-битное значение, заданное в программе системного обслуживания очереди запросов ввода-вывода, описывающее операцию ввода-вывода, которую необходимо выполнить (например, чтение, запись, перемотка).

00152-01 31 01

I/O FUNCTION MODIFIER (модификатор функции ввода-вывода)

- 10-битное значение, указанное в программе системного обслуживания очереди запросов ввода-вывода, модифицирующее код функции ввода-вывода, например, ввод с терминала без эхо-сопровождения.

I/O LOCKDOWN (блокировка страницы по вводу-выводу)

- состояние страницы, при котором она не может быть выгружена из памяти ни программой обмена страниц, ни планировщиком памяти.

I/O REQUEST PACKET (IRP) (пакет запроса ввода-вывода)

- структура в базе данных ввода-вывода, описывающая отдельный запрос ввода-вывода. Такой пакет для каждого запроса ввода-вывода создается программой системного обслуживания очереди запросов ввода-вывода. Мос вп и драйвер приемного устройства используют информацию этого пакета для обработки запроса.

I/O RUNDOWN (дозавершение ввода-вывода, довывод)

- функция операционной системы, осуществляющая чистку всех выполняемых операций ввода-вывода при завершении образа.

I/O SPACE (пространство ввода-вывода)

- область в физическом адресном пространстве, которая содержит регистры конфигурации, команд и состояния, а также данных. Такие области физически не смежны.

I/O STATUS BLOCK (IOSB) (блок состояния ввода-вывода)

- структура данных, связанная с программой системного обслуживания очереди запросов ввода-вывода. Эта

программа дополнительно возвращает в IOSB код состояния, число переданных байтов, а также информацию, зависящую от устройства и функции.

JOB (задание)

- учетная единица, эквивалентная процессу и его подпроцессам (если они есть) и всем подпроцессам, которые они создали. Задания делятся на пакетные и интерактивные. Например, когда пользователь входит в систему, диспетчер заданий создает интерактивное задание для обработки пользовательских запросов, а когда программа управления симбионтами передает этому диспетчеру командный входной файл, диспетчер создает пакетное задание.

JOB CONTROLLER (диспетчер заданий)

- системный процесс, который создает контекст процесса некоторого задания, запускает процесс, выполняющий образ LOGIN для этого задания, сопровождает учетный раздел задания, управляет симбионтами, а также завершает процесс и его подпроцессы.

JOB INFORMATION BLOCK (JIB) (информационный блок задания)

- связанная с заданием структура данных, которая содержит квоты, собранные вместе от всех процессов задания.

JOURNAL FILE (журнальный файл)

- файл, который содержит данные, вводимые с терминала за один сеанс редактирования редактором EDT.

JOB QUEUE (очередь заданий)

- список файлов, которые были переданы процессору для работы с конкретным устройством, например, построчно-печатающим устройством.

KERNEL MODE (режим ядра)

- самый привилегированный режим доступа к процессору. В режиме ядра работают наиболее привилегированные программы операционной системы, например, драйверы ввода-вывода и программа обмена страниц.

KEY (ключ, символ)

- для индексных файлов ключ - это строка символов, упакованное десятичное число, двух- или четырехбайтное двоичное число без знака, двух- или четырехбайтное число со знаком внутри каждой записи данных индексного файла. Пользователь определяет длину и местоположение ключа внутри записей; СУД-32 использует ключ для построения индекса. См. PRIMARY KEY (первичный ключ), ALTERNATE KEY (альтернативный ключ), RANDOM ACCESS BY KEY VALUE (произвольный доступ по значению ключа). В относительных файлах ключ - это относительный номер каждой записи данных в файле данных. СУД-32 использует относительные номера записей для идентификации записей данных и обращения к ним при произвольном режиме доступа к относительному файлу см. RELATIVE RECORD NUMBER (относительный номер записи). В обслуживающей программе SORT (сортировка) ключ - это поле данных в записи, содержащее

00152-01 31 01

информацию, при помощи которой пользователь сортирует записи.

KEYPAD (малая клавиатура)

- малый набор клавиш, следующий за основной клавиатурой терминала.

KEYWORD (ключевое слово)

- имя команды, квалификатор или возможность. Большинство ключевых слов должны быть напечатаны полностью, другие же могут быть сокращены в соответствии с правилами языка DCL.

KNOWN COMPONENT (известный элемент)

- классификация для одного или нескольких одинаковых элементов. Эта классификация включает все активные и неактивные проявления типов элементов, например, известные узлы включают активные и неактивные узлы в сети.

LABEL (метка)

- запись, которая определяет и ограничивает том магнитной ленты или секцию файла.

LABEL GROUP (групповая метка)

- множество, состоящее из одного или нескольких непрерывных наборов меток.

LABEL IDENTIFIER (идентификатор метки)

- первые три символа имени метки, которые определяют одну или несколько меток внутри набора меток. Эти символы будут всегда одинаковыми, например, идентификатор файла HDR будет использоваться для определе-

ния меток заголовков внутри набора меток заголовков.

LABEL NUMBER (номер метки)

- номер, который указывает положение метки внутри набора меток. Для стандартных меток номер метки присутствует всегда, если существует набор меток.

LABEL SET (набор меток)

- одна или несколько непрерывных меток на том же магнитной ленте или секции файла с одинаковыми идентификаторами меток.

LEVEL 1 ROUTER (маршрутный узел 1 уровня)

- узел, который может посылать или получать пакеты, а также маршрутизировать пакеты от одного узла к другому, но только внутри одного региона.

LEVEL 2 ROUTER (маршрутный узел 2 уровня)

- узел, который может посылать или получать пакеты, а также маршрутизировать пакеты от одного узла к другому как внутри своего региона, так и между регионами. Он также называется AREA ROUTER (маршрутный узел региона).

LEXICAL FUNCTION (лексическая функция)

- конструкция командного языка DCL, которую интерпретатор команд вычисляет и заменяет до синтаксического разбора командной строки. Лексические функции выдают информацию о текущем процессе (например, информацию о коде UIC или о подразумеваемом по умолчанию каталоге) и о символьных строках (например, об их длине или размещении подстрок).

LIBRARIAN (библиотекарь)

- программа, которая позволяет пользователю создавать, корректировать, модифицировать, распечатывать и сохранять файлы, объектные библиотеки помощи, библиотеки текстов и библиотечные файлы макроассемблера.

LIBRARY FILE (библиотечный файл)

- файл прямого доступа, содержащий один или несколько модулей одного и того же типа.

LIFO (дисциплина обслуживания LIFO)

- порядок выполнения процессов. Данные в LIFO очереди будут обрабатываться на основе дисциплины "последним пришел, первым обслужен" см. Также FIFO (дисциплина обслуживания FIFO).

LIMIT (лимит)

- размер или число элементов требуемых системных ресурсов (таких, как почтовые ящики, захваченные страницы, запросы ввода-вывода, открытые файлы и т.д.), которые одновременно разрешается иметь выполняющемуся заданию. Лимиты задаются администратором системы в файле авторизации пользователя. См. QUOTA (квота).

LINE (линия)

- элемент управления сетью, который предоставляет отдельный физический путь данных.

LINE BUFFER (буфер строки)

- область памяти, которая используется для хранения последней строки, удаленной соответствующей опера-

00152-01 31 01

цией редактора EDT.

LINE NUMBER (номер строки)

- число, которое используется для идентификации текстовой строки в файле, обрабатываемом текстовым редактором.

LINE PRINTER (построчно-печатающее устройство)

- устройство вывода, распечатывающее файлы сразу по строке. Оно используется для печати больших объемов информации вывода, которая иначе была бы привязана к более медленному устройству. Почти в каждой системе есть устройство, предназначенное для построчной печати. В некоторых случаях "построчно-печатающее устройство" фактически является высокоскоростным терминалом.

LINK ACCESS PROTOCOL (LAP) (протокол доступа к каналу)

- набор процедур, используемый для управления каналом в сетях с коммутацией пакетов данных. Рекомендация X.25 определяет два набора процедур:

1) LAP: интерфейс DTE/DCE, действующий в двухстороннем сбалансированном, асинхронном режиме (ARM) с оборудованием DTE и DCE, содержащим основную и вторичную функцию;

2) LABP: интерфейс DTE/DCE определен, как действующий в двухстороннем асинхронном, сбалансированном режиме (ABM).

LINKER (компоновщик)

- системная программа, которая создает выполняемую

программу (называемую образом) из одного или нескольких объектных модулей, выданных транслятором. Для того, чтобы программы могли быть выполнены, они вначале должны быть скомпонованы.

LINKING (компоновка)

- разрешение внешних ссылок между используемыми для создания образа объектными модулями, получение ссылочных библиотечных программ, обслуживание точек входа и данных для образа, назначение виртуальных адресов компонентам образа.

LITERAL MODE (литеральный режим)

- режим адресации, в котором операндом инструкции является константа. Ее значение определено в 6-битном поле инструкции. Если тип данных операнда есть байт, слово, длинное слово или квадрослово, то операнд расширяется нулями и может задавать значения в диапазоне от 0 до 63 (десятичное). Если тип данных операнда есть обычное или удвоенное число с плавающей запятой, то 6-битное поле состоит из двух трехбитных полей: одно - для экспоненты, другое - для мантииссы. Операнд расширяется до обычного или удвоенного формата числа с плавающей запятой.

LOAD DEVICE (устройство загрузки)

- накопитель, содержащий носитель дистрибутивов во время установки программного обеспечения.

LOCAL DISK (локальный диск)

- дисковый накопитель в режиме CI, который не зависит

00152-01 31 01

от управления HSC.

LOCAL NODE (локальный узел)

- узел сети, на котором располагается пользователь.

LOCALITY (локальность)

- см. PROGRAM LOCALITY (локальность программы).

LOCAL SYMBOL (локальный символ)

- 1) символ, имеющий смысл только в том модуле, в котором он определен. Символы, не опознанные транслятором как глобальные, считаются локальными символами. Локальные символы не известны компоновщику и не могут быть доступны другому объектному модулю. Однако, при помощи компоновщика они могут быть переданы символическому отладчику. Противопоставляется понятию GLOBAL SYMBOL (глобальный символ).

- 2) символ командного языка, который доступен только на текущем командном уровне и на уровнях, вызванных с него.

LOCATE MODE (режим указания)

- метод доступа к записям СУД-32, при котором для уменьшения накладных расходов, программа осуществляет доступ к записям в буферной области ввода-вывода СУД-32.

LOCK MODE (режим захвата)

- значение, связанное с запросом к программам системного обслуживания управления захватом, которое характеризует совместимость запрошенного захвата с другими захватами.

LOCK VALUE BLOCK (блок значения захвата)

- необязательный блок данных, связанный с блоком состояния захвата. Блок значения захвата может быть использован для обмена информацией между процессами, разделяющими ресурс.

LOCKED PASSWORD (заблокированный пароль)

- пароль, который не может быть изменен владельцем учетного раздела. Его могут изменить лишь администратор системы или пользователи, имеющие привилегию SYSPRV.

LOCKING A PAGE IN MEMORY (блокировка страницы в памяти)

- действие, в результате которого страница процесса становится недоступной для обмена страниц или обмена рабочих наборов процессов. Страница остается заблокированной в памяти до тех пор, пока операционная система ее специально не разблокирует.

LOCKING A PAGE IN THE WORKING SET (блокировка страницы в рабочем наборе)

- действие, в результате которого страница процесса становится недоступной для выгрузки из рабочего набора процесса. Эта страница может быть выгружена при обмене рабочих наборов процессов. Страница остается заблокированной в рабочем наборе до тех пор, пока она не будет специально разблокирована.

LOG (регистрирующая запись)

- запись о характеристиках работы.

LOGGING (регистрация)

- элемент управления сетью, который маршрутизирует данные событий к регистрирующим приемникам, например, к консоли или к файлу.

LOGGING FILE (файл регистрации)

- регистрирующий приемник, который должен получить используемую впоследствии машинно-читаемую запись о событиях. Файл регистрации определяется пользователем.

LOGGING IN (вход в систему)

- опознавание пользователя в операционной системе. При входе в систему пользователи печатают учетное имя и пароль в ответ на соответствующие подсказки системы. Если имя и пароль соответствуют учетной информации, имеющейся в системе, то пользователю разрешается обращаться к данной учетной информации.

LOGGING CONSOLE (консольный регистрационный приемник)

- регистрирующий приемник, который должен получить запись о событиях в форме, понятной пользователям системы. Как правило, регистрационным приемником является терминал или указанный пользователем файл.

LOGGING MONITOR (регистрационный монитор)

- регистрирующий приемник, который должен получить машинно-читаемую запись о событиях с целью возможного выбора последующих действий в реальном времени. Как правило, регистрационный монитор является определенной пользователем программой.

LOGGING OUT (выход из системы)

- ввод команды LOGOUT языка DCL, которая информирует операционную систему о том, что пользователь завершил работу на данном терминале.

LOGICAL BLOCK NUMBER (LBN) (номер логического блока)

- связанный с томом адрес блока на устройстве большой емкости. Блоки, из которых составлен том, помечены последовательно, начиная с нулевого логического блока. См. Также PHYSICAL BLOCK NUMBER (номер физического блока).

LOGICAL CHANNEL (логический канал)

- логическое соединение между DTE и DCE. Физические линии коммуникаций между DTE и DCE делятся на ряд логических каналов.

LOGICAL CHANNEL NUMBER (LCN) (номер логического канала)

- уникальный номер, который определяет логический канал. Оборудование DTE распознает виртуальную цепь по связанному с ним номеру LCN.

LOGICAL I/O FUNCTION (функции логического уровня ввода-вывода)

- набор операций ввода-вывода (например, читать и писать логический блок), которые, используя адреса логических блоков, позволяют ограничивать непосредственный доступ к операциям ввода-вывода на уровне устройств.

LOGICAL LINK (логический канал)

- 1) маршрут связи между программами двух узлов сети.

- 2) несущая частота единственного потока дуплексной связи между двумя процессами пользовательского уровня.

LOGICAL NAME (логическое имя)

- имя, определенное пользователем для любой части или всей спецификации файла. Определения логических имен сохраняются в таблицах логических имен каждого процесса, каждой группы и системы. Логическим именам могут быть приписаны такие атрибуты трансляции, как терминальность или сокрытость. См. Также SEARCH LIST (список поиска).

LOGICAL NAME TABLE (таблица логических имен)

- таблица, которая содержит набор логических имен и соответствующие им эквивалентные имена для конкретного процесса, конкретной группы или системы.

LOGICAL RECORD (логическая запись)

- группа связанных полей, трактуемая как элемент.

LOGIN FILE (файл входа в систему)

- командная процедура, выполняемая автоматически при входе пользователя в систему и в начале пакетного задания.

LONGWORD (длинное слово)

- четыре смежных байта (32 бита), начинающиеся на границе адресуемого байта. Биты нумеруются справа налево от 0 до 31. Адресом длинного слова является адрес байта, содержащего бит 0. При арифметической интерпретации длинное слово является дополнительным кодом

00152-01 31 01

целого числа, вес битов в котором увеличивается от бита 0 до бита 30. Если длинное слово интерпретируется как целое со знаком, то бит 31 является знаковым битом. Значение целого числа со знаком находится в диапазоне от -2147483648 до 2147483647. Если длинное слово интерпретируется как беззнаковое целое, то вес битов увеличивается от бита 0 до бита 31. Значение беззнакового целого находится в диапазоне от 0 до 4894967295.

LOOP NODE (замкнутый узел)

- локальный узел, связанный с конкретной линией и трактуемый как удаленный узел.

MAILBOX (почтовый ящик)

- структура данных программного обеспечения, которая трактуется как устройство, ориентированное на записи для общей связи между процессами. Связь с использованием почтового ящика аналогична другим видам независимого от устройств ввода-вывода. Отправитель записывает информацию в почтовый ящик, получатель читает ее из почтового ящика. Определены некоторые общесистемные почтовые ящики; информацию из них читает регистратор ошибок и процесс OPCOM.

MAIN MEMORY (основная память)

- см. PHYSICAL MEMORY (физическая память)

MAIN TEXT BUFFER (основной текстовый буфер)

- в редакторе EDT - это подразумеваемый по умолчанию текстовый буфер для ввода символов клавиатуры, вход-

00152-01 31 01

ных файлов, а также источник выходных файлов.

MANUAL RECORD UNLOCKING (неавтоматическое освобождение записей)

- возможность СУД-32, позволяющая пользователям захватывать в файл одновременно несколько записей. Пользователь управляет откреплением записей явным образом. Захват записей происходит в том случае, когда бит RABV_ULK поля возможностей обработки записей задает выполнение программ обслуживания GET (получить), FIND (искать), PUT (поместить). Если освобождение записи разрешено и происходит ее захват, то она будет захвачена до явного освобождения программами обслуживания FREE (освободить) или RELEASE (снять), или до окончания потока.

MAPPING WINDOW (окно отображения)

- часть информации, применяемой для поиска файла, она используется для трансляции номеров виртуальных блоков в номера логических блоков.

MASS STORAGE DEVICE (устройство большой емкости)

- устройство ввода-вывода, хранящее данные и другие типы файлов до тех пор, пока они не будут использованы. Типичными устройствами большой емкости являются диски, магнитные ленты и гибкие диски.

MAXIMUM VISITS (максимальное число транзитных узлов)

- максимальное число узлов, через которые может маршрутизироваться пакет на пути к пункту назначения.

MEMBER NUMBER (номер члена)

- второе число или его буквенно-цифровой эквивалент в коде идентификации пользователя, оно однозначно определяет данный код.

MEMORY (память)

- ряд физических ячеек, предназначенных для размещения данных или инструкций в виде двоичных слов. Каждая ячейка памяти может иметь адрес, а ее содержимое может быть изменено.

MEMORY MANAGEMENT (управление памятью)

- функции операционной системы, к которым относятся отображение и защита страниц (аппаратные компоненты), активизатор образов и программа обмена страниц (компоненты операционной системы).

MODEM (модем)

- сокращение от понятия модулятор/демодулятор. Устройство, которое модулирует сигналы, пересылаемые по аппаратуре связи, и демодулирует их перед получением.

MODIFY ACCESS TYPE (тип доступа на модификацию)

- тип доступа, при котором во время выполнения инструкции или процедуры специфицированный операнд читается, возможно, модифицируется и записывается.

MODULE (модуль)

- 1) часть программы или библиотеки программ, например, исходный модуль, объектный модуль или модуль образа;

00152-01 31 01

- 2) плата, на которой смонтированы логические устройства;
- 3) элемент управления сетью.

MONITOR CONSOLE ROUTINE (MCR) (программа связи с оператором)

- интерпретатор командного языка в системе ОС РВ, а также дополнительный интерпретатор командного языка в системе МОС ВП.

MOUNT A VOLUME (монтировать том)

- 1) логически связать том с тем физическим устройством, на котором он находится (действие, выполняемое системными программами по запросу оператора).
- 2) загрузить или поместить магнитную ленту или пакет дисков на накопитель и подключить накопитель к линии (действие, выполняемое оператором системы).

MOVE MODE (режим пересылки)

- метод доступа при вводе-выводе записей СУД-32, при котором программа осуществляет доступ к записям в своей собственной рабочей области памяти. См. LOCATE MODE (режим захвата).

MULTIACCESS CHANNEL (канал параллельного доступа)

- среда (например, магистраль ETHERNET), за доступ к которой конкурируют множество передатчиков.

MULTICAST ADDRESSING (многоточечная адресация)

- режим адресации, в котором данный пакет сообщений направляется в группу логически связанных узлов.

MULTICAST GROUP ADDRESS (многоточечный групповой адрес)

- адрес, приписанный ряду узлов магистрали ETHERNET и используемый для пересылки сообщений на все узлы группы за одну передачу.

MULTINATIONAL CHARACTER SET (многонациональный набор символов)

- набор восьмибитовых символов, содержащий множество буквенно-цифровых символов, включая символы с диакритическими знаками.

MULTIPOINT CIRCUIT (цепь в многоточечной системе)

- цепь, соединяющая две системы; одна из систем (управляющая станция) управляет цепью, а другая служит в качестве подчиненной станции.

MUTEX (взаимное исключение)

- семафор, используемый для управления исключаящим доступом к той области программы, которая может разделять структуру данных или другой ресурс. Взаимное исключение гарантирует, что в некоторый момент к этой области программы имеет доступ лишь один процесс.

NAME BLOCK (NAM) (блок имени файла)

- пользовательская структура данных СУД-32, которая содержит дополнительную информацию, используемую при синтаксическом разборе спецификации файла.

NATIVE IMAGE (образ основного режима)

- образ, инструкции которого выполняются в основном режиме.

00152-01 31 01

NATIVE MODE (основной режим)

- режим работы процессора, в котором выполняются инструкции ВК СМ 1700.

NETWORK (сеть)

- набор взаимосвязанных, отдельных вычислительных систем.

NETWORK CONNECT BLOCK (NCB) (блок соединений сети)

- для систем "трал" или "магистр" NCB представляет собой сгенерированную пользователем структуру данных, применяемую непрозрачной задачей с целью идентификации удаленной задачи и дополнительной пересылки пользовательских данных в вызовы процедур для запроса, приема или отказа на соединение по логическому каналу.

NETWORK OBJECT (сетевой объект)

- любая задача с ненулевым типом объекта. К таким задачам относятся, например, программы FAL, NML, которые обеспечивают общее обслуживание через сеть.

NETWORK STATUS NOTIFICATIONS (уведомления о состоянии сети)

- уведомления с информацией о состоянии логических и физических каналов, по которым взаимодействуют две задачи. Эта информация может быть использована непрозрачной задачей для выполнения соответствующего действия при таких условиях, как отсоединение третьей стороны и выход партнера из взаимодействия перед завершением ввода-вывода.

NETWORK TASK (сетевая задача)

- непрозрачная задача, которая способна обрабатывать множество запросов на соединение, т.е. задача, объявившая имя сети или номер объекта.

NIBBLE (тетрада)

- старшие или младшие четыре бита в байте.

NODE (узел)

- отдельная вычислительная система в сети, которая может взаимодействовать с другими вычислительными системами данной сети.

NODE ADDRESS (адрес узла)

- обязательное однозначное числовое обозначение, необходимое для каждого узла сети.

NODE NAME (имя узла)

- необязательное буквенно-цифровое обозначение, связанное с адресом узла в точном взаимнооднозначном отображении. Имя узла должно содержать по крайней мере один буквенный символ.

NODE SPECIFICATION (спецификация узла)

- первое поле в спецификации файла, которое определяет местоположение вычислительной системы в сети.

NONPRINTING CHARACTER (непечатаемый символ)

- символ в кодовом наборе вычислительной машины, для которого не существует соответствующего графического символа.

NONPRIVILEGED (непривилегированный)

- в терминологии систем "трал" или "магистр" этот тер-

00152-01 31 01

мин означает отсутствие привилегий, за исключением привилегии NETMSX, которая представляет собой минимальное необходимое условие для каких-либо сетевых действий.

NONROUTING NODE (немаршрутный узел)

- конечный узел.

NULL (пусто, отсутствие информации)

- 1) строка 000 в коде КОИ-8.
- 2) отсутствие информации.

NULL PROCESS (пустой процесс)

- малый системный процесс, который имеет самый низкий приоритет в системе и занимает целиком один класс приоритетов. Единственная функция пустого процесса заключается в том, чтобы потреблять свободное время процессора.

NULL STRING (пустая строка)

- строка без содержимого или строка, представленная стоящими рядом кавычками.

NUMERIC STRING (числовая строка)

- непрерывная последовательность байтов, представляющая до 31 десятичной цифры (по одной в байте) и, возможно, знак. Числовая строка определяется младшей адресуемой ячейкой, длиной и представлением знака.

ОБЪЕКТ (объект, объектный)

- 1) процесс системы "трал", который получает запрос от логического канала. Он выполняет специфичную сетевую функцию или представляет собой определенный

00152-01 31 01

пользователем образ прикладной задачи специального назначения.

- 2) элемент управления, содержащий записи для определения учетной информации поступающих вызовов и для определения командной процедуры, которая инициализируется при поступлении вызова.
- 3) системный ресурс, например, файл, устройство или каталог.

OBJECT MODULE (объектный модуль)

- двоичная выходная информация транслятора, которая используется в качестве входной информации для компоновщика.

OBJECT TYPE (тип объекта)

- дискретный идентификатор для некоторой задачи или программы обслуживания, входящих в состав систем "трал" или "магистр" на удаленном узле. Идентификаторы типов объектов могут представлять собой или 0 плюс имя (альтернативно, TASK=имя), или ненулевое значение без имени (например, 17= или FAL=).

OCTAWORD (октаслово, увосьмеренное слово)

- 16 смежных байтов (128 бит), начинающихся на границе адресуемого байта. Биты нумеруются справа налево, от 0 до 127. Октаслово определяется адресом байта, содержащего самый младший бит (бит 0).

OFFSET (смещение)

- фиксированный сдвиг от начала структуры данных. Системное смещение для элементов некоторой структуры

00152-01 31 01

обычно имеет соответствующее символическое имя, используемое вместо числового сдвига. Там, где определены символические имена, программисты всегда обращаются к элементам структуры данных по этим именам, а не по числовому сдвигу.

ON DISK STRUCTURE LEVEL 1 (ODS-1) (файловая структура уровня 1)

- см. FILES-SM STRUCTURE LEVEL 1 (файловая структура уровня 1).

ON DISK STRUCTURE LEVEL 2 (ODS-2) (файловая структура уровня 2)

- см. FILES-SM STRUCTURE LEVEL 2 (файловая структура уровня 2).

OPCODE (код операции)

- набор битов в инструкции, определяющий операцию, которую надо выполнить.

OPEN ACCOUNTS (открытые учетные разделы)

- учетные разделы, не требующие паролей.

OPERAND SPECIFIER (спецификатор операнда)

- набор битов в инструкции, который указывает режим адресации и регистр или смещение, определяющее операнд инструкции.

OPERAND SPECIFIER TYPE (тип спецификатора операнда)

- тип доступа и тип данных операнда инструкции. Например, поскольку инструкции TEST лишь считывают значения операнда, то они имеют тип доступа на чтение.

00152-01 31 01

тип данных операнда может быть байт, слово или длинное слово в зависимости от того, для какой инструкции TSTB, TSTW или TSTL задан код операции.

OPERATING SYSTEM (операционная система)

- объединенный набор программ, который управляет работой вычислительного комплекса и выполняет системные функции.

OPERATOR COMMUNICATION MANAGER (OPCOM) (процесс сообщений оператору)

- системный процесс, получающий входную информацию от процесса, которому надо сообщить оператору об определенном состоянии или условии, передающий сообщение оператору и выдающий сообщение. Процесс OPCOM всегда находится в активном состоянии.

OPERATOR'S CONSOLE (консоль оператора)

- любой терминал, определенный как терминал, за которым работает оператор системы.

OUTBOUND CONNECTION (запрос на внешнее соединение)

- запрос задачи на соединение по логическому каналу с другим узлом.

OUTPUT FILE (выходной файл)

- файл, который содержит результаты операции обработки, например, отсортированный или отредактированный файл.

OWNER (владелец)

- в контексте "система/владелец/группа/все" владелец - это конкретный член (группы), которому принадлежит

00152-01 31 01

файл, глобальная секция, почтовый ящик или кластер флагов событий.

OWNER PROCESS (процесс-владелец)

- процесс или подпроцесс, который создал подпроцесс.

PACKED DECIMAL (упакованный формат десятичных чисел)

- представление десятичного числа парой десятичных цифр, хранящихся в одном байте. В этом методе использовано то обстоятельство, что для представления чисел от 0 до 9 требуется лишь четыре бита.

PACKED DECIMAL STRING (упакованная десятичная строка)

- непрерывная последовательность байтов (до 16), интерпретируемая как строка из четырехбайтовых полей. Каждое поле представляет цифру, исключение составляют четыре младших бита в байте со старшим адресом, которые представляют знак. Упакованная десятичная строка определяется ячейкой с младшим адресом и числом цифр.

PACKET. (пакет)

- блок данных, который должен маршрутизироваться от исходного узла к приемному узлу. Обычно поле данных пользователя сопровождается заголовком с адресатом данных и с другой информацией.

PACKET ASSEMBLY/DISASSEMBLY (PAD) FACILITY (средство сборки/разборки пакетов)

- устройство узла в сети PDSN, к которому разрешен доступ с асинхронного терминала. Такой терминал подсоединяется к средству PAD, и оно собирает в пакеты

00152-01 31 01

(ассемблирует) входные данные, поступающие с терминала, а также забирает из пакетов (деассемблирует) выходные данные.

PACKET SWITCHING (коммутация пакетов)

- процесс передачи данных, использующий адресованные пакеты, благодаря чему канал бывает занят только во время передачи пакета.

PACKET SWITCHING DATA NETWORK (PSDN) (сеть с коммутацией пакетов данных)

- совокупность оборудования и внутренних каналов, которая предоставляет абонентам услуги по передаче коммутируемых пакетов.

PAGE (страница)

- 1) условная единица деления непрерывного пространства памяти на поля фиксированной длины (512 байтов). Используется как единица отображения и защиты памяти.
- 2) данные, размещаемые между началом файла и маркером страницы, между двумя маркерами и между маркером и концом файла.

PAGE FAULT (страничный отказ)

- исключительная ситуация, возникающая при обращении к странице, которая не входит в рабочий набор процесса, вызвавшего отказ.

PAGE FAULT CLUSTER SIZE (размер кластера при страничном отказе)

00152-01 31 01

- количество страниц, считываемых в память при страничном отказе.

PAGE FRAME NUMBER (PFN) (номер физической страницы)

- 21 старших бита физического адреса страницы, находящейся в физической памяти.

PAGE FRAME NUMBER MAPPING (PFN MAPPING) (отображение номера физической страницы)

- отображение некоторой секции в одну или несколько страниц физической памяти или в пространство ввода-вывода (противопоставляется отображению секции в дисковый файл).

PAGE MARKER (маркер страницы)

- символ или символы (как правило, это перевод формата), которые разделяют страницы файла, обработанного текстовым редактором.

PAGE TABLE ENTRY (PTE) (элемент таблицы страниц)

- структура данных, определяющая физическое размещение и состояние страницы виртуального адресного пространства. Если виртуальная страница находится в основной памяти, то PTE содержит номер физической страницы, необходимый для отображения виртуальной страницы в физическую страницу. Если же страница не находится в основной памяти, то элемент таблицы страниц содержит информацию, необходимую для ее поиска во внешней памяти (на диске).

PAGER (программа обмена страниц)

- набор процедур режима ядра, который выполняется при

страничном отказе. Программа обмена страниц делает доступной в физической памяти ту страницу, на которой произошел страничный отказ, после чего образ может продолжить свою работу.

PAGING (обмен страниц)

- действие переноса страниц выполняющегося процесса из внешней памяти на диск в физическую память, когда процесс обращается к ним. В МОС ВП обмен страниц процесса выполняется в том случае, когда этот процесс обращается к большему количеству страниц, чем ему разрешено иметь в его рабочем наборе, или же, когда процесс первый раз активизирует образ в памяти. При обращении процесса к странице, которой нет в рабочем наборе, происходит страничный отказ. Он приводит к тому, что системная программа обмена страниц считывает в память данную страницу, если она находится на диске (и, необязательно, другие связанные с ней страницы, в зависимости от фактора кластера), заменяя по мере необходимости те страницы, отказ по которым произошел наиболее давно. Обмен страниц процесса производится только за счет самого процесса.

PARAMETER (параметр)

- 1) значение, поступившее в командную процедуру и эквивалентное некоторому символу в диапазоне от P1 до P8. См. Также COMMAND PARAMETERS (параметр команды).
- 2) запись в постоянной или непостоянной базах данных

00152-01 31 01

для элемента управления сетью.

PARSING (синтаксический Анализ)

- 1) разложение командной строки на элементы для ее интерпретации.
- 2) интерпретация спецификации файла, выполняемая СУД-32.

PASSWORD (пароль)

- символьная строка, используется для защиты программ и данных пользователя и системы от несанкционированного доступа. Вводится пользователем при входе в систему и в процессе сеанса работы. Существуют два вида паролей - системные и пользовательские. К паролям пользователей относятся основной и вторичный пароли.

PASTE BUFFER (буфер для склеек)

- подразумеваемый по умолчанию текстовый буфер для операций резки и склейки.

PATH (путь, маршрут)

- маршрут пакетных задач от источника до приемника.

PATH COST (стоимость маршрута)

- сумма стоимостей цепей вдоль маршрута между двумя узлами.

PATH LENGTH (длина маршрута)

- число транзитных участков вдоль маршрута между двумя узлами, т.е. число цепей, по которым должен пройти пакет до приемника.

PERIPHERAL DEVICES (периферийные устройства)

- любое устройство, отличное от CPU и физической памяти, которое может выдать системе входную информацию или получить от нее выходную информацию. Терминалы, построочно-печатающие устройства и диски являются периферийными устройствами.

PERMANENT DATABASE (постоянная база данных)

- файл, содержащий информацию об элементах управления сетью. См. Также VOLATILE DATABASE (непостоянная база данных) и CONFIGURATION DATABASE (база данных конфигурации).

PERMANENT VIRTUAL CIRCUIT (PVC) (постоянная виртуальная цепь)

- постоянная логическая связь между двумя DTE, она аналогична арендуемой линии. Пакеты маршрутизируются непосредственно по сети от одного DTE к другому.

PER-PROCESS ADDRESS SPACE (адресное пространство процесса)

- см. PROCESS ADDRESS SPACE (адресное пространство процесса)

PHYSICAL ADDRESS (физический адрес)

- 1) адрес, используемый аппаратурой для определения ячейки в физической памяти или на устройствах непосредственно адресуемой внешней памяти, например, на дисках. Адрес физической памяти состоит из номера физической страницы и номера байта в странице. Физический адрес блока на диске состоит из номера цилиндра или дорожки и номера сектора.

00152-01 31 01

- 2) уникальное значение адреса, связанное с данной системой в цепи ETHERNET. Физический адрес ETHERNET по определению должен отличаться от всех других физических адресов ETHERNET.

PHYSICAL ADDRESS SPACE (пространство физических адресов)

- набор всех возможных 30-битных физических адресов, которые могут быть использованы для обращения к ячейкам памяти (пространству памяти) или к регистрам устройств (пространству ввода-вывода).

PHYSICAL BLOCK NUMBER (номер физического блока)

- физический адрес блока на устройстве большой емкости. Противопоставляется понятиям LOGICAL BLOCK NUMBER (номер логического блока) и VIRTUAL BLOCK NUMBER (номер виртуального блока).

PHYSICAL I/O FUNCTIONS (функции физического уровня ввода-вывода)

- множество функций ввода-вывода, которые предоставляют доступ ко всем операциям ввода-вывода на уровне устройств, кроме операций режима сохранения.

PHYSICAL LINK (физический канал)

- передающий сигналы носитель, который связывает два узла в сети. Противопоставляется понятию LOGICAL LINK (логический канал).

PHYSICAL MEMORY (физическая память)

- модули памяти, используемые для хранения как инструкций, которые процессор может сразу выбрать и выполнить, так и данных, которые управляют процес-

00152-01 31 01

сорной обработкой. Физическая память имеет также название MAIN MEMORY (основная память).

POINT-TO-POINT CIRCUIT (цепь точка-точка)

- цепь, которая соединяет два узла и реализует передачу информации по единственной линии.

POLLING (разрешение передачи)

- действие, выполняемое управляющей станцией по отношению к подчиненным станциям цепи многоточечной системы для того, чтобы дать им разрешение на передачу информации.

POSITION-DEPENDENT CODE (позиционно-зависимая программа)

- программа, которая может выполнить свои функции правильно лишь при размещении в виртуальном адресном пространстве, приписанном ей компоновщиком.

POSITION-INDEPENDENT CODE (позиционно-независимая программа)

- программа, которая без модификации может выполнить свои функции правильно, независимо от того, где она размещена в виртуальном адресном пространстве (даже если ее местоположение изменено после компоновки). Обычно такая программа использует режимы адресации, которые формируют исполнительный адрес относительно счетчика РС.

PRIMARY KEY (первичный ключ)

- обязательный ключ в записях данных индексного файла; используется СУД-32 для определения местоположения

00152-01 31 01

записей в файле и для построения основного индекса. См. KEY (INDEXED FILE) (ключ индексного файла) и ALTERNATE KEY (альтернативный ключ).

PRIMARY PASSWORD (основной пароль)

- тип пароля, который запрашивается у пользователя первым. Система может дополнительно затребовать и вторичный пароль. Он, как и всякий пароль пользователя, должен быть паролем, который привязывается к имени пользователя, т.е. вводится вместе с ним.

PRIMARY VECTOR (первичный вектор)

- ячейка с начальным адресом той программы обработки исключительного состояния, которая должна выполняться при его возникновении.

PRIORITY (приоритет)

- ранг, приписанный процессу для определения его старшинства при получении ресурсов системы.

PRIVATE SECTION (личная секция)

- секция образа процесса, которая не разделяется между процессами. См. Также GLOBAL SECTION (глобальная секция).

PRIVILEGED (привилегированный)

- этот термин обычно относится к инструкциям, образам, учетным разделам, которые используются операционной системой, специальными системными программами или отдельными пользователями системы.

PRIVILEGES (привилегии)

- средства защиты при использовании некоторых систем-

00152-01 31 01

ных функций, которые могут воздействовать на ресурсы системы и/или на ее целостность. Приписывание привилегий пользователям в соответствии с их потребностями или открепление привилегий выполняет системный администратор. См. Также PROCESS PRIVILEGES (привилегии процесса), USER PRIVILEGES (привилегии пользователя), IMAGE PRIVILEGES (привилегии образа).

PROCEDURE (процедура)

- программа, запускаемая инструкцией CALL (звон). См. также COMMAND PROCEDURE (командная процедура).

PROCESS (процесс)

- основная единица, планируемая системными программами и обеспечивающая контекст, в котором выполняется образ. Процесс состоит из адресного пространства, а также аппаратного и программного контекста.

PROCESS ADDRESS SPACE (адресное пространство процесса)

- см. PROCESS SPACE (пространство процесса).

PROCESS CONTEXT (контекст процесса)

- аппаратный и программный контексты процесса.

PROCESS CONTROL BLOCK (PCB) (блок управления процессом)

- структура данных, используемая для хранения контекста процесса. Аппаратный блок PCB содержит аппаратный контекст. Программный блок PCB содержит программный контекст, который включает указатель на аппаратный PCB.

PROCESS HEADER (заголовок процесса)

- структура данных, которая содержит аппаратный блок

00152-01 31 01

PCB, учетный раздел, информация о квотах, таблицу секций процесса, список рабочего набора и таблицы страниц, определяющие виртуальное размещение процесса.

PROCESS HEADER SLOTS (слоты для заголовков процессов)

- часть адресного пространства системы, в котором хранятся заголовки процессов, входящих в балансный набор. Число слот в системе определяет число процессов, которые могут одновременно входить в балансный набор.

PROCESS IDENTIFICATION (PID) (идентификатор процесса)

- 32-битное двоичное значение, которое однозначно определяет процесс. Каждый процесс имеет идентификатор и имя.

PROCESS I/O CHANNEL (канал ввода-вывода процесса)

- см. CHANNEL (канал).

PROCESS I/O SEGMENT (сегмент ввода-вывода процесса)

- та часть области управления процесса, которая содержит, во-первых, блок внутреннего (для СУД-32) доступа к каждому открытому файлу, относящемуся к постоянному процессу, и, во-вторых, буфера ввода-вывода, включая буфер команд соответствующего интерпретатора и дескрипторы команд.

PROCESS NAME (имя процесса)

- символьная строка, содержащая от 1 до 15 символов в коде КОИ-8, которую можно использовать для идентификации процессов, выполняющихся под одним и тем же

номером группы.

PROCESS PAGE TABLES (таблицы страниц процесса)

- таблицы страниц, используемые для описания виртуальной памяти процесса.

PROCESS PRIORITY (приоритет процесса)

- приоритет, приписанный процессу в целях планирования. Операционная система распознает 32 уровня приоритетов процессов, где 0 - это низший, а 31 - высший уровень. Уровни от 16 до 31 используются для процессов реального времени. Система не модифицирует приоритет такого процесса (хотя это могут сделать администратор системы или сам процесс). Уровни от 0 до 15 используются для обычных процессов. Система может временно увеличить приоритет такого процесса, основываясь на его активности.

PROCESS PRIVILEGES (привилегии процесса)

- привилегии, предоставляемые процессу системой. Эти привилегии являются комбинацией привилегий пользователей и привилегий образов. Они, например, включают следующие привилегии: воздействовать на другие процессы той же группы, что и группа пользователя; воздействовать на любой процесс системы, независимо от UID; создавать постоянные кластеры флагов событий; создавать другой процесс; создавать почтовые ящики; выполнять прямой ввод-вывод на устройстве с файловой структурой; выполнять сетевые операции.

00152-01 31 01

PROCESS SECTION (секция процесса)

- см. PRIVATE SECTION (личная секция).

PROCESS SPACE (пространство процесса)

- половина адресного пространства с младшими адресами, в которой размещаются инструкции и данные. Пространство процесса делится на область программы и область управления.

PROCESSOR REGISTER (регистр процессора)

- 32-разрядная ячейка внутренней памяти процессора, используемая программами операционной системы для управления состояниями при работе вычислительной системы. К регистрам процессора относятся, например, регистр длины и базовый регистр системы, базовый регистр и регистр длины области управления и области программы, базовый регистр блока управления системой и регистр запросов программных прерываний.

PROCESSOR STATUS LONGWORD (PSL) (длинное слово состояния процессора)

- привилегированный регистр процессора, состоящий из слова привилегированного состояния процессора и слова состояния процессора. Информация о привилегированном состоянии процессора включает текущий уровень приоритета прерываний, предыдущий режим доступа, текущий режим доступа, бит стека прерываний, бит подавления трассировки и бит режима совместимости.

PROCESSOR STATUS WORD (PSW) (слово состояния процессора)

- младшее слово длинного слова состояния процессора.

00152-01 31 01

информация о состоянии процессора включает коды условий (перенос, переполнение, ноль, отрицательное число), биты разрешения арифметических ловушек (целое переполнение, машинный ноль) и бит разрешения трассировки.

PROGRAM (программа)

- последовательность инструкций, предназначенная для получения конкретного результата.

PROGRAM COUNTER (PC) (счетчик инструкций)

- общий регистр 15 (R15). В начале выполнения инструкций PC обычно содержит адрес ячейки памяти, из которой процессор будет выбирать следующую выполняемую инструкцию.

PROGRAM LOCALITY (локальность программы)

- характеристика программы, показывающая насколько близко или далеко размещены в виртуальной памяти ячейки, к которым обращается программа за некоторый период времени. Программа с высокой степенью локальности не обращается за короткий промежуток времени ко многим далеко отстоящим друг от друга виртуальным адресам.

PROGRAM REGION (PO) (область программы)

- половина адресного пространства процесса с младшими адресами (область PO). Область программы содержит текущий, выполняемый в рамках процесса образ и другие программы пользователя, вызванные этим образом.

PROGRAM REGION BASE REGISTER (POBR) (базовый регистр области программы)

- регистр процессора или его эквивалент в аппаратном блоке управления процессом, содержащий базовый виртуальный адрес того элемента таблицы страниц, который задает виртуальную страницу с номером 0 в области программы данного процесса.

PROGRAM REGION LENGTH REGISTER (POLR) (регистр длины области программы)

- регистр процессора или его эквивалент в аппаратном блоке управления процессом, который содержит количество элементов в таблице страниц области программы данного процесса.

PROGRAM SECTION (PSECT) (программная секция)

- часть программы с заданной защитой и набором атрибутов управления памятью. Программные секции, имеющие одинаковые атрибуты, собираются компоновщиком вместе и образуют секцию образа.

PROGRAMMER NUMBER (номер программиста)

- см. MEMBER NUMBER (номер члена).

PROJECT NUMBER (номер проекта)

- см. GROUP NUMBER (номер группы) или ACCOUNT NUMBER (учетный номер).

PROMPT (подсказка)

- символ, указывающий, что пользователь должен ввести информацию.

PROTECTION (защита)

- атрибуты некоторого ресурса, которые лимитируют имеющийся у пользователя тип доступа к данному ресурсу. См. Также USER IDENTIFICATION CODE (код идентификации пользователя) и ACCESS CONTROL LIST (список управления доступом).

PROTOCOL (протокол)

- согласованный набор правил, управляющий работой канала связи.

PROXY LOGIN (вход в систему по доверенности)

- тип входа в систему, позволяющий пользователям удаленного узла входить в систему локального узла так, как будто этот пользователь является владельцем учетного раздела на локальном узле. Однако, в строке управления доступом пользователь не указывает пароль. В некоторых случаях удаленный пользователь создает собственный учетный раздел, а в других случаях он обращается к такому разделу совместно с другими пользователями.

PSEUDO DEVICE (псевдоустройство)

- объект, трактуемый системой или пользователем как устройство ввода-вывода, хотя оно и не является некоторым отдельным физическим устройством. Псевдоустройство - это пересылающий адрес, посредством которого можно достичь реального физического устройства. См. PROCESSOR STATUS LONGWORD (длинное слово состояния процессора).

00152-01 31 01

PURE CODE (чистый код)

- см. REENTRANT CODE (реентерабельный код).

QIO (поставить запрос в очередь ввода-вывода)

- программа системного обслуживания очереди запросов ввода-вывода, которая входит в состав МОС ВП и управляет запросами `QIO` и `QIOW`. Эта программа готовит запрос ввода-вывода для обработки драйвером и выполняет независимую от устройств предобработку данного запроса. Она также вызывает программы FDT драйвера.

QUADWORD (квадрослово)

- четыре смежных слова (64 бита), начинающиеся на границе любого адресуемого байта. Биты нумеруются от 0 до 63 справа налево. Квадрослово определяется адресом слова, содержащего младший бит (бит 0). При арифметической интерпретации квадрослово является целым числом в дополнительном коде с увеличением веса битов от бита 0 до бита 62. Бит 63 используется как знак. Значение целого числа лежит в диапазоне от $2^{**(-63)}$ до 2^{**63} .

QUALIFIER (квалификатор)

- часть командной строки, которая путем выбора одной или нескольких возможностей изменяет действие команды или ее параметр. Квалификатор, если он есть, следует за командой или параметром, к которому он относится, и имеет вид /квалификатор[=возможность].

QUANTUM (шаг квантования)

- минимальное количество времени, в течении которого процесс может оставаться в основной памяти; кроме того, это максимальное количество времени, в течении которого процесс может выполняться. Всякий раз, когда процесс входит в состояние ожидания, из шага квантования вычитается определенная величина.

QUEUE (очередь)

- последовательность заданий, которые необходимо обработать (например, очередь пакетных заданий или очередь заданий печати). Обработка выполняется в основном по дисциплине FIFO, но при этом обязательно учитывается приоритет того процесса, которому подчинено задание. См. Также STATE QUEUE (очередь процессов по состоянию) и SYSTEM QUEUE (системная очередь).

QUEUE PRIORITY (приоритет в очереди)

- приоритет, приписанный заданию, помещенному в очередь спула или в пакетную очередь.

QUOTA (квота)

- общее количество системного ресурса, например, времени CPU, которое за учетный период разрешено использовать некоторому заданию. Это указывается системным администратором в файле авторизации пользователя.

RANDOM ACCESS (произвольный доступ)

- термин, относящийся к памяти или к устройству большой емкости с равнодоступной информацией. При произ-

00152-01 31 01

вольном доступе следующая ячейка, из которой надо получить данные, не связана с ячейкой, из которой перед этим данные были получены.

RANDOM ACCESS BY KEY (произвольный доступ по ключу)

- поиск или хранение записи путем определения значения ключа. Этот метод применяется только к индексным файлам.

RANDOM ACCESS BY RECORD'S FILE ADDRESS (произвольный доступ по адресу записи в файле)

- поиск или хранение записи по ее уникальному адресу, который выдается программе системой СУД-32 при успешном выполнении операций #GET или #FIND. Впоследствии адрес записи в файле может быть использован для произвольного доступа к этой записи.

RANDOM ACCESS BY RELATIVE RECORD NUMBER (произвольный доступ по относительному номеру записи)

- поиск или хранение записи с помощью определения ее положения относительно начала файла. Этот метод применяется только к последовательным файлам с записями фиксированной длины и к относительным файлам.

RANGE SPECIFICATION (спецификация диапазона)

- средство идентификации текстовой строки в редакторе EDT.

REACHABLE NODE (доступный узел)

- некоторый узел, к которому локальный узел имеет удобный путь связи.

00152-01 31 01

READ ACCESS TYPE (тип доступа на чтение)

- атрибут операнда инструкции или процедуры, указывающий, что при выполнении данной инструкции или процедуры специфицированный операнд лишь читается.

REAL-TIME PROCESS (процесс реального времени)

- процесс, который реагирует на события в связанных с ним (или управляемых им) процессах, не когда вычислительная машина готова на них реагировать, а когда они происходят. Тем самым, результаты вычислений могут использоваться при обработке. Процессу реального времени присписывается уровень программного приоритета в интервале от 16 до 31 включительно. Запланированный приоритет, присписанный процессу реального времени, никогда планировщиком не меняется, хотя он может быть изменен системным администратором или самим процессом.

RECORD (запись)

- набор связанных данных, трактуемых программой, как некоторая единица.

RECORD ACCESS BLOCK (RAB) (блок доступа к записям)

- пользовательский блок управления в СУД-32, выделяемый во время ассемблирования или выполнения программы для связи с СУД-32. Он описывает записи в конкретном файле и взаимодействует с блоком доступа к файлу с целью формирования потока доступа к записям. Блок RAB определяет характеристики, необходимые для выполнения таких связанных с записями операций, как

00152-01 31 01

корректировка, удаление или получение записей.

RECORD ACCESS MODE (метод доступа к записям)

- метод, используемый СУД-32 для поиска и хранения записей в файле. Доступ осуществляется одним из четырех методов: последовательным, произвольным по ключу, произвольным по адресу записи в файле и произвольным по относительному номеру записи.

RECORD ACCESS MODE SWITCHING (переключение метода доступа к записям)

- термин, применяемый по отношению к переключению с одного метода доступа к записям на другой во время обработки файла.

RECORD BLOCKING (блокирование записей)

- метод группирования множества записей в один блок. На магнитной ленте межблочный промежуток помещается не после каждой записи, а после блока. Этот метод уменьшает число пересылок информации ввода-вывода, требуемых для чтения или записи данных, и, кроме того, увеличивает (на магнитной ленте) объем используемой памяти. Блокирование записей применяется и для дисковых файлов.

RECORD CELL (ячейка для записи)

- область фиксированной длины в относительном файле, которая используется для хранения одной записи.

RECORD FORMAT (формат записи)

- атрибут записи, определяющий способ занесения записи в файл. Формат записи задает метод определения длины

записи.

RECORD LENGTH (длина записи)

- размер записи в байтах.

RECORD LOCKING (захват записи)

- способность управлять операциями, выполняющимися над теми относительными и индексными файлами, к которым обращаются несколько программ и/или несколько потоков записей. Захват записи приводит к тому, что, когда некоторая программа добавляет, удаляет или модифицирует запись, то программе или потоку не разрешен доступ к этой записи или ячейке для записи. См. также AUTOMATIC RECORD LOCKING (автоматический захват записи) и MANUAL RECORD LOCKING (неавтоматический захват записи).

RECORD MANAGEMENT SERVICES (система управления данными СУД-32)

- набор процедур операционной системы, который вызывается программами с целью обработки файлов и записей внутри файлов. СУД-32 дает программам возможность выдать запросы GET (получить) и PUT (поместить) на уровне записей, а также читать и записывать блоки (ввод-вывод блоков). Процедуры СУД-32 выполняются в режиме управления.

RECORD-ORIENTED DEVICE (устройство, ориентированное на записи)

- устройство (типа терминала, построчно-печатающего

00152-01 31 01

устройства или устройства считывания перфокарт), на котором наибольшей единицей, к которой может обратиться некоторая программа за одну операцию ввода-вывода, является физическая запись.

RECORD'S FILE ADDRESS (RFA) (адрес записи в файле)

- уникальный адрес записи в файле, позволяющий осуществить впоследствии произвольный доступ к той записи, доступ к которой осуществлялся ранее.

REENTRANT CODE (реентерабельный код)

- код, который при выполнении никогда не модифицируется. Одну и ту же копию процедуры или программы, написанной в виде реентерабельного кода, могут разделять несколько пользователей.

REGISTER (регистр)

- ячейка памяти в логическом узле аппаратуры, отличном от основной памяти. См. также GENERAL REGISTER (общий регистр), PROCESSOR REGISTER (регистр процессора) и DEVICE REGISTER (регистр устройства).

REGISTER DEFERRED INDEXED MODE (косвенно-регистрационный режим с индексацией)

- индексный режим адресации, в котором спецификатор базового операнда использует регистр, отличный от регистра режима адресации.

REGISTER DEFERRED MODE (косвенно-регистрационный режим)

- режим адресации, в котором содержимое указанного регистра используется как адрес фактического операнда инструкции.

REGISTER MODE (регистрационный режим)

- режим адресации, в котором содержимое указанного регистра используется как фактический операнд инструкции.

RELATIVE FILE ORGANIZATION (относительная организация файла)

- размещение записей в файле, при котором все записи внутри бакета занимают ячейки одинаковой длины. Каждой ячейке присваивается последовательный номер, задающий ее положение относительно начала файла.

RELATIVE RECORD NUMBER (относительный номер записи)

- идентификационный номер, используемый для определения положения ячейки записи относительно начала файла; применяется как ключ в режиме произвольного доступа по ключу к относительным файлам.

REMOTE COMMAND TERMINAL (сетевой командный терминал)

- терминал, который через сеть логически присоединяется к другому узлу, так, что команда физически связывается с узлом посредством коммутируемой линии.

REMOTE DEVICE (удаленное устройство)

- устройство, которое непосредственно не связано с локальным узлом, но доступно через кластер.

REMOTE DTE (удаленное DTE)

- любое оконечное оборудование данных в сети, отличное от того, в котором располагается пользователь.

REMOTE NODE (удаленный узел)

- любой узел в сети, отличный от того, в котором рас-

00152-01 31 01

полагается пользователь. См. также ADJACENT NODE (смежный узел), LOCAL NODE (локальный узел), EXECUTOR NODE (управляющий узел).

REORGANIZATION (копирование файлов с сохранением атрибутов)

- копирование записей индексного файла в другой индексный файл с такими же атрибутами ключей, как во входном файле.

RESOURCE (ресурс)

- любое средство вк, выделяемое для реализации задания: физическая память, устройства ввода-вывода, центральный процессор, наборы данных. Контроль за использованием ресурсов выполняется с помощью квот и лимитов.

RESOURCE WAIT MODE (режим ожидания ресурса)

- состояние выполнения, при котором процесс, выдавая запрос на обслуживание требуемым ресурсом, указывает, что он будет ждать, пока системный ресурс не станет доступным. Если процессу необходимо уведомление о занятости ресурса, то на время выполнения программы он может запретить режим ожидания ресурса.

RESUME (продолжить, возобновить)

- активизировать приостановленный процесс. Противопоставляется понятию WAKE (пробудить).

RETURN STATUS CODE (возвращаемый код состояния)

- см. STATUS CODE (код состояния).

REVERSE VIDEO (реверсирование видеоизображения)

- свойство терминала реверсировать подразумеваемый по

умолчанию контраст видеоизображения. Если подразумеваемое по умолчанию изображение представлено черными буквами на белом фоне, то реверсированное изображение - это белые буквы на черном фоне.

RIGHTS DATABASE (база данных прав по идентификаторам)

- набор данных, поддерживаемый и используемый МОС ВП для определения идентификаторов и для связи идентификаторов с их владельцами.

RIGHTS LIST (список прав по идентификаторам)

- список, связанный с каждым процессом и включающий все идентификаторы, которыми владеет процесс.

ROUND ROBIN (циклическое планирование, циклическая диспетчеризация)

- вид разделения времени, при котором равноприоритетным образам предоставляется одинаковый доступ к CPU. Операционная система МОС ВП использует циклическое планирование для каждого из шестнадцати программных уровней приоритетов. При обработке очереди процессов заданного уровня приоритета используется дисциплина FIFO.

ROUTER (маршрутный узел)

- узел, который может посылать и получать пакеты, а также маршрутизировать пакеты от одного узла к другому.

ROUTING (маршрутизация)

- функция сети, определяющая маршрут, по которому данные передаются в пункт назначения.

00152-01 31 01

RUNAWAY TAPE CONDITION (состояние неконтролируемой работы магнитной ленты)

- безостановочное вращение магнитной ленты на накопителе. Обычно, состояние неконтролируемой работы магнитной ленты возникает из-за того, что на некоторую операцию не влияет состояние тайм-аута.

RUN-TIME PROCEDURE LIBRARY (исполнительная библиотека)

- набор программ, доступных образам основного режима во время их работы. Эти программы могут использоваться всеми образами основного режима, независимо от применяемого транслятора.

SCATTER/GATHER (разброс/сбор данных)

- способность передавать за одну операцию ввода-вывода данные из несмежных страниц памяти в смежные блоки на диске или данные из смежных блоков на диске в несмежные страницы памяти.

SCREEN WIDTH (ширина экрана)

- число символов, которые могут быть выведены в строке.

SCROLLING ("прокрутка" информации)

- свойство видеотерминала, которое за счет вертикального перемещения строк позволяет выводить текст, превышающий по объему полный экран. Например, при выдаче команды TYPE новые выводимые строки появляются внизу экрана и, в конечном счете, исчезают в его верхней части.

SEARCH LIST (список поиска)

- логическое имя, для которого эквивалентное имя имеет не единственное значение, а множество значений. Обычно, список поиска используется для нахождения некоторого файла, путем изучения размещения множества файлов.

SEARCH STRING (строка поиска)

- группа символов, определенная в некоторой команде, как объект операции поиска.

SECONDARY PASSWORD (вторичный пароль)

- пароль пользователя, который может быть запрошен во время входа в систему сразу после корректного ввода основного пароля.

SECONDARY STORAGE (внешняя память)

- память произвольного доступа большого объема.

SECONDARY VECTOR (вторичный вектор)

- ячейка, определяющая начальный адрес той программы обработки исключительного состояния, которая должна выполняться при следующих условиях: во-первых, данное состояние имеет место, и, во-вторых, либо первичный вектор содержит 0, либо заданная им программа обработки принимает решение не обрабатывать это состояние.

SECTION (секция)

- часть виртуальной памяти процесса, которая имеет общие атрибуты управления памятью (защита, доступ, фактор кластера и т. д.). Секция создается из секции

00152-01 31 01

образа, дискового файла или в результате работы программы системного обслуживания "создать виртуальное адресное пространство". См. GLOBAL SECTION (глобальная секция), PRIVATE SECTION (личная секция), IMAGE SECTION (секция образа) и PROGRAM SECTION (программная секция).

SECURE TERMINAL SERVER (программная защита терминала)

- средство МОС ВП, предназначенное для обеспечения входа пользователей в систему только с того терминала, с которого уже осуществлен выход из системы. Когда пользователь нажимает на терминальную клавишу BREAK, программная защита (если она задействована) вначале уничтожает любой процесс, созданный при входе с этого терминала, а затем инициализирует вход в систему. Если таких процессов нет, то вход в систему может выполняться сразу.

SECURITY ALARM (аварийный сигнал защиты)

- сообщение, посылаемое на терминалы тех операторов, которым предоставлены полномочия операторов защиты. Аварийный сигнал защиты срабатывает при возникновении события, заранее определенного как аварийное из-за его причастности к защите.

SECURITY OPERATOR TERMINAL (терминал оператора защиты)

- вид терминала, который предназначен для приема сообщений, посылаемых "оператору защиты" процессом OPCOM. Эти сообщения выдаются по аварийному сигналу защиты. Обычно такой терминал представляет собой

00152-01 31 01

терминал с твердой копией, расположенный в отдельной комнате, а выводимая на него информация регистрирует связанные с защитой события и сообщает детали для определения источника события.

SEQUENTIAL ACCESS MODE (последовательный метод доступа)

- способ поиска и хранения записи, при котором программа читает и заносит записи одну за другой в порядке их появления, причем начиная с любой произвольной точки в файле.

SEQUENTIAL FILE ORGANIZATION (последовательная организация файла)

- организация файла, при которой записи появляются в порядке их первоначального занесения. Записи могут быть как фиксированной, так и переменной длины. Последовательная организация файла допускает последовательный доступ к записям и произвольный доступ по адресу записи в файле. Последовательная организация файла с записями фиксированной длины допускает произвольный доступ по относительному номеру записи.

SERVED DEVICE (обслуживаемое устройство)

- устройство, локальный узел которого делает его доступным для других узлов кластера.

SHAREABLE IMAGE (разделяемый образ)

- образ, у которого разрешены все внутренние ссылки, однако, для получения выполняемого образа он должен быть скомпонован с одним или несколькими объектными модулями. Разделяемый образ не может быть выполнен.

00152-01 31 01

его файл может применяться для хранения библиотеки или программ. Разделяемый образ может использоваться системным администратором для создания глобальной секции.

SHELL PROCESS (процесс-прототип)

- заранее определенный процесс, который копируется инициатором заданий с целью создания минимального контекста, необходимого для объявления процесса.

SHRINK WORKING SET (сжатие рабочего набора)

- один из имеющихся у планировщика памяти способов получения страниц в физической памяти. С целью их получения планировщик памяти сжимает размеры рабочих наборов выбранных процессов. См. также SWAPPING (обмен рабочих наборов).

SIGNAL (сигнал)

- 1) электрический импульс, передающий информацию.
- 2) программный механизм, используемый для указания того, что обнаружено исключительное состояние.

SINK NODE (приемный узел)

- узел, на котором физически размещены регистрирующие типы приемников, например, файл или консоль.

SLAVE TERMINAL (подчиненный терминал)

- терминал, который обменивается информацией ввода-вывода с некоторым образом и непосредственно с операционной системой. С него невозможно выдать команды интерпретатору команд.

SMALL PROCESS (малый процесс)

- системный процесс, который в своем виртуальном адресном пространстве не содержит области управления и имеет сокращенный контекст. Примерами малых процессов являются процесс обмена рабочих наборов и пустой процесс. Малые процессы диспетчеризируются так же, как и пользовательские процессы, но до завершения своей работы они должны оставаться резидентными, т.е. не могут быть выгружены.

SOFTWARE CONTEXT (программный контекст)

- контекст, поддерживаемый операционной системой для описания процесса. См. SOFTWARE PROCESS CONTROL BLOCK (программный блок управления процессом).

SOFTWARE INTERRUPT (программное прерывание)

- прерывание, генерируемое на уровнях приоритетов прерываний от 1 до 15; оно может быть запрошено только программой.

SOFTWARE PRIORITY (приоритет программы)

- см. PROCESS PRIORITY (приоритет процесса) и QUEUE PRIORITY (приоритет в очереди)

SOFTWARE PROCESS CONTROL BLOCK (SOFTWARE PCB) (программный блок управления процессом) (программный PCB)

- структура данных, используемая для хранения программного контекста процесса. Операционная система определяет программный PCB для каждого процесса при его создании. Программный PCB включает следующие виды информации о процессе: текущее состояние; адрес

00152-01 31 01

во внешней памяти выгруженного из основной памяти процесса; уникальный идентификатор процесса; адрес заголовка процесса (который содержит аппаратный PCB). Программный PCB размещается в пространстве виртуальных адресов области системы. Он не участвует в обмене вместе с процессом.

SORTING (сортировка)

- упорядочивание записей в указанной последовательности.

SOURCE FILE (исходный файл)

- текстовый файл, содержащий информацию, предназначенную для трансляции в объектный модуль. Такие файлы не могут быть выполнены или скомпонованы.

SOURCE TASK (исходная задача)

- задача, которая инициирует запрос на соединение по логическому каналу в режиме связи задач.

SPANNED RECORD (сцепленная запись)

- запись, которая может пересекать границы блока. Сцепленная запись состоит из одного или нескольких сегментов данных. Положение некоторого сегмента внутри записи и его длина указывается словом управления сегментом - первыми пятью символами каждого сегмента.

SPOOL QUEUE (очередь спула)

- список файлов процессов, которые должны быть обработаны симбионтом. Например, очередь построочно-печатающего устройства - это список файлов, которые

необходимо распечатать на таком устройстве.

SPoolING (спул)

- метод использования высокоскоростного устройства большой емкости для буферизации данных, передаваемых между низкоскоростными устройствами ввода-вывода и высокоскоростной памятью. Спул вывода - это метод, в соответствии с которым информация, выводимая на низкоскоростные периферийные устройства (например, на построчно-печатающее устройство), помещается в очереди, содержащиеся на высокоскоростном устройстве (таком, как диск), где она ожидает пересылки на низкоскоростное устройство. Спул ввода - это метод, по которому информация, вводимая с низкоскоростного периферийного устройства (например, с устройства считывания перфокарт), помещается в очереди, содержащиеся на высокоскоростном устройстве (таком, как диск), где она ожидает пересылки к заданию, обрабатывающему такую информацию.

STACK (стек)

- отдельная область памяти, предназначенная для временного хранения информации, для процедур и для перехода с возвратом при обслуживании прерываний. Для обработки стека используется дисциплина LIFO. Когда элемент добавляется в стек, указатель SP уменьшается. Когда элемент извлекается из стека, SP увеличивается.

STACK FRAME (кадр стека)

- стандартная структура данных, построенная в стеке во время вызова процедуры. Она начинается с ячейки, адрес которой содержится в регистре FP, и продолжается по направлению к младшим адресам; удаляется при возврате из процедуры. Кадр стека имеет и другое название - CALL FRAME (кадр вызова).

STACK POINTER (SP) (указатель стека)

- общий регистр 14 (R14). SP содержит адрес "вершины" (наименьший адрес) определенного процессором стека. При обращении к SP выбирается один из пяти возможных указателей стека - ядра, управления, супервизора, пользователя или прерываний, в зависимости от значения битов текущего режима и стека прерываний в длинном слове состояния процессора.

START I/O ROUTINE (программа запуска ввода-вывода)

- программа, содержащаяся в драйвере устройства, которая отвечает за получение необходимых ресурсов (например, канала данных контроллера) и за активизацию устройства.

STATE (элемент сети, состояние)

- функции, которые в данный момент действуют на данном элементе сети. Элементы сети включают линию, цепь, локальный узел, модуль, оборудование DTE и регистрацию.

STATE QUEUE (очередь процессов по состоянию)

- список процессов, находящихся в конкретном состоянии

00152-01 31 01

обработки. Очереди процессов по состояниям используются планировщиком для сохранения следа выбора процессов на выполнение. Такие очереди включают процессы, ждущие общих флагов событий, приостановленные процессы и готовые к выполнению процессы.

STATUS (состояние)

- тип отображения для команд SHOW и LIST программы NCP. Состояние имеет отношение к динамической информации об элементах, которые содержатся в постоянной или непостоянной базах данных.

STATUS CODE (код состояния)

- значение длинного слова, которое указывает на успешное или ошибочное выполнение конкретной функции. Например, программы системного обслуживания при завершении своей работы возвращают код состояния в регистре R0.

STORE THROUGH (сквозное хранение)

- см. WRITE THROUGH (сквозная запись)

STREAM (поток)

- окно доступа к файлу, связанному с блоком управления доступом к записям. Оно поддерживает запросы операций над записями.

STREAM RECORD FORMAT (формат потока)

- свойство файла, определяющее, что данные в файле интерпретируются как непрерывная последовательность байтов без управляющей информации. Формат потока применяется только для последовательных файлов.

STRING (строка)

- связанная последовательность символов. Когда редактор текста ищет слово или фразу в текстовом файле, он ищет строку. Последовательность символов, которая образует команду, часто называется командной строкой.

STRING SEARCH BUFFER (буфер поиска строки)

- буфер для хранения искомой строки.

STRONG DEFINITION (сильное определение)

- определение глобального символа, явно доступного для ссылок из модулей, скомпонованных с модулем, в котором сделано данное определение. Компоновщик вносит глобальный символ с сильным определением в символьную часть карты памяти. Библиотекарь включает глобальный символ с сильным определением в таблицу глобальных символов библиотеки.

STRONG REFERENCE (сильная ссылка)

- ссылка на глобальный символ в объектном модуле, которая требует, чтобы компоновщик выдавал сообщение об ошибке, когда при компоновке для данного символа не будет найдено определение. Если библиотека содержит определение, то компоновщик включает библиотечный модуль, определяющий глобальный символ, в образ, который содержит сильную ссылку.

SUBDIRECTORY (подкаталог)

- файл каталога, размещенный в каталоге более высокого уровня и перечисляющий дополнительные файлы, которые

00152-01 31 01

принадлежат владельцу каталога.

SUBPROCESS (подпроцесс)

- вспомогательный процесс, созданный другим процессом. Процесс, который создал подпроцесс, является его владельцем. Подпроцесс и его подпроцессы разделяют пул квот и лимитов. Когда процесс-владелец удаляется из системы, все его подпроцессы (и их подпроцессы) также удаляются.

SUBROUTINE (подпрограмма)

- вспомогательная программа, выполняемая при вызове другой программой. Часто подпрограмма вызывается неоднократно до тех пор, пока не будет преодолено некоторое исключительное состояние.

SUBSTATE (подсостояние)

- промежуточное состояние цепи, которое выводится при выдаче команд SHOW и LIST программы NCP.

SUBSTITUTE BUFFER (буфер замены)

- буфер, используемый для хранения той строки, которая должна заменить искомую строку.

SUMMARY (конспект)

- подразумеваемый по умолчанию тип отображения для команд SHOW и LIST программы NCP. Конспект включает наиболее полезную информацию, отобранную из информации о состоянии и характеристиках некоторого элемента.

SUPERVISOR MODE (режим супервизора)

- третий наиболее привилегированный режим доступа к

процессору. В режиме супервизора работает интерпретатор команд операционной системы.

SUSPENSION (приостанов)

- состояние, в котором процесс не активен, но известен системе. Приостановленный процесс становится снова активным только в том случае, когда другой процесс просит операционную систему возобновить работу приостановленного процесса.

SWAP MODE (режим обмена)

- состояние выполнения процесса, которое определяет возможность выгрузки процесса из балансного набора. Если для процесса режим обмена запрещен, то рабочий набор процесса блокируется в балансном наборе.

SWAPPER (планировщик памяти)

- процесс, выполняющий общесистемное планирование памяти. Планировщик памяти переписывает модифицированные страницы во внешнюю память, создает прототип новых процессов, сжимает физический размер неактивных процессов, удаляет процессы из балансного набора и заносит в балансный набор процессы, ожидающие выполнения.

SWAPPING (обмен рабочих наборов)

- метод разделения ресурсов памяти между несколькими процессами за счет переписи всего рабочего набора во внешнюю память (выгрузка) и чтения другого рабочего набора в оперативную память (загрузка). Например, рабочий набор процесса может быть переписан во внеш-

00152-01 31 01

ную память на время ожидания этим процессом завершения ввода-вывода, выполняющегося на медленном устройстве. Он будет снова занесен в балансный набор после завершения ввода-вывода. Противопоставляется понятию PAGING (обмен страниц).

SWITCH (переключатель)

- см. QUALIFIER (квалификатор).

SWITCHED VIRTUAL CIRCUIT (SVC) (коммутируемая виртуальная цепь)

- временная логическая связь между двумя DTE, подсоединенными к сети PSDN, которая является аналогом соединения по коммутируемой линии. SVC устанавливается только на время передачи данных и разрывается после завершения передачи.

SYMBIONT (симбионт)

- полный процесс, передающий данные, ориентированные на запись, с устройства или на устройство большой емкости. Например, симбионт ввода передает данные с устройства считывания перфокарт на диск. Симбионт вывода передает данные с дисков на построчно-печатающие устройства.

SYMBIONT MANAGER (управление симбионтами)

- функция (в системном процессе, называемом диспетчером заданий), которая обслуживает очереди спула и динамически создает процессы-симбионты для выполнения необходимых операций ввода-вывода.

SYMBOL (символическое имя, символ)

- величина, которая будучи определена, изображает конкретную функцию или объект (например, командную строку, имя каталога или имя файла) в конкретном контексте. См. LOCAL SYMBOL (локальный символ), GLOBAL SYMBOL (глобальный символ) и UNIVERSAL SYMBOL (универсальный символ).

SYMBOLIC DEBUGGER (символический отладчик)

- программа, помогающая программисту при поиске ошибок в других программах.

SYNCHRONOUS DISCONNECT (синхронное раз'единение)

- раз'единение, которое выполняется при выдаче непрозрачной задачей вызова на завершение операций ввода-вывода в логическом канале без его открепления. Тем самым, эта задача может использовать данный канал для последующих операций ввода-вывода с той же или другой удаленной задачей.

SYNCHRONOUS RECORD OPERATION (синхронная операция над записью)

- метод обработки записи, в котором пользовательская программа выдает запрос на ввод и вывод записи и затем ждет удовлетворения этого запроса прежде чем продолжить свое выполнение.

SYNTAX (синтаксис)

- конкретный вид команды, включающий написание, порядок квалификаторов и параметров.

SYSTEM (система)

- в контексте "система/владелец/группа/все", система - это номера групп, не превышающие 10 (восьмеричное), которые используются операционной системой, операторами и системным администратором.

SYSTEM ADDRESS SPACE (адресное пространство системы)

- см. SYSTEM SPACE (пространство системы) и SYSTEM REGION (область системы).

SYSTEM BASE REGISTER (SBR) (базовый регистр системы)

- регистр процессора, содержащий физический адрес базы таблицы страниц системы.

SYSTEM BUFFERED I/O (системный буферизованный ввод-вывод)

- операция ввода-вывода, например, операция ввода-вывода с терминала или почтового ящика, в которой вместо буфера, определяемого процессом, используется промежуточный буфер из системного буферного пула.

SYSTEM CONTROL BLOCK (SCB) (блок управления системой)

- структура данных в пространстве системы, которая содержит все известные системе вектора прерываний и исключительных ситуаций.

SYSTEM CONTROL BLOCK BASE REGISTER (SCBB) (базовый регистр блока управления системой)

- регистр процессора, содержащий базовый адрес блока управления системой.

SYSTEM-DEFINED IDENTIFIER (системо-определенный идентифи-

катор)

- один из трех классов идентификаторов. Системо-определенный идентификатор задается системой с целью определения групп пользователей по их обращениям к системе. Например, все пользователи, обращающиеся к системе по козовому вызову, получают идентификатор DIALUP (козовый вызов).

SYSTEM DEVICE (системное устройство)

- устройство произвольного доступа большой емкости, на котором размещен том, содержащий операционную систему.

SYSTEM DYNAMIC MEMORY (системная динамическая память)

- память, резервируемая для операционной системы. Из нее, по мере необходимости, выделяются блоки для временного хранения информации. Если, например, образ выдает запрос ввода-вывода, то для хранения пакета этого запроса используется системная динамическая память. Каждый процесс имеет лимит на размер системной динамической памяти, которая может быть выделена ему сразу.

SYSTEM IDENTIFICATION REGISTER (регистр идентификации системы)

- регистр процессора, который содержит тип процессора и его серийный номер.

SYSTEM IMAGE (системный образ)

- образ, который считывается в оперативную память с диска при запуске системы.

00152-01 31 01

SYSTEM LENGHT REGISTER (SLR) (регистр длины системы)

- регистр процессора, содержащий размер таблицы страниц системы в длинных словах.

SYSTEM PAGE TABLE (SPT) (таблица страниц системы)

- структура данных, которая отображает виртуальные адреса области системы, включая адреса, используемые для обращения к таблицам страниц процессов. Таблица страниц системы содержит по одному элементу PTE для каждой страницы виртуальной памяти области системы. Физический базовый адрес таблицы SPT содержится в регистре, называемом SBR.

SYSTEM PASSWORD (системный пароль)

- пароль, запрашиваемый перед тем, как может быть инициализирован вход в систему.

SYSTEM QUEUE (системная очередь)

- очередь, используемая и обслуживаемая процедурами операционной системы. См. Также STATE QUEUE (очередь процессов по состоянию)

SYSTEM REGION (область системы)

- третья четверть виртуального адресного пространства или половина пространства системы с младшими адресами. Виртуальные адреса области системы разделяются между процессами. Примерами структур данных, отображаемых виртуальными адресами области системы, являются вектора входов в систему, блок SCB, таблица SPT и таблицы страниц процесса.

00152-01 31 01

SYSTEM SERVICES (системное обслуживание, программа системного обслуживания)

- процедуры, предоставляемые операционной системой; они могут быть вызваны пользовательскими образами.

SYSTEM SPACE (пространство системы)

- половина виртуального адресного пространства со старшими адресами. См. также **SYSTEM REGION** (область системы).

SYSTEM VIRTUAL ADDRESS (SVA) (системный виртуальный адрес)

- виртуальный адрес, определяющий ячейку в пространстве системы.

SYSTEM VIRTUAL SPACE (системное виртуальное пространство)

- см. **SYSTEM SPACE** (пространство системы).

TARGET TASK (приемная задача)

- задача, которая получает и обрабатывает запрос на соединение по логическому каналу в режиме связи задач.

TASK (задача)

- в сетях - это образ, выполняющийся в контексте процесса.

TASK SPECIFIER (спецификатор задачи)

- информация, выдаваемая системе "трал" и разрешающая ей завершить соединение по логическому каналу с отдаленной задачей. Эта информация содержит имя отдаленного узла, на котором выполняется приемная задача, и имя самой задачи.

TERMINAL (терминал)

- общее имя для тех периферийных устройств, которые имеют клавиатуру и видеоэкраны или печатающие устройства. Терминалы позволяют пользователям печатать команды и данные на клавиатуре и получать сообщения на видеоэкран или печатающее устройство.

TEXT BUFFER (текстовый буфер)

- область памяти в текстовом редакторе.

TIED ACCOUNT (ограничивающий учетный раздел)

- см. CAPTIVE ACCOUNT (ограничивающий учетный раздел).

TIMEOUT (тайм-аут)

- истечение лимита времени, за которое устройство должно завершить передачу информации ввода-вывода. Лимит тайм-аута определяется ожиданием драйвером запроса прерывания.

TIMER (таймер)

- два системных процесса, один из них ведет время дня и дату, а другой просматривает тайм-ауты устройств и выполняет зависящее от времени планирование запросов. Процесс таймера создается программой обслуживания прерываний таймера.

TIME-SHARING (разделение времени)

- метод выделения процессам времени CPU, при котором каждый процесс по очереди получает одинаковое количество времени. См. также REAL-TIME PROCESS (процесс реального времени).

TRACEBACK (обратное отслеживание)

- системное средство, которое проверяет и отображает состояние пользовательского стека вызова при ненормальном завершении образа.

TRACK (дорожка)

- набор блоков с единым радиусом на одной записывающей поверхности диска.

TRANSFER ADDRESS (адрес точки входа)

- адрес ячейки, содержащей точку входа в программу (первую выполняемую инструкцию).

TRANSLATION BUFFER (буфер трансляции)

- внутренняя кэш-память процессора, содержащая результат трансляции последних использованных виртуальных адресов.

TRANSPARENT (прозрачный)

- характеристика работы функций, которая не видна пользователю. Например, при вводе команды интерпретатор команд проводит синтаксический разбор командной строки и вызывает соответствующую системную программу. Пользователь видит не саму обработку, а лишь ее результат.

TRAP (ловушка)

- состояние исключительной ситуации, которое имеет место в конце инструкции, вызвавшей эту ситуацию. Счетчик инструкций, сохраняемый в стеке, является адресом следующей инструкции, которая должна быть выполнена обычным образом.

TRAP ENABLE (разрешение ловушки)

- три бита в слове состояния процессора, которые управляют действиями процессора при некоторых арифметических исключительных ситуациях.

TRIBUTARY (подчиненная станция)

- физическое окончание цепи в многоточечной системе, которое не является управляющей станцией.

TRIBUTARY ADDRESS (адрес подчиненной станции)

- числовой адрес, который используется управляющей станцией для опроса подчиненной станции.

TROJAN HORSE PROGRAM (программа "троянский конь")

- программа пользователя, целью которой является получение конфиденциальной информации или разрушение информации других пользователей и которая получает доступ к этой информации под предлогом выполнения полезных действий.

TURNKEY ACCOUNT (ограничивающий учетный раздел)

- см. CAPTIVE ACCOUNT (ограничивающий учетный раздел).

TWO'S COMPLEMENT (дополнительный код)

- дополнение до основания двоичной системы счисления. Получается путем прибавления 1 в наименьший значащий разряд обратного кода двоичного числа.

TWO-WAY ASSOCIATIVE CACHE (ассоциативная двусторонняя кэш-память)

- организация кэш-памяти, при которой имеются две группы непосредственно отображаемых блоков. В той и другой группе содержится несколько блоков для каждой

00152-01 31 01

позиции индекса в кэш-памяти. Блок данных из основной памяти может перейти в любую группу, соответствующую его позиции индекса. Двусторонняя ассоциативная кэш-память - это компромисс между двумя крайностями: полностью ассоциативной организацией и организацией с прямым отображением кэш-памяти; она имеет преимущества той и другой организации.

TYPE-AHEAD (предварительный ввод)

- метод управления терминалом, при котором пользователь может ввести команды и данные, когда программа обрабатывает ранее введенную команду. Предварительно введенные команды не отображаются на терминале до тех пор, пока командный процессор не будет готов к их обработке. Они хранятся в буфере предварительного ввода.

UNBLOCKED RECORDS (разблокированная запись)

- запись, которая содержится в одном блоке. В нем нет других записей или частей записей.

UNIBUS ADAPTER

- адаптер общей шины.

UNIT CONTROL BLOCK (UCB) (блок управления устройством)

- структура в базе данных ввода-вывода, которая описывает характеристики и текущие действия устройства. Блок управления устройством содержит также форк-блок для драйвера устройства; форк-блок является критической частью форк-процесса драйвера. Кроме того, UCB предоставляет для драйвера динамическую область

памяти.

UNIT INITIALIZATION ROUTINE (программа инициализации устройства)

- программа, которая готовит контроллеры и устройства к работе. Контроллеры и устройства требуют инициализации после сбоя питания и во время процедуры загрузки драйвера.

UNIVERSAL SYMBOL (универсальный символ)

- глобальный символ в разделяемом образе, который может использоваться модулями, скомпонованными с разделяемым образом. Универсальные символы обычно являются подмножеством всех глобальных символов в разделяемом образе. При создании разделяемого образа компоновщик гарантирует, что универсальные глобальные символы останутся доступными для ссылки после разрешения символов.

UNWIND THE CALL STACK (развернуть стек вызова)

- удалить кадры вызова из стека путем обратного отслеживания вложенных вызовов процедур. При этом используется текущее содержимое регистра FP и содержимое регистров FP, хранимых в стеке каждого кадра вызова.

UPLINE DUMP (дамп смежного узла)

- функция системы "трал", которая позволяет смежному узлу, управляемому МИОС (малой исполнительной ОС), создать дамп своей памяти в файле МОС ВП.

URGENT INTERRUPT (срочное прерывание)

- прерывание, получаемое на уровнях приоритетов преры-

00152-01 31 01

ваний от 24 до 31. Такие прерывания могут генерироваться процессором только для интервальных часов, серьезных ошибок и сбоя питания.

USER AUTHORIZATION FILE (файл авторизации пользователя)

- файл, содержащий запись о каждом пользователе, которого системный администратор авторизует для получения доступа к системе. Каждая запись определяет имя пользователя, пароль, подразумеваемый по умолчанию учетный раздел, код идентификации пользователя (UIC), квоты, лимиты и привилегии, приписанные отдельным лицам, использующим систему.

USER ENVIRONMENT TEST PACKAGE (UETP) (средства проверки функционирования системы)

- набор программ, которые проверяют полноту, правильность установки и готовность к использованию аппаратного и программного обеспечения ВК.

USER FILE DIRECTORY (UFD) (каталог файлов пользователя)

- см. DIRECTORY (каталог).

USER IDENTIFICATION CODE (UIC) (код идентификации пользователя)

- 32-х битное значение, приписанное пользователям и файлам, глобальным секциям, кластерам общих флагов событий и почтовым ящикам. Это значение определяет тип доступа (доступ на чтение и/или на запись, а для файлов - доступ на выполнение и/или удаление), предоставленного системе, группе, владельцу или всем.

00152-01 31 01

числовой UIC состоит из идентификатора группы и идентификатора члена группы, разделенных запятыми и заключенных в квадратные скобки. Эти идентификаторы могут быть представлены в виде буквенно-цифровых символов. Буквенно-цифровой UIC состоит из номера члена и, необязательно, номера группы.

USER MODE (режим пользователя)

- наименее привилегированный режим доступа к процессору. В режиме пользователя работают пользовательские процессы и программы исполнительной библиотеки.

USER NAME (имя пользователя)

- имя, которое пользователь печатает на терминале для регистрации в системе. См. также PASSWORD (пароль).

USER NUMBER (номер пользователя)

- см. MEMBER NUMBER (номер члена).

USER PASSWORD (пароль пользователя)

- связанный с пользователем пароль. При попытке пользователя войти в систему, пароль должен быть выдан корректно, так чтобы пользователь был опознан и допущен к системе. Имеется два типа паролей пользователя: основной и вторичный.

USER PRIVILEGES (привилегии пользователя)

- привилегии, предоставляемые пользователю системным администратором. См. PROCESS PRIVILEGES (привилегии процесса).

UTILITY (обслуживающая программа, утилита)

- программа, которая реализует набор связанных общеце-

00152-01 31 01

левых функций. Таковы, например, обслуживающие программы для разработки программ (редактор, компоновщик), обслуживающие программы управления файлами (копирования файлов, преобразования формата файлов) или обслуживающие программы управления операциями (программа дисковых квот, диагностики).

VALUE RETURN REGISTERS (регистры возврата значений)

- общие регистры R0 и R1, используемые по соглашению для возврата значений функций. Вызванными процедурами эти регистры не сохраняются. Они доступны любой вызванной процедуре в качестве временных регистров. Все остальные регистры (R2, R3, ..., R11, AP, FP, SP, PC) могут быть сохранены при вызовах процедур.

VARIABLE-LENGTH BIT FIELD (V9BF) (битовое поле переменной длины)

- набор от 0 до 32 смежных битов, размещаемых произвольно относительно границ байтов. Битовое поле переменной длины определяется четырьмя атрибутами: адресом "а" байта; позицией "Р" бита первой ячейки битового поля относительно бита 0 в байте с адресом А; размером в битах битового поля; индикатором, указывающим является поле знаковым или беззнаковым.

VARIABLE-LENGTH RECORD FORMAT (формат записи переменной длины)

- формат файла, в котором записи могут иметь различную длину.

VARIABLE WITH FIXED-LENGTH CONTROL RECORD FORMAT (запись переменной длины с постоянной управляющей частью)

- формат файла, в котором записи переменной длины содержат дополнительную управляющую область фиксированной длины. Управляющая область может использоваться для хранения номеров строк файла и/или для управления форматом печати.

VECTOR (вектор)

- 1) вектор прерывания или исключительной ситуации - это известная системе ячейка памяти, содержащая начальный адрес программы, которая должна выполняться при появлении данного прерывания или исключительной ситуации. Система определяет отдельные вектора для каждого контроллера прерывающего устройства и для классов исключительных ситуаций. Каждый системный вектор представляет собой длинное слово.
- 2) для обработки исключительных ситуаций пользователи могут объявлять два вектора программных исключительных ситуаций (первичный и вторичный) для любого из четырех режимов доступа. Каждый вектор содержит адрес программы обработки исключительной ситуации.
- 3) одномерный массив.

VERSION NUMBER (номер версии)

- 1) поле в спецификации файла, следующее за типом файла. Оно начинается с точки с запятой (;) или с точки (.), за которой следует число, обычно определяющее последний созданный файл среди всех файлов,

00152-01 31 01

имеющих ту же спецификацию.

- 2) число, которое используется для определения уровня изменения программы.

VIDEO TERMINAL (видеотерминал)

- терминал, с видеозэкраном для приема выводимой информации. См. TERMINAL (терминал).

VIRTUAL ADDRESS (виртуальный адрес)

- 32-битное целое число, определяющее место байта в виртуальном адресном пространстве. Аппаратура управления памятью преобразует виртуальный адрес в физический адрес.

VIRTUAL ADDRESS SPACE (виртуальное адресное пространство)

- набор всех виртуальных адресов, которые может использовать образ, выполняющийся в контексте данного процесса, для определения месторасположения инструкции или данных. Виртуальное адресное пространство, представляет собой линейный массив из 4294967296 (2**32) адресов байтов.

VIRTUAL BLOCK NUMBER (VBN) (номер виртуального блока)

- относящийся к файлу адрес блока на устройстве большой емкости. Первый блок в файле всегда является виртуальным блоком 1. Противопоставляется понятиям LOGICAL BLOCK NUMBER (номер логического блока) и PHYSICAL BLOCK NUMBER (номер физического блока).

VIRTUAL CIRCUIT (виртуальная цепь)

- связь между двумя DTE, подсоединенными к сети PSDN. Благодаря данной связи, DTE способны взаимодей-

00152-01 31 01

вать так, как если бы им на все время передачи информации была выделена конкретная цепь.

VIRTUAL DISK (виртуальный диск)

- выделенная часть пространства диска, предназначенная для хранения консольных файлов во время выполнения на консольном носителе операций сохранения или восстановления файлов. Виртуальный диск исключает необходимость в создании на консольном носителе диска/каталога.

VIRTUAL I/O FUNCTIONS (функции виртуального ввода-вывода)

- набор функций ввода-вывода, которые должны интерпретироваться вспомогательным управляющим процессом.

VIRTUAL MEMORY (виртуальная память)

- набор ячеек в физической памяти и на диске, к которым обращаются по виртуальным адресам. Размер виртуальной памяти в любой системе зависит от размера доступной физической памяти и размера дисковой памяти, используемой для нерезидентной виртуальной памяти.

VIRTUAL PAGE NUMBER (VPN) (номер виртуальной страницы)

- виртуальный адрес страницы виртуальной памяти.

VOLATILE DATABASE (непостоянная база данных)

- образ памяти, который содержит информацию об элементах управления сетью.

VOLUME (том)

- носитель большой емкости или бобина магнитной ленты. Том является самой большой логической единицей фай-

ловой структуры.

VOLUME SET (набор томов)

- набор данных файловой структуры на одном или нескольких носителях большой емкости.

WAIT (ждать)

- стать неактивным. Процесс входит в состояние ожидания, когда он приостанавливает себя, спит или объявляет, что ему надо ждать события, ресурса, взаимного исключения и т.д.

WAIT FOR INTERRUPT REQUEST (запрос на ожидание прерывания)

- запрос, выданный программой драйвера запуска ввода-вывода после того, как она активизирует устройство. Этот запрос приводит к приостанову форк-процесса драйвера до тех пор, пока устройство не запросит прерывание или не возьмет тайм-аут.

WAKE (пробудить)

- активизировать спящий процесс. Такой процесс может быть разбужен вызовом запланированного на определенное время будильника.

WEAK DEFINITION (слабое определение)

- определение глобального символа, явно не доступного для ссылки из модулей, скомпонованных с модулем, в котором делается это определение. Библиотекарь не включает глобальный символ со слабым определением в таблицу глобальных символов библиотеки. Слабые определения часто используются при создании библиотек для указания тех глобальных символов, которые необ-

00152-01 31 01

ходимы только в том случае, когда содержащий их модуль скомпонован с некоторой программой иным способом. Противопоставляется понятию STRONG DEFINITION (сильное определение).

WEAK REFERENCE (слабая ссылка)

- ссылка на глобальный символ, которая при отсутствии определения символа в модуле, явно заданном компоновщику, требует от последнего не выдавать сообщение об ошибке или же найти подразумеваемую по умолчанию таблицу глобальных символов библиотеки для своего разрешения. Слабые ссылки часто используются при создании объектных модулей для указания тех глобальных символов, которые не могут потребоваться во время выполнения.

WILDCARD CHARACTER (символ обобщения)

- графический символ типа звездочки или знака процента, который используется в спецификации файла внутри или вместо имени файла, типа файла, имени каталога или номера версии с целью указания понятия "все" для данного поля.

WINDOW (окно)

- 1) см. MAPPING WINDOW (окно отображения).
- 2) ряд пакетов, авторизованных для передачи по интерфейсу X.25 DTE/DCE. Младший номер последовательности в окне называется нижней гранью окна (0, когда виртуальная цепь только создана). Пакет, посланный номер последовательности первого пакета дан-

ных, не авторизованного для прохождения по интерфейсу, является значением верхней грани окна (т.е. Нижняя грань окна плюс размер окна).

WORD (слово)

- два смежных байта (16 битов), начинающихся на границе адресуемого байта. Биты нумеруются справа налево от 0 до 15. Слово определяется адресом байта, содержащего бит 0. При арифметической интерпретации слово является целым числом в дополнительном коде, с весом битов, увеличивающимся от бита 0 до бита 14. Если слово интерпретируется как число со знаком, то бит 15 является знаковым. Значение целого числа лежит в диапазоне от -32768 до 32767. При интерпретации слова, как беззнакового целого, вес битов увеличивается от бита 0 до бита 15, и значение беззнакового целого лежит в диапазоне от 0 до 65535.

WORKING SET (рабочий набор)

- набор страниц в пространстве процесса, к которым выполняющийся процесс может обращаться без страничного отказа. Рабочий набор выполняющегося процесса должен быть размещен в основной памяти. Оставшиеся страницы этого процесса, если они имеются, находятся или в основной памяти, но не в рабочем наборе, или во внешней памяти.

WORKING SET SWAPPER (процесс обмена рабочих наборов)

- системный процесс, который заносит рабочие наборы процессов в балансный набор и удаляет их из баланс-

ного набора.

WORLD (все)

- в контексте "система/владелец/группа/все", последнее относится ко всем пользователям, включая системных операторов, системного администратора и пользователей, принадлежащих как к группе владельца, так и к любой другой группе.

WRITE ACCESS TYPE (тип доступа на запись)

- при выполнении инструкции или процедуры специфицированный операнд этой инструкции или процедуры только записывается.

WRITE ALLOCATE (распределяемая запись)

- метод управления кэш-памятью, в котором кэш-память выделяется по отсутствию записи, а также по обычному отсутствию чтения.

WRITE BACK (обратная запись)

- способ управления кэш-памятью, при котором данные операции записи в кэш-память копируются в основную память только тогда, когда данные в кэш-памяти должны быть перезаписаны. Это приводит к временному несоответствию между кэш-памятью и основной памятью. Противопоставляется понятию WRITE THROUGH (сквозная запись).

WRITE THROUGH (сквозная запись)

- метод управления кэш-памятью, в котором данные из операции записи копируются в кэш-память и в основную память. Данные в кэш-памяти и в основной памяти

00152-01 31 01

всегда совпадают. Противопоставляется понятию WRITE BACK (обратная запись).

XDELTA (системный отладчик)

- средство для отладки операционных систем и драйверов.

X.3 (средство PAD X.3)

- рекомендация мкктт, которая определяет средство сборки/разборки пакетов в открытой сети.

X.25 (интерфейс X.25)

- рекомендация мкктт, которая определяет интерфейс между несколькими DTE и DCE для аппаратуры, работающей в пакетном режиме в открытых сетях.

X.28 (интерфейс X.28)

- рекомендация мкктт, которая определяет интерфейс DTE/DCE для стартстопного режима оборудования DTE, обращающегося к средству сборки/разборки пакетов (PAD) в открытых сетях данных некоторых стран.

X.29 (процедуры обмена X.29)

- рекомендация мкктт, которая определяет процедуры обмена управляющей информацией и пользовательскими данными между DTE, работающим в пакетном режиме, и средством сборки/разборки пакетов (PAD).

X.29 TERMINAL (терминал X.29)

- терминал, подсоединенный к средству сборки/разборки пакетов (PAD).

3. СОКРАЩЕНИЯ И УСЛОВНЫЕ ОБОЗНАЧЕНИЯ МОС ВП

ACE	- элемент списка управления доступом
ACL	- список управления доступом
ACP	- вспомогательный управляющий процесс
ADP	- блок управления адаптером
AME	- управляющая программа совместимости
AP	- указатель аргумента
AST	- асинхронное системное прерывание
ASTLVL	- уровень асинхронного системного прерывания
BCUG	- двусторонняя замкнутая группа пользователей
CCB	- блок управления каналом
CRB	- блок запросов канала
CF	- указатель текущего кадра
CPU	- центральный процессор
CRC	- контроль по циклической избыточности
CSR	- регистр команд и состояния
CSMA/CD	- сигнал передачи при параллельном доступе с обнаружением наложения в канале
CUG	- замкнутая группа пользователей
DCE	- аппаратура передачи данных
DCL	- диалоговый командный язык (язык DCL)
DBB	- блок данных устройства
DDT	- таблица точек входа драйвера
DLM	- разделение канала передачи данных
DMA	- прямой доступ к памяти

DPT	- прологовая таблица драйвера
DST	- таблица символов отладчика
DTE	- оконечное оборудование данных
ECB	- блок управления выходом
ECC	- код с исправлением ошибок
ESP	- указатель стека режима управления
ESR	- программа обслуживания исключительной ситуации
FAB	- блок доступа к файлу
FAL	- программа удаленного доступа к файлам
FCB	- блок управления файлом
FCS	- система управления файлом
FDL	- язык описания файла
FDT	- таблица выбора функций
FIB	- блок описания файла
FID	- идентификатор файла
FIFO	- первым пришел - первым обслужен
FP	- указатель кадра
GST	- таблица глобальных символов
ISECT	- секция образа
IDB	- блок данных прерывания
IOSB	- блок состояния ввода-вывода
IPL	- уровень приоритета прерываний
IRG	- межблочный промежуток
IRP	- пакет запросов ввода-вывода
IS	- стек прерываний
ISP	- указатель стека прерываний

00152-01 31 01

- ISR - программа обслуживания прерывания
- JIB - информационный блок задания
- KSP - указатель стека ядра
- LAP - протокол доступа к каналу
- LBN - номер логического блока
- LCN - номер логического канала
- LIFO - последним пришел - первым обслужен
- MBZ - должно равняться 0
- MCR - программа связи с оператором
- MFD - главный каталог файлов
- МККТТ - международный консультативный комитет по телефонии и телеграфии
- NAM - блок имени файла
- NCS - блок соединений сети
- NCP - программа управления сетью
- NSP - протокол обслуживания в сети
- ODS-1 - уровень 1 файловой структуры на диске
- ODS-2 - уровень 2 файловой структуры на диске
- OPCOM - процесс сообщений оператору
- PAD - средство сборки/разборки пакетов
- PC - счетчик инструкций
- PCB - блок управления процессом
- PCBV - базовый регистр блока управления процессом
- PFN - номер физической страницы
- PID - идентификатор процесса
- PSDN - сеть с коммутацией пакетов данных
- PSECT - программная секция

00152-01 31 01

PSL	- длинное слово состояния процессора
PSW	- слово состояния процессора
PTE	- элемент таблицы страниц
PVC	- постоянная виртуальная цепь
PO	- область программы
POBR	- базовый регистр области программы
POLR	- регистр длины области программы
POPT	- таблица страниц области программы
P1	- область управления
P1PT	- таблица страниц области управления
P1BR	- базовый регистр области управления
P1LR	- регистр длины области управления
QIO	- очередь ввода-вывода
RAB	- блок доступа к записям
RFA	- адрес записи в файле
RVN	- относительный номер тома
RWED	- чтение, запись, выполнение, удаление
SBI	- внутренняя синхронная шина
SCB	- блок управления системой
SBR	- базовый регистр системы
SCBB	- базовый регистр блока управления системой
SLR	- регистр длины системы
SP	- указатель стека
SPT	- таблица страниц системы
SRP	- сокращенный пакет запросов
SSP	- указатель стека режима супервизора
SVA	- системный виртуальный адрес

- SVC - коммутируемая виртуальная цепь
- UBA - адаптер общей шины
- UBI - внутреннее соединение общей шины
- UCB - блок управления устройством
- UETP - средства проверки функционирования системы
- UFD - каталог файлов пользователя
- UIC - код идентификации пользователя
- USP - указатель стека режима пользователя
- VBF - битовое поле переменной длины
- VBN - номер виртуального блока
- VCB - блок управления томом
- WCB - блок управления окном
- WCS - перезаписываемая управляющая память
- WDCS - перезаписываемая диагностическая управляющая память
- XAB - блок дополнительных атрибутов

Перечень ссылочных документов

1. Операционная система МОС ВП. Система управления данными СУД-32. Описание применения. 00152-01 31 02.
2. Операционная система МОС ВП. Подсистема системного сервиса. Программы системного обслуживания. Справочный материал. 00152-01 97 06.
3. Операционная система МОС ВП. Подсистема управления. Диалоговый командный язык. Руководство пользователя. 00152-01 92 19-1.
4. Операционная система МОС ВП. Подсистема режима совместимости с ОС РВ. Руководство пользователя. 00152-01 92 08.
5. Операционная система МОС ВП. Архитектура и система машинных инструкций. Справочный материал. 00152-01 97 01-1.
6. Операционная система МОС ВП. Подсистема редактирования и форматирования. Универсальный текстовый редактор. Руководство оператора. 00152-01 34 02.
7. Операционная система МОС ВП. Подсистема разработки программ. Программа компоновщик. Руководство программиста. 00152-01 33 04.
8. Операционная система МОС ВП. Подсистема исполнительной библиотеки. Руководство программиста 00152-01 33 09-1.
9. Операционная система МОС ВП. Подсистема разработки

Перечень ссылочных документов

1. Операционная система МОС ВП. Система управления данными СУД-32. Описание применения. 26.00152-01 31 02.
2. Операционная система МОС ВП. Подсистема системного сервиса. Программы системного обслуживания. Справочный материал. 26.00152-01 97 06.
3. Операционная система МОС ВП. Подсистема управления. Диалоговый командный язык. Руководство пользователя. 26.00152-01 92 19-1.
4. Операционная система МОС ВП. Подсистема режима совместимости с ос рв. Руководство пользователя. 26.00152-01 92 08.
5. Операционная система МОС ВП. Архитектура и система машинных инструкций. Справочный материал. 26.00152-01 97 01-1.
6. Операционная система МОС ВП. Подсистема редактирования и форматирования. Универсальный текстовый редактор. Руководство оператора. 26.00152-01 34 02.
7. Операционная система МОС ВП. Подсистема разработки программ. Программа компоновщик. Руководство программиста. 26.00152-01 33 04.
8. Операционная система МОС ВП. Подсистема исполнительной библиотеки. Руководство программиста. 26.00152-01 33 09-1.

00152-01 31 01

9. Операционная система МОС ВП. Подсистема разработки программ. Библиотекарь. Руководство пользователя. 00152-01 92 16.

10. Операционная система МОС ВП. Подсистема разработки программ. Символический отладчик. Руководство оператора. 00152-01 34 07-1.

11. Операционная система МОС ВП. Подсистема разработки программ. Программа корректировки образа задачи. Руководство программиста. 00152-01 33 05.

12. Операционная система МОС ВП. Подсистема системного программиста. Средства проверки функционирования системы эксперт. Руководство оператора. 00152-01 34 01.

13. Операционная система МОС ВП. Подсистема системного программиста. Программа регистрации ошибок. Руководство оператора. 00152-01 34 08.

14. Операционная система МОС ВП. Подсистема системного программиста. Программа анализа аварийного состояния системы. Руководство пользователя. 00152-01 92 15.

26.00152-01 31 01

9. Операционная система МОС ВП. Подсистема разработки программ. Библиотекарь. Руководство пользователя. 26.00152-01 92 16.

10. Операционная система МОС ВП. Подсистема разработки программ. Символический отладчик. Руководство оператора. 26.00152-01 34 07-1.

11. Операционная система МОС ВП. Подсистема разработки программ. Программа корректировки образа задачи. Руководство программиста. 26.00152-01 33 05.

12. Операционная система МОС ВП. Подсистема системного программиста. Средства проверки функционирования системы эксперт. Руководство оператора. 26.00152-01 34 01.

13. Операционная система МОС ВП. Подсистема системного программиста. Программа регистрации ошибок. Руководство оператора. 26.00152-01 34 08.

14. Операционная система МОС ВП. Подсистема системного программиста. Программа анализа аварийного состояния системы. Руководство пользователя. 26.00152-01 92 15.