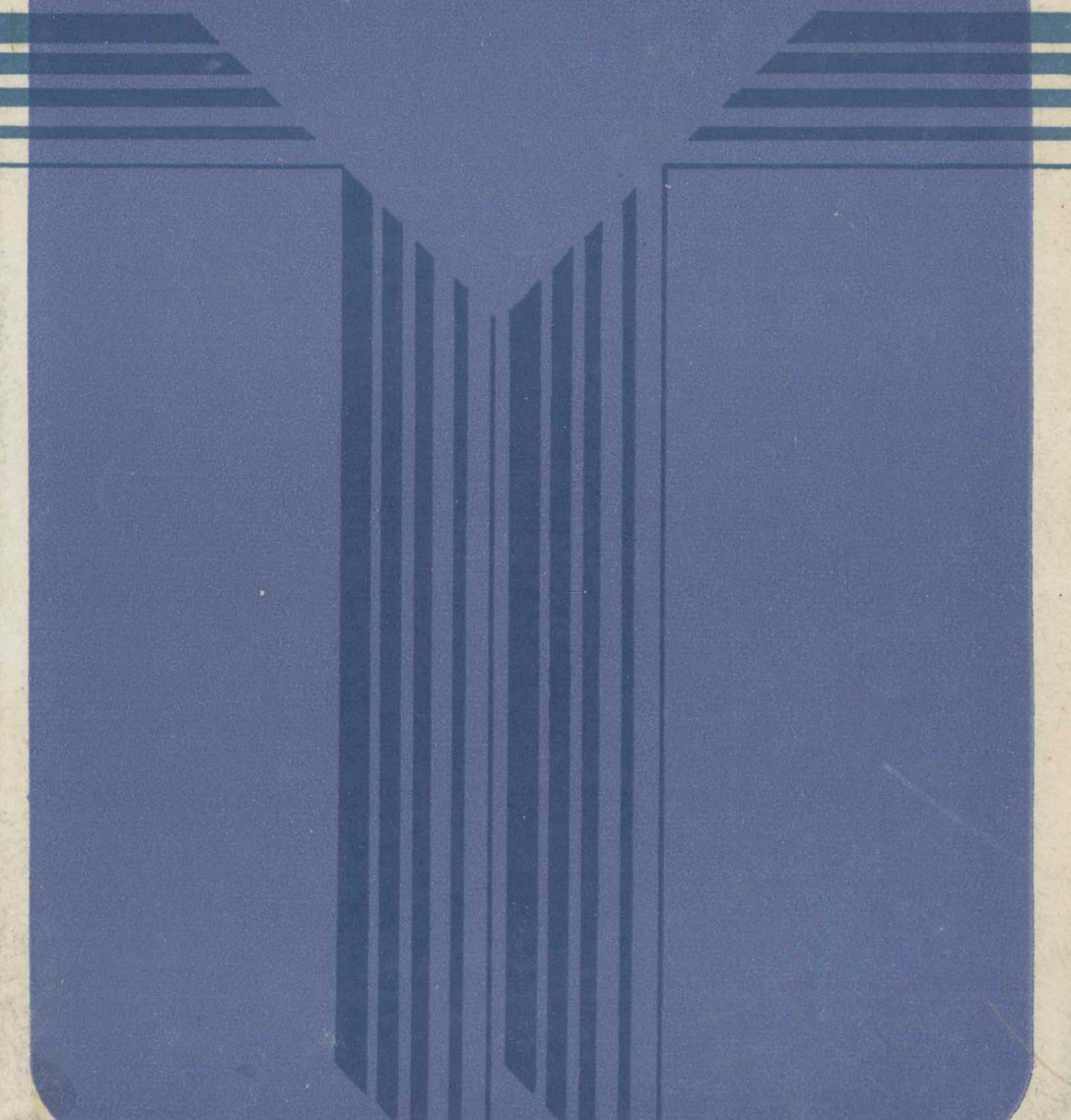


Г.Н. ЛИХАЧЕВА
В.Д. МЕДВЕДЕВ

ОПЕРАЦИОННЫЕ СИСТЕМЫ



*Галина Николаевна Лихачева,
Виктор Дмитриевич Медведев*

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Рецензенты *Л. Т. Чупрыгина, А. Н. Трошин*

Зав. редакцией *И. Г. Дмитриева*

Редактор *Э. С. Асланова*

Техн. редактор *Л. Г. Чельшева*

Корректоры *Г. В. Хлопцева, Г. М. Брызгунова*

Худ. редактор *Э. А. Смирнов*

Обложка художника *А. Малкина*

ИБ № 838

Сдано в набор 24.12.79 г. Подписано в печать 14.05.80 г.
А 10735. Формат 60×90¹/₁₆. Бум. тип. № 3.
Гарнитура «Литературная». Печать высокая. П. л. 14,5.
Усл. п. л. 14,5. Уч.-изд. л. 16,2. Тираж 21000 экз.
Заказ 77 Цена 65 коп.

Издательство «Статистика», Москва, ул. Кирова, 39

Великолукская городская типография управления
издательств, полиграфии и книжной торговли
Псковского облисполкома, г. Великие Луки,
ул. Полиграфистов, 78/12

Г. Н. ЛИХАЧЕВА, В. Д. МЕДВЕДЕВ

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Допущено Министерством высшего и среднего специального образования СССР в качестве учебного пособия для студентов вузов, обучающихся по специальности «Организация механизированной обработки экономической информации»



МОСКВА «СТАТИСТИКА»
1980

Л65 **Лихачева Г. Н., Медведев В. Д.**
Операционные системы. — М.: Статистика, 1980. — 231 с.
65 к.

Излагаются основные принципы построения и использования операционных систем для ЭВМ третьего поколения. Приводятся алгоритмы управления задачами, данными, заданиями при мультипрограммировании с фиксированным числом задач. Описываются схема действия и использование основных макрокоманд Супервизора для ОС ЕС и ОС ИБМ/360, Редактора связей и Загрузчика.

Для студентов вузов, специализирующихся по организации механизированной обработки экономической информации, а также для специалистов, занятых разработкой и применением средств программного обеспечения.

Л **30502—062**
008(01)—80 **98—80** **2405000000**

ББК 32.973.2
6Ф7.3

ВВЕДЕНИЕ

Использование современных вычислительных машин в настоящее время не представляется возможным без операционной системы (ОС). Операционная система — это организованный набор программ и данных, разработанный специально для управления ресурсами вычислительной системы, для облегчения создания программ для ЭВМ и для управления процессом их выполнения. Знать возможности операционных систем, их функциональные характеристики, тенденции развития должен не только разработчик систем, но и квалифицированный программист-пользователь.

Современные ОС насчитывают более миллиона команд и содержат средства управления данными, аппарат создания больших программ из отдельных отлаженных модулей, различные служебные программы, средства управления прохождением задания через вычислительную систему.

При изучении основ операционной системы упор делается на метод проектирования ОС, основные взаимосвязи между отдельными компонентами, функциональные алгоритмы этих компонент. Цели создания и внутренняя организация ОС рассматриваются в учебном пособии с современных позиций. Функциональные характеристики и алгоритмы излагаются применительно к ОС пакетного типа, реализующей мультипрограммирование с фиксированным числом задач. Такая ориентация обусловлена разработкой в нашей стране ОС Единой системы ЭВМ (ЕС ЭВМ).

Учебное пособие содержит четыре раздела.

В первом разделе даются основные понятия, используемые в операционных системах, определяются место операционной системы в составе программного обеспечения, роль прерываний при организации многопрограммной работы ЭВМ, типы многопрограммной работы и операционных систем.

Второй раздел посвящен вопросам проектирования и использования Супервизора и его составных частей: Супервизора задач, Супервизора перекрытий, Супервизора ввода-вывода. По каждой компоненте изложение ведется по одной схеме. Пояснения к блок-схемам даются в соответствии с номерами блоков на рисунке.

В главе 8 приведено описание и схема действия основных макрокоманд Супервизора ОС ЕС ЭВМ и ОС ИБМ/360.

В третьем разделе излагаются вопросы подготовки и прохождения заданий в операционной системе. В главе 10 дается описание

основных методов доступа, системы буферизации, основных обслуживающих программ, обеспечивающих работу с файлами данных. В главе 11 излагаются вопросы проектирования и использования Редактора связей и Загрузчика, реализованных в ДОС и ОС ЕС ЭВМ и ДОС и ОС ИБМ/360.

В четвертом разделе показаны перспективы развития операционных систем. В главе 12 излагаются вопросы совершенствования управления задачами и памятью. В главе 13 приводятся основные алгоритмы обслуживания ресурсов системы.

С целью закрепления материала после каждой главы приводятся вопросы.

В конце учебного пособия приводится список литературы по данному вопросу.

В основу учебного пособия легло обобщение материалов разработки отечественных операционных систем (ДОС ЕС ЭВМ, ОС ЕС ЭВМ) и зарубежных (ОС ИБМ/360, система Джей английской фирмы *ICL*, ОС Сименс 4004 и др.).

Все примеры, приведенные в учебном пособии, носят иллюстративный характер. Там, где материал относится к конкретной операционной системе, по тексту сделаны ссылки.

В учебном пособии использовалась терминология, установленная ГОСТами.

Учебное пособие составлено в соответствии с программой «Операционная система» для специальности 1738. Оно также будет полезно специалистам, занимающимся разработкой и применением программного обеспечения, аспирантам соответствующих специальностей.

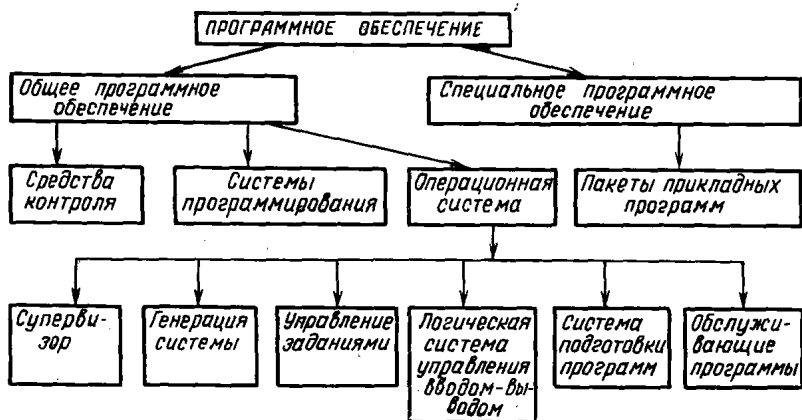
ВВЕДЕНИЕ В ОПЕРАЦИОННЫЕ СИСТЕМЫ

Глава I

НАЗНАЧЕНИЕ, СОСТАВ И ФУНКЦИОНИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1.1. Место операционной системы в составе программного обеспечения

Под *программным обеспечением* (ПО) понимается совокупность программ вычислительной машины, процедур и правил вместе со всей связанной с этими компонентами документацией, позволяющая использовать вычислительную машину для решения раз-



1.1. Структура программного обеспечения

личных задач. По задачам и функциям, выполняемым различными элементами ПО, можно выделить две большие группы: общее и специальное программное обеспечение. Классификационная схема ПО приведена на рис. 1.1.

К *общему программному обеспечению* относится совокупность программ, описаний и инструкций, предназначенных для автоматизации трудоемких технологических этапов разработки алгоритмов и программ, а также для организации и контроля вычислительного процесса на машине во время ее функционирования. Оно вклю-

чает системы программирования, операционную систему, осуществляющую общую организацию процесса обработки информации и связь ЭВМ с пользователем, средства контроля и диагностики неисправностей в ходе работы ЭВМ.

Специальное программное обеспечение представляет собой совокупность программ решения конкретных задач из различных сфер применения, реализованных в виде пакетов прикладных программ.

Операционная система предоставляет программисту совокупность приемов и процедур для работы на машине.

Под *операционной системой* понимается часть ПО, предназначенная для планирования и организации процесса обработки, ввода-вывода и управления данными, распределения ресурсов, подготовки и отладки программы и других вспомогательных операций обслуживания.

Основное назначение операционной системы:

- создание гибкой организации прохождения потока заданий через вычислительную систему;

- обеспечение простого и стандартного способа оформления заданий;

- обеспечение равномерной загрузки оборудования;

- организация хранения в системе больших файлов данных стандартными способами при наличии сравнительно большого числа методов доступа к этим данным;

- оптимальное использование всех ресурсов вычислительной системы.

Высокий уровень производительности вычислительной системы обеспечивается операционной системой тремя способами. Во-первых, в режиме пакетной обработки операционная система позволяет оператору выполнять подготовительные действия, которые совмещаются с исполнением ранее воспринятых заданий. Во-вторых, при мультипрограммном режиме операционная система распределяет и перераспределяет ресурсы машины между различными программами, сводя простой ресурсов к минимуму. В-третьих, производительность вычислительной системы определяется теми возможностями, которые она предоставляет пользователю. Операционная система имеет разнообразный набор программных средств для выполнения трудоемких операций ввода-вывода, отладки. В состав программного обеспечения включено также достаточное число компиляторов с входных языков для различных областей применения.

Операционная система обеспечивает целый спектр времени ответа, варьируя приоритетами выполняемых работ. Под временем ответа понимается время от поступления работы на выполнение до ее завершения, т. е. получения результатов. Время ответа часто называется еще и временем прохождения. Совмещая режимы пакетной обработки и разделения времени, операционная система выдает результаты более коротких работ во время выполнения более длительных работ.

Увеличение производительности труда программиста осуществляется путем предоставления ему разнообразных средств программирования: большого числа языков программирования, библиотек, содержащих наиболее употребительные программы, удобных средств ввода-вывода, отладки, а также средств удобного оформления заданий.

Адаптируемость программ к изменяющимся ресурсам вычислительной системы обеспечивается тем, что операционная система имеет средства обслуживать большой диапазон машинных конфигураций. Кроме того, операционная система дает возможность строить программы, независимые от устройств ввода-вывода, что упрощает проблему модификации программ, связанных с заменой этих устройств.

Возможность расширения означает, что операционную систему можно использовать на моделях машин с новыми внешними устройствами, для новых областей применения. При появлении новых требований в операционную систему могут быть добавлены новые средства, реализующие дополнительные возможности, не влияющие на задел программ, добавлены компиляторы с новых языков.

Разработка операционной системы является трудоемкой и дорогостоящей. Так, если над созданием операционной системы для машин второго поколения работало около 200 человек, то над созданием операционной системы для машин третьего поколения работало примерно 1000 человек и еще столько же работало над созданием компиляторов. Стоимость разработки программного обеспечения фирмы ИБМ достигла 1 млрд. долларов. При этом на проектирование (вплоть до составления укрупненных блок-схем) относится 20% затрат, а на программирование, отладку, комплексную увязку и документирование — 80% затрат. Подсчитано, что за 20 лет после появления первой ЭВМ стоимость разработки всех программ достигает 36 млрд. долларов. Общественную ценность представляют из них только те программы, общая стоимость которых составляет 2—3 млрд. долларов. Ожидается, что объем программного обеспечения за ближайшие 15 лет увеличится в 10 раз.

Таким образом, важное значение приобретают вопросы изучения операционных систем, основной целью которых является увеличение производительности вычислительной системы и персонала, сбалансированное с капитальными вложениями и отдачей.

1.2. Основные понятия ОС

Операционная система представляет собой библиотеку программ, часть которой постоянно находится в основной памяти машины, другая — загружается в нее по мере необходимости. Часть операционной системы, постоянно находящаяся в основной памяти, называется *резидентной частью ОС*, или *ядром системы*; программы, вызываемые в основную память для выполнения определенных функций и не хранящиеся там постоянно, — *транзитными программами* или просто *транзитами*.

Транзиты вызываются в один и тот же участок основной памяти — транзитную область управляющей программы. Вся область основной памяти, занимаемая ядром и транзитами, называется областью управляющей программы. Остальная часть основной памяти образует область проблемных программ.

Разделение ОС на ядро и транзиты позволяет минимизировать область управляющей программы и соответственно увеличить область проблемных программ.

Операционная система спроектирована для работы в различных режимах, поэтому на вход системы может одновременно поступать большое количество работ, выполнить которые сразу невозможно просто из-за того, что нет достаточного количества ресурсов. К ресурсам системы относятся время центрального процессора, место в основной и внешней памяти, ключи защиты, отдельные программы, устройства ввода-вывода, каналы, таймер, пульт управления. Чтобы выполнить работу на машине, программист пишет программу, представляющую последовательность команд для решения какой-либо задачи. Чтобы указать операционной системе, какую работу нужно выполнить, программа описывается с помощью операторов языка управления заданиями.

Вся работа, поступающая на вход системы, делится на независимые задания, являющиеся внешней единицей работы для операционной системы. Задания независимы, так как ни у операционной системы, ни у программиста нет возможности указать, как одно задание зависит от другого. Задания могут группироваться для ввода в систему во входном потоке заданий. *Поток заданий*, или *пакет заданий*, — совокупность нескольких заданий. *Задание* — совокупность декларируемых управляющими операторами текстов программ и входных данных для отдельных программ. Основное назначение управляющих операторов — определить задание или его часть.

Составная часть задания, выполняемая самостоятельно, называется *шагом задания*, или *пунктом*. Шаги одного задания могут быть взаимозависимы.

Операционная система имеет модульную структуру, которая позволяет пользователю приспособить систему к конкретным конфигурациям технических средств. Отдельные программы операционной системы, реализующие конкретные функции ОС, могут включаться в систему по желанию пользователя. Процесс создания конкретной операционной системы, учитывающий особенности вычислительной машины и задач пользователя, называется *генерацией системы*. Средства генерации системы представляют собой совокупность программ и правил, необходимых для выполнения генерации.

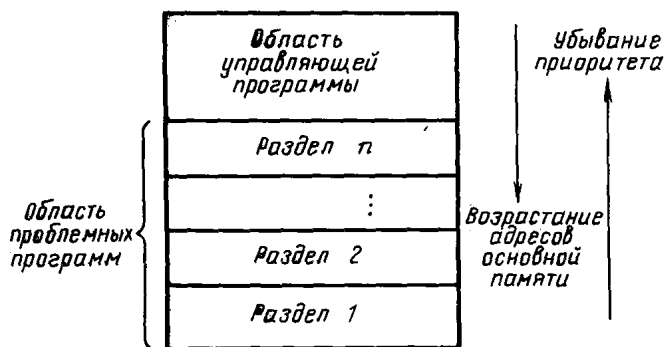
Эффективное использование основной памяти важно как с точки зрения построения программы, так и с точки зрения работы вычислительной системы. Рассмотрим два способа управления памятью, которые нашли самое широкое распространение в силу своей простоты и дешевизны разработки ОС: *одиночное непрерыв-*

ное распределение памяти и распределение разделами. При первом способе вся память, не занятая управляющей программой, выделяется одному заданию. Процессор также полностью отдается в распоряжение этого задания. Хотя это и расточительный метод использования памяти и времени центрального процессора, но он чрезвычайно прост и дешев в реализации разработки ОС.

При втором способе основная память, предоставляемая проблемным программам, при генерации ОС делится на фиксированное число разделов. Число таких разделов зависит от конфигурации операционной системы и колеблется от трех для менее мощных до пятнадцати для наиболее мощных систем. В каждом разделе одновременно может выполняться одно задание. Программы, выполняемые в разных разделах, не зависят друг от друга.

Для предотвращения возможного разрушения программы, выполняемой в одном разделе, в результате неверной работы программы другого раздела каждому из разделов присваивается собственный ключ защиты памяти. Область управляющей программы имеет ключ защиты 0, ключи разделов — от 1 до 15.

Между разделами установлен жесткий *приоритет*. Самый низкий приоритет имеет раздел, расположенный сразу за областью управляющей программы, самый высокий — наиболее удаленный. Это проиллюстрировано на рис. 1.2.



1.2. Распределение основной памяти между разделами

Каждому разделу отводится не только участок основной памяти, но и устройства ввода-вывода. Для каждого из разделов существует свой входной поток заданий.

С точки зрения пользователя при многопрограммной работе каждый из разделов вместе с набором внешних устройств, закрепленных за ним, может рассматриваться как отдельная машина. Будем называть такую «самостоятельную» машину *поток*ом.

Возможности потоков (объем основной памяти, количество разделов и закрепление внешних устройств) задаются во время генерации ОС, поэтому пользователь может проанализировать характер выполняемых им на машине работ, использовать более эффек-

тивно ресурсы системы. Например, пусть конфигурация машины имеет 256 Кбайт основной памяти, одно устройство чтения с перфокарт, два печатающих устройства, три пульта и другие стандартные устройства. Предполагаемую работу, которую нужно выполнять на машине, можно подразделить на 3 класса: задачи класса *A*, которые требуют небольшого объема памяти, устройство чтения с перфокарт и устройство печати; задачи класса *B*, которые требуют большого объема памяти и не требуют медленных устройств; задачи класса *C*, которые требуют небольшого объема памяти и печатающего устройства. В этом случае наиболее эффективным использованием машины будет разбиение ее на три потока *A*, *B*, *C*. Пример деления памяти на разделы показан на рис. 1.3.

Область управляющей программы	Поток <i>A</i> 30 КБ	Поток <i>B</i> 100 КБ	Поток <i>C</i> 20 КБ
-------------------------------------	-------------------------	--------------------------	-------------------------

1.3. Пример распределения основной памяти между потоками

Потоку *A* будет выделено устройство чтения с перфокарт и печатающее устройство, потоку *C* — печатающее устройство. В этом случае программы могут работать в индивидуальном потоке и не требовать перераспределения устройств. Кроме того, можно лучше управлять вводом заданий, используя поток с минимальной памятью, например *C*, а не поток *B*, требуя перераспределения устройств, если задания вводятся с карт.

С понятием задания связано понятие *ранг*, которое играет ту же роль для заданий, что и приоритет. Ранг позволяет управлять порядком загрузки заданий для выполнения. Например, в системе Джей фирмы *ICL* приняты следующие значения ранга задания:

0 — задания с этим рангом не будут загружаться, а будут ждать, пока не изменится ранг;

1 — задания будут загружаться последовательно. Если хотя бы одно задание не может быть загружено, то ни одно задание с рангом 1 не будет загружено в этот поток;

2—8 — задания с рангами 2—8 загружаются по мере появления и наличия для них требуемых ресурсов;

9 — этот ранг присваивается заданию, которое не может быть загружено, так как не все требуемые ему устройства свободны или недостаточен объем свободной памяти. Для задания с таким рангом все свободные устройства и память резервируются до тех пор, пока не будут выполнены требования на устройства и память;

10 — этот ранг используется, чтобы отменить резервирование устройств и памяти для заданий с рангом 9.

Как только операционная система распознала шаг задания и определила наличие требуемых ему ресурсов (устройства ввода-вывода, ключ защиты, объем основной памяти), формируется за-

дача — внутренняя единица действия в операционной системе. Следует четко различать понятия: задание, задача, программа.

До создания операционных систем различие между программой и задачей было излишнее, так как в каждый момент времени выполнялась одна программа. При мультипрограммном режиме одна и та же программа ОС может вызываться несколькими различными проблемными программами одновременно. Возникает конфликтная ситуация, когда за один ресурс борются две и более программ. Для разрешения конфликта следует ввести средство, хранящее информацию о состоянии программы, ее запросах на ресурсы, приоритете обслуживания и др. Поэтому было введено понятие задачи. *Задача* представляет собой совокупность управляющей информации о программе, самой программы и данных, которые обрабатываются этой программой. В задаче отражена информация об обеспечении программы всеми ресурсами, необходимыми для ее выполнения.

Поэтому мультипрограммирование означает управление задачами, а не программами. Именно задачи конкурируют между собой за получение времени процессора и других ресурсов системы.

Чтобы выполнить программу, нужно ввести задание на ее выполнение. Например, нужно выполнить в машине программу *ABC*, написанную на Алголе. Следовательно, сначала должен работать транслятор с языка Алгол. Оттранслированная программа представлена в объектном коде и еще не готова к выполнению. Работу по подготовке программ к выполнению осуществляет программа операционной системы, называемая Редактор связей. Оттранслированную и отредактированную программу можно загружать в основную память и выполнять.

Таким образом, задание на выполнение программы *ABC* будет состоять из трех шагов: транслировать программу *ABC*, отредактировать, выполнить. Как только операционная система распознает первый шаг задания — транслировать программу *ABC* — и выделит требуемые ресурсы, будет сформирована задача, действие которой заключается в выполнении транслятора с языка Алгол. Программа, которая выполняется для данной задачи, называется *подчиненной данной задаче*. В нашем примере транслятор с Алгола подчинен задаче, образованной из первого шага задания.

Задаче, образованной из второго шага задания — отредактировать, будет подчинена программа Редактор связей. Задаче, образованной из третьего шага задания, подчиняется оттранслированная и отредактированная программа *ABC*.

1.3. Управляющая программа

Все программы, выполняемые машиной, можно подразделить на две большие группы: управляющая программа и обрабатывающие программы.

Основу операционной системы составляет *управляющая* программа. Ее назначение — автоматизировать управление работой

вычислительной машины. Управляющая программа выполняет функции подготовки операционной системы к функционированию, приему заданий и подготовки их к выполнению, управлению ходом выполнения задач и процедурами ввода-вывода.

К *обрабатывающим* относятся программы, облегчающие процесс составления, отладки, хранения программ. Группу обрабатывающих программ составляют трансляторы с различных языков программирования, Редактор связей, средства генерации системы, обслуживающие программы и проблемные программы.

По выполняемым функциям управляющая программа делится на три части: управление заданиями, управление задачами, управление данными.

Управление заданиями — часть управляющей программы, которая организует прием заданий из входного пакета заданий, осуществляет их контроль, подготовку к выполнению и запуск.

Управление заданиями выполняет следующие функции: анализ входного пакета заданий, запись входного пакета на устройства с произвольным доступом, подготовка задания к выполнению, распределение устройств ввода-вывода, управление установкой носителей, подготовка к запуску программ с контрольной точки, общее планирование порядка выполнения работ.

Анализ входного пакета заданий заключается в определении начала текущего задания, контроле правильности управляющих операторов и порядка их следования, установлении конца описания одного задания и начала следующего или конца заданий.

Запись входного пакета на устройства с произвольным доступом осуществляется для ускорения прохождения заданий.

Подготовка задания к выполнению заключается в инициировании шага задания, под которым будем понимать его распознавание, определение наличия требуемых ресурсов и формирование задачи. Управление заданиями распределяет устройства ввода-вывода, основную память, ключи защиты.

Распределение устройств ввода-вывода заключается в закреплении их за отдельными заданиями и производится с целью более эффективного их использования, а также для обеспечения независимости программ от конкретных физических адресов устройств с помощью метода логических устройств.

Управление установкой носителей заключается в том, что оператору заранее выдается информация о том, какие устройства ввода-вывода следует включить, какие надо установить тома для бесперебойной работы системы.

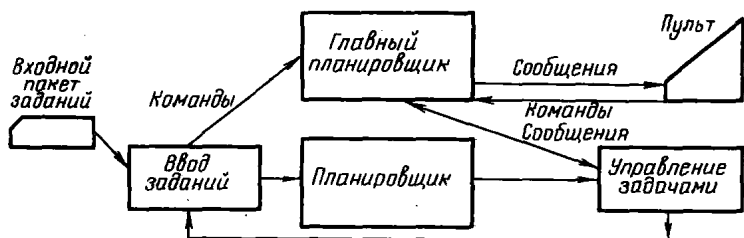
Подготовка к запуску программ с контрольной точки означает те же действия, что и подготовка задания к выполнению. Операционная система позволяет программисту указать в программе контрольные точки, начиная с которых при необходимости можно возобновить выполнение программы.

Для выполнения своих функций управление заданиями использует информацию, содержащуюся в управляющих операторах языка управления заданиями из входного потока, в директивах,

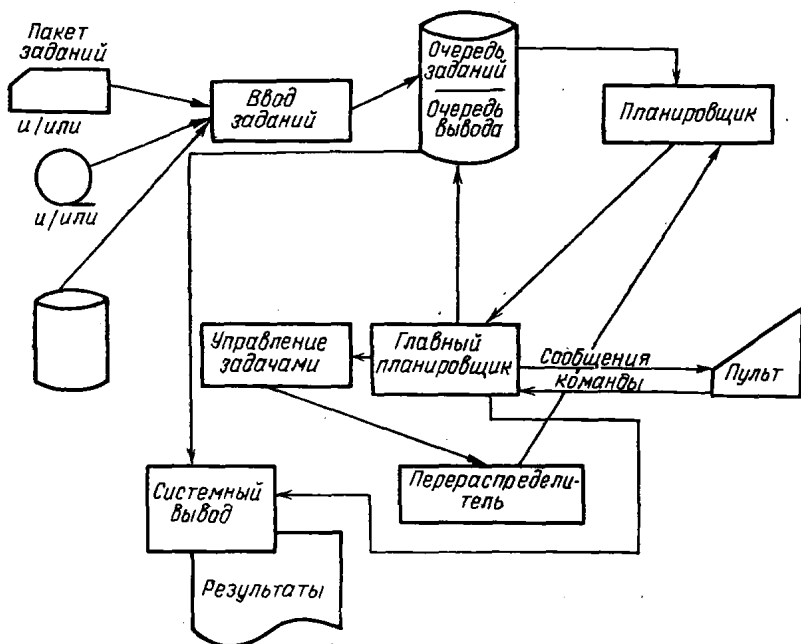
поступающих от оператора, а также пользуется информацией, сформированной другими частями управляющей программы.

Управление заданиями представляет собой совокупность программ. В зависимости от организации планирования прохождения заданий различаются: система последовательного планирования и система приоритетного планирования.

Общая схема системы последовательного планирования заданий изображена на рис. 1.4. Она состоит из программ Системный



1.4. Последовательное планирование



1.5. Приоритетное планирование

ввод, Планировщик, Главный планировщик. Последовательное планирование заключается в том, что задания из входного пакета вводятся последовательно программой Системный ввод. Для каждого задания программой Планировщик просматриваются и анализируются управляющие операторы, выделяются устройства вво-

да-вывода, а оператору выдается сообщение, какие тома должны быть установлены. После того как задание получило требуемые ему ресурсы, Главный планировщик формирует задачу и управление передается управлению задачами.

Система приоритетного планирования заданий является более мощной (рис. 1.5). Задания не выполняются, как только они обнаружены во входном потоке. Вместо этого вся управляющая информация, относящаяся к одному заданию, помещается на устройство с произвольным доступом и образует Очередь заданий (Очередь ввода), из которой она в дальнейшем выбирается в зависимости от ранга. Очередь может пополняться из нескольких входных пакетов заданий.

Аналогично, вместо непосредственной печати или перфорации результатов, выходные данные могут быть с большой скоростью размещены на устройстве с произвольным доступом для более позднего вывода по мере освобождения требуемых устройств. Для этой цели управляющая информация о выходных данных каждого задания помещается на устройство с произвольным доступом, образуя Очередь выходных работ (Очередь вывода).

Система приоритетного планирования заданий реализуется программами Системный ввод, Планировщик, Главный планировщик, Перераспределитель и Системный вывод. Системный ввод, Планировщик, Главный планировщик выполняют значительно больше функций, чем одноименные программы, реализующие последовательное планирование заданий.

Система приоритетного планирования является более гибкой, позволяет более эффективно использовать ресурсы, обеспечивая порядок выполнения заданий в зависимости от их приоритета и наличия ресурсов. Система может реагировать на задержки, вызываемые установкой или отключением отдельных устройств ввода-вывода, не прерывая выполнения других заданий.

Система приоритетного планирования заданий обеспечивает также ведение Системного журнала. Для каждого задания в Системном журнале указывается его имя, время, необходимое для выполнения каждого шага задания, и другая информация, которая может использоваться для анализа.

Управление задачами — часть управляющей программы, которая контролирует выполнение задачи на всех этапах, начиная с момента формирования задачи и до получения результатов работы.

Управление задачами осуществляют программы Супервизор задач, Супервизор времени, Супервизор перекрытий, Супервизор основной памяти, Супервизор прерываний, которые динамически распределяют ресурсы системы между несколькими задачами в процессе их выполнения. Управление задачами распределяет время центрального процессора, участки основной памяти, таймер, программы.

Управление задачами выполняет следующие основные функции: обработку прерываний, формирование задач, управление основной памятью, связь с оператором, загрузку программ в основ-

ную память для выполнения, выполнение процедур, связанных с окончанием задания, службу времени, создание контрольных точек и возможность возобновления обработки с контрольной точки, защиту памяти, обеспечение одновременного выполнения нескольких задач.

Супервизор получает управление по прерываниям. Обработка прерываний заключается в том, что Супервизор распознает тип прерывания и передает управление соответствующей программе его обработки. Программист полностью освобожден от необходимости работать с прерываниями.

Существует несколько способов формирования задачи. Первый способ — инициирование задачи из шага задания — реализуется управлением заданиями. Второй способ — формирование задачи с помощью макрокоманды создания новой задачи во время выполнения программы. Созданная задача выполняется одновременно с породившей ее задачей и с задачами других заданий, находящихся в данный момент в памяти. Формирование задачи производит Супервизор. С третьим способом формирования задачи мы познакомимся в главе 4.

Средства управления основной памятью позволяют эффективно ее использовать задачами, созданными из шагов заданий, и составлять программы, не зависящие от расположения в основной памяти их самих и обрабатываемых ими данных.

Управление задачами предоставляет оператору возможность управлять работой системы, обеспечивая его двусторонней связью. С одной стороны, система выдает оператору сообщения, позволяющие следить за ходом вычислительного процесса, и требует отвечать на запросы в тех ситуациях, когда работа не может быть продолжена без вмешательства оператора. С другой стороны, система принимает ответы оператора на поставленный запрос, а также директивы оператора системе.

Загрузка программ в основную память означает определение местоположения программы в одной из системных библиотек, размещаемых на магнитном диске, и чтение ее в память.

Программа может закончить свое выполнение нормально или аварийно. В любом случае управление передается Супервизору, который «освобождает» используемые программой ресурсы: устройства ввода-вывода, память, ключ защиты. В случае аварийного окончания может быть произведена распечатка основной памяти для диагностики и исправления ошибок.

Служба времени обеспечивает слежение за временем суток, которое проблемная программа может запросить в любой момент своего выполнения, печать времени начала и окончания выполнения задания, управление интервалом времени, запрошенного программой.

Контрольные точки создаются программистом, чтобы иметь возможность временно прекратить выполнение программы и возобновить ее выполнение не с начала, а с данной контрольной точки. Супервизор запоминает информацию, необходимую для возобнов-