

*Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования*

*АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ*

*Факультет математики и информатики*

# **ОПЕРАЦИОННЫЕ СИСТЕМЫ**

Учебно-методическое пособие

Благовещенск

2017

ББК 32.973-018.2  
Г 15

*Печатается по решению  
редакционно-издательского совета  
факультета математики и информатики  
Амурского государственного  
университета*

***Составитель Т.А. Галаган***

Операционные системы. Учебно-методическое пособие для бакалавров направлений подготовки 09.03.01 – Информатика и вычислительная техника, 09.03.02 – Информационные системы и технологии, 01.03.02 – Прикладная математика и информатика, 38.03.05 – Бизнес-информатика. – Благовещенск: Амурский гос. ун-т, 2017, 24 с.

Пособие предназначено для студентов второго курса, изучающих дисциплину «Операционные системы». Оно содержит основные определения, а также схемы и таблицы, позволяющие лучше структурировать информацию лекционного курса и помочь студентам в самостоятельной работе и при подготовке к промежуточной аттестации.

*Рецензент: Алутина Е.Ф., начальник управления учебно-методической работы ФГБОУ ВО «благовещенский педагогический государственный университет», канд. физ.- мат. наук*

© Амурский государственный университет, 2017  
© Галаган Т. А., составитель, 2017

## ОГЛАВЛЕНИЕ

Введение	4
Основные понятия	5
Архитектура операционной системы	9
Процессы и потоки	11
Управление оперативной памятью	14
Подсистема управления вводом-выводом	19
Файловая система NTFS	23
Литература	24

## ВВЕДЕНИЕ

Операционная система предназначена для решения двух основных задач:

предоставление пользователю (программисту) вместо реальной аппаратуры компьютера некоторого образа расширенной виртуальной машины, с которой удобней работать и которую легче программировать;

повышение эффективности использования компьютера за счет рационального управления его ресурсами в соответствии с некоторыми известными критериями.

Функции операционной системы автономного компьютера обычно группируются либо в соответствии с типами локальных ресурсов, которыми она управляет, либо в соответствии со специфическими задачами, которые она выполняет для всех ресурсов. Такие группы функций называют подсистемами. Наиболее важными из подсистем управления ресурсами являются подсистемы управления процессами, памятью, файлами и внешними устройствами, а подсистемами, общими для всех ресурсов – подсистемы пользовательского интерфейса, защиты данных и администрирования.

Основные принципы организации подсистем управления ресурсами и алгоритмы их работы излагаются в курсе лекций по дисциплине «Операционные системы» для указанных направлений подготовки. Рассматриваются не конкретные операционные системы, а принципы и концепции построения, справедливые для большинства современных операционных систем.

## Раздел 1. ОСНОВНЫЕ ПОНЯТИЯ

*Операционная система (ОС)* – это комплекс специальных программных средств, предназначенных для управления загрузкой, запуском и выполнением других (пользовательских) программ, а также для планирования и управления вычислительными ресурсами ЭВМ.

В ее состав входят взаимосвязанные программы, неоднородные по характеру и многоплановые по уровню. Этот комплекс программ динамичен по своему составу: из него можно удалять и в него добавлять некоторые составные части.

Операционные системы относятся к системному программному обеспечению.



Рисунок 1 – Классификация операционных систем

Современные операционные системы должны поддерживать мультипрограммирование, виртуальную память, многооконный графический интерфейс пользователя, но кроме этого удовлетворять, так называемым эксплуатационным требованиям, представленным на следующем рисунке.

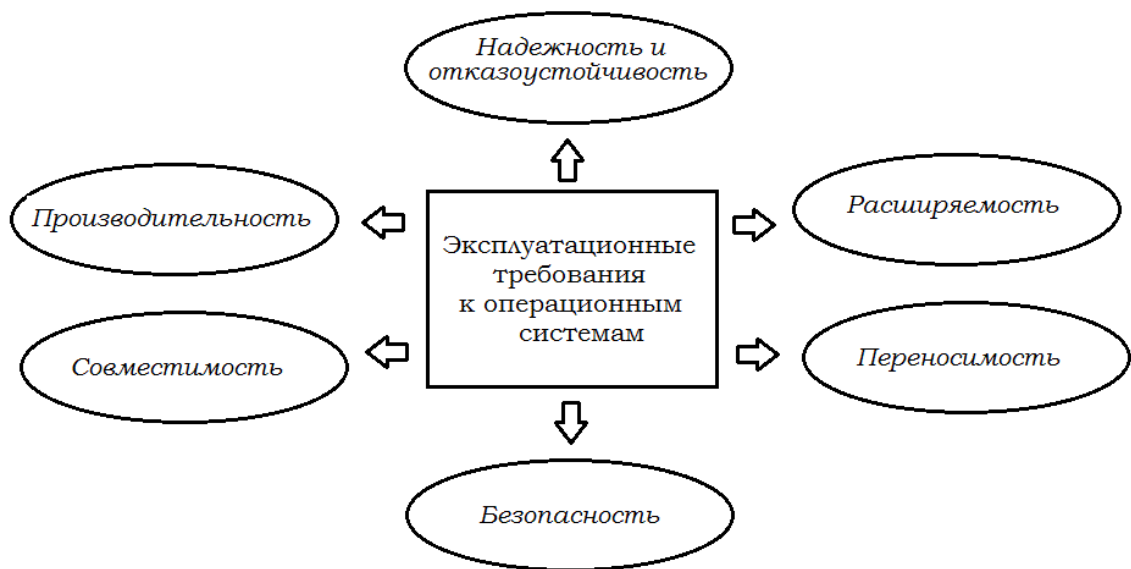


Рисунок 2 – Требования к современным операционным системам

Функционирование ОС после загрузки ЭВМ осуществляется как реакция на происходящие события, наступление которых сигнализируется прерываниями – *Interrupt*. Источниками прерываний могут быть как аппаратура, так и программы. Каждое прерывание обрабатывается соответственно обработчиком прерываний (*Interrupt handler*), входящим в состав ОС.



Рисунок 3 – Этапы обработки прерывания

Таблица 1 – Классы прерываний

Наименование	Возникновение	Примеры
<i>Внешние</i> (асинхронные) прерывания	происходят вне прерываемого процесса	прерывания внешних устройств, ввода-вывода, нарушение питания
<i>Внутренние</i> (синхронные) Прерывания	связаны с работой процессора и синхронны с его операциями	нарушение адресации, деление на нуль, ошибки защиты памяти
<i>Программные</i> прерывания	выполняются по соответствующей команде; введены для переключения на системные программные модули	переключение процессора в привилегированный режим работы



Рисунок 4 – Приоритеты прерываний

Процессор может обладать средствами защиты от прерываний: отключение системы прерываний, маскирование (запрет) отдельных сигналов. Программное управление этими средствами позволяет ОС регулировать обработку сигналов прерываний: откладывать их обработку или игнорировать.

Программное управление специальными регистрами маски (маскирование сигналов прерывания) позволяет реализовывать различные дисциплины обслуживания прерываний.



Рисунок 5 – Дисциплины обслуживания прерываний



## Раздел 2 АРХИТЕКТУРА ОПЕРАЦИОННОЙ СИСТЕМЫ

Архитектура ОС – структурная организация на основе различных программных модулей.

В макроядерной (монолитная) архитектуре все модули поделены на две группы. Ядро – модули, выполняющие основные функции операционной системы, решающие внутрисистемные задачи организации вычислительного процесса, такие как переключение контекстов, обработка прерываний и др.

Вспомогательные модули ОС выполняют полезные, но менее обязательные функции. Их обычно подразделяют на следующие группы:

- утилиты – программы, решающие отдельные задачи управления и сопровождения компьютерной системы, такие, например, как программы сжатия дисков, архивирования данных на магнитную ленту;

- системные обрабатывающие программы – текстовые или графические редакторы, компиляторы, компоновщики, отладчики;

- программы представления пользователю дополнительных услуг – специальный вариант пользовательского интерфейса, калькулятор и даже игры;

- библиотеки процедур различного назначения, упрощающие разработку приложений, например, библиотека ввода-вывода, специальных символов.

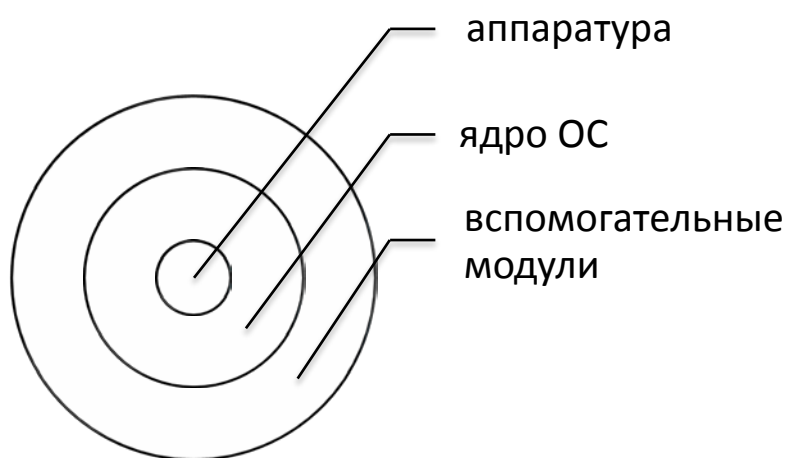


Рисунок 6 – Архитектура вычислительной системы, работающей под управлением макроядерной ОС



Рисунок 7 = Многослойная структура ядра операционной системы

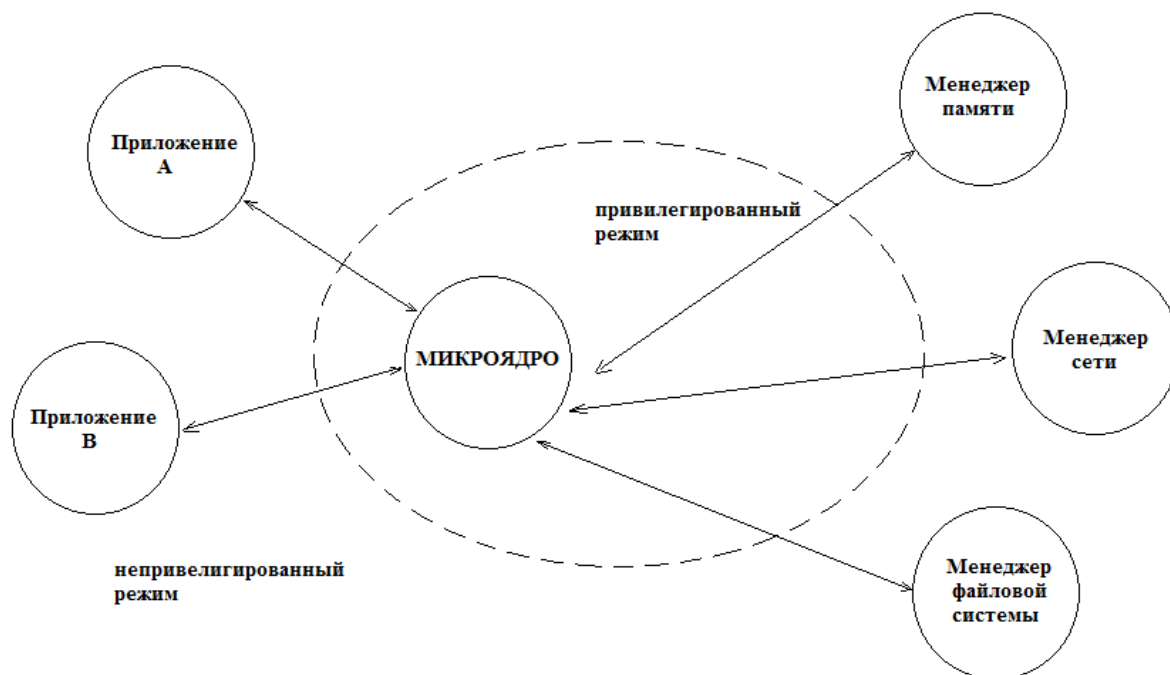


Рисунок 8 – Микроядерная архитектура операционной системы

### Раздел 3 ПРОЦЕССЫ И ПОТОКИ

Внутренние единицы работы операционной системы:  
процесс – задача на стадии выполнения;  
поток (нить) – последовательность команд, часть процесса.

*Процесс* рассматривается ОС как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени. Процессорное время распределяется ОС между – *потоками*.

При управлении процессами операционная система использует два основных типа информационных структур:

*дескриптор процесса*, содержащий информацию, необходимую ядру в течение всего жизненного цикла процесса ( о состоянии процесса, о расположении образа процесса в оперативной памяти и на диске, о значении отдельных составляющих приоритета, а также о его итоговом значении – глобальном приоритете, об идентификаторах пользователя, создавшего процесс, о родственных процессах и другая информация);

*контекст процесса*, содержащий менее оперативную, но более объемную часть информации, необходимую для возобновления выполнения процесса с прерванного места (содержимое регистров процессора, коды ошибок выполняемых процессором системных вызовов, информация обо всех файлах, открытых процессом, о незавершенных операциях ввода-вывода).

Контекст процесса также как дескриптор доступен только программам ядра, однако он хранится не в области ядра, а непосредственно примыкает к образу процесса и перемещается вместе с ним при необходимости из оперативной памяти на диск и обратно.

Для реализации мультипрограммирования на протяжении существования процесса выполнение его потоков может быть многократно прервано и продолжено. Переход от выполнения одного потока к другому осуществляется в результате планирования и диспетчеризации.

*Планирование* включает определение момента времени для смены текущего потока, а также выбор нового потока для выполнения. *Диспетчеризация* заключается в реализации найденного в результате планирования решения, т.е. в переключении процесса с одного потока на другой.

Диспетчеризация включает в себя: сохранение контекста текущего потока; загрузку контекстов нового потока, выбранного в результате планирования; запуск нового потока на выполнение.

ОС выполняет планирование потоков, принимая во внимание их состояние. В мультипрограммной системе различают три основных состояния потока:

выполнение – активное состояние потока, во время которого поток обладает всеми необходимыми ресурсами и непосредственно выполняется процессом;

ожидание – пассивное состояние потока, находясь в котором, поток заблокирован по своим внутренним причинам (ждет осуществления некоторого события, чаще всего завершения ввода-вывода, или освобождения некоторого ресурса);

готовность – пассивное состояние процесса, но при этом поток заблокирован в связи с внешними обстоятельствами.

Иногда еще добавляют состояние новый, т.е. только созданный поток, и завершённый.

В течение жизни поток переходит из одного состояния в другое в соответствии с алгоритмом планирования.

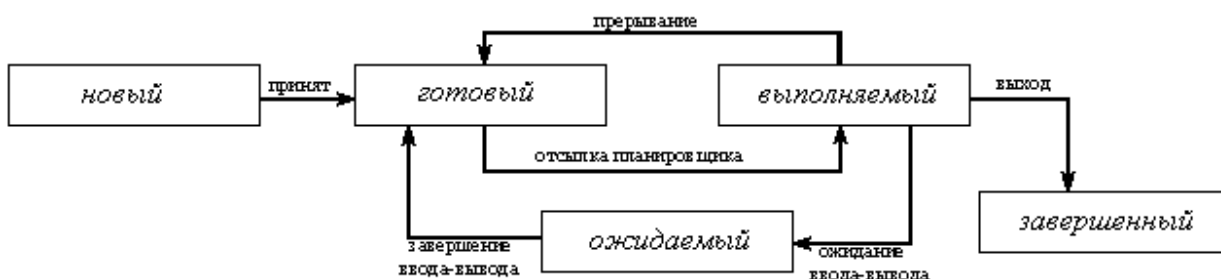


Рисунок 9 – Типовая диаграмма переходов для состояний процесса.

Квантом называют ограниченный непрерывный период процессорного времени, который предоставляется каждому потоку поочередно для выполнения.

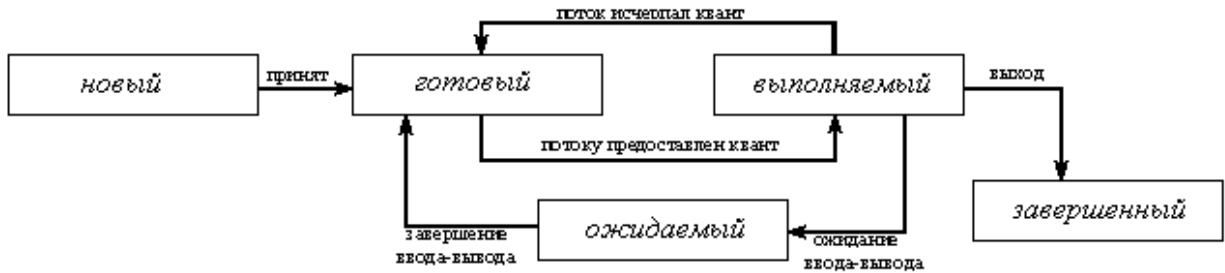


Рисунок 10 – Диаграмма переходов в алгоритме квантования



Рисунок 11 – Диаграмма переходов в алгоритме с абсолютным приоритетом

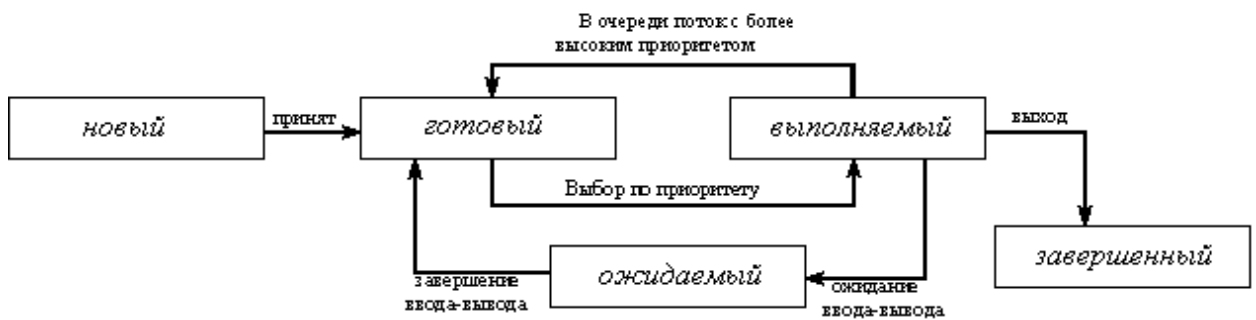


Рисунок 12 – Диаграмма переходов в алгоритме с относительным приоритетом.

## Раздел 4 Управление оперативной памятью

В различных операционных системах каждому процессу назначается либо непрерывная область памяти, либо кусочно-непрерывная; в одних заложено периодическое перемещение частей программы, загружаемых в память, в течение ее выполнения, а в других – их стационарное расположение во время всего периода выполнения процесса. Последовательность действий в случае, если программы не помещаются в имеющуюся в наличии память, также может быть различной. Все эти действия определяют алгоритмы управления памятью.



Рисунок 13 – Классификация методов распределения памяти

Таблица 2 – Особенности алгоритмов управления памятью

Наименование	Принцип действия	Достоинства	Недостатки
1	2	3	4
фиксированными разделами	Границы разделов устанавливаются при старте не изменяются	Простота	не позволяет выполнить процессы, образы которых не помещаются ни в один из разделов
динамическими разделами	Каждому вновь поступающему на выполнение приложению на этапе создания процесса выделяется вся необхо-	Выполнение процессов большого объема	Если достаточный объем памяти отсутствует, то приложение не принимается, и

1	2	3	4
	дима ему память.		процесс не создается. Фрагментация памяти
перемещаемыми разделами	перемещение всех занятых участков в сторону старших или младших адресов так, чтобы вся свободная память образовывала единую свободную память.	Отсутствие фрагментации	процедура сжатия может потребовать значительного времени
страничная виртуальная память	организует перемещение данных между памятью и диском страницами – частями виртуального адресного пространства, фиксированного и сравнительно небольшого размера	простота	сложность организации защиты данных
сегментная виртуальная память	предусматривает перемещение данных сегментами – частями виртуального адресного пространства произвольного размера, полученными с учетом смыслового значения данных	возможность использования разделяемых данных разными процессами, защита данных	избыточный объем перемещаемой информации Фрагментация памяти
сегментно-страничная организация	использует двухуровневое деление: виртуальное адресное пространство делится на сегменты, а затем сегменты делятся на страницы	сочетает преимущества страничного и сегментного распределений	

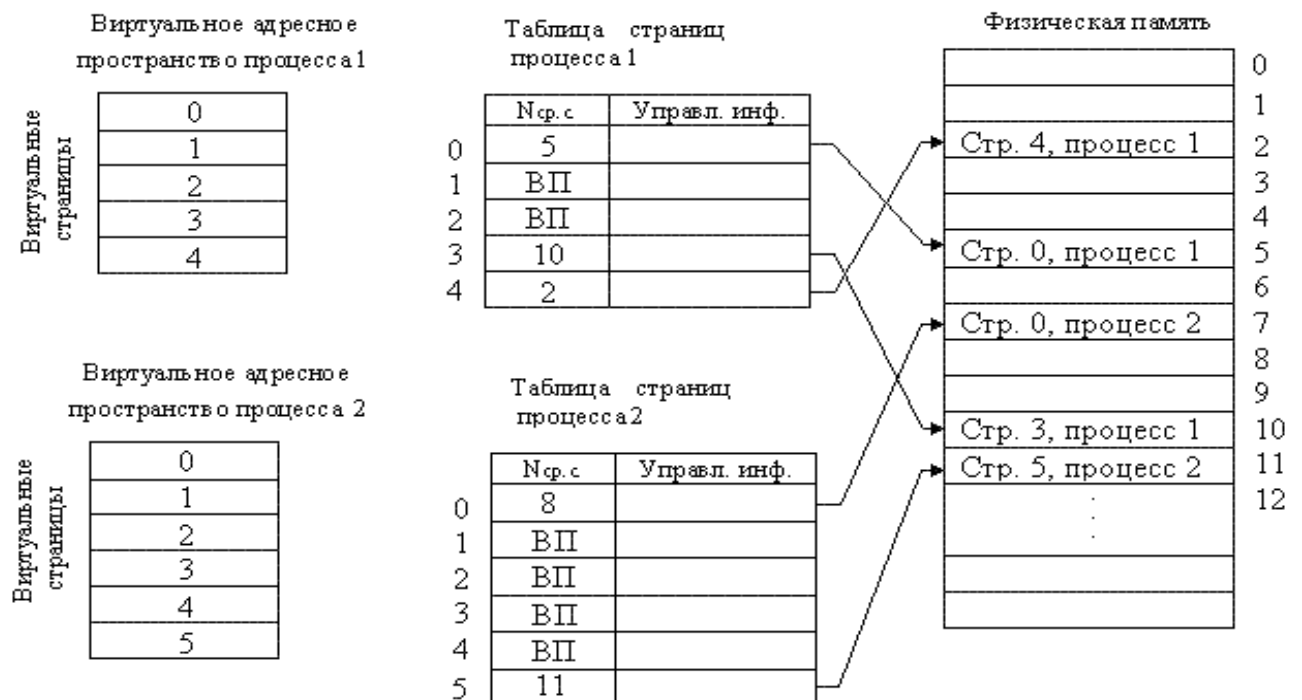


Рисунок 14 – Страничное распределение памяти

Дескриптор страницы и содержит следующую информацию:

номер физической страницы, в которую загружена данная виртуальная страница;

признак присутствия, устанавливаемый в единицу, если виртуальная страница находится в оперативной памяти;

признак модификации страницы, который устанавливается в единицу всякий раз, когда производится запись по адресу, относящемуся к данной странице;

признак обращения к странице, называемой также битом доступа, который также устанавливается в единицу при каждом обращении по адресу, относящемуся к данной странице.

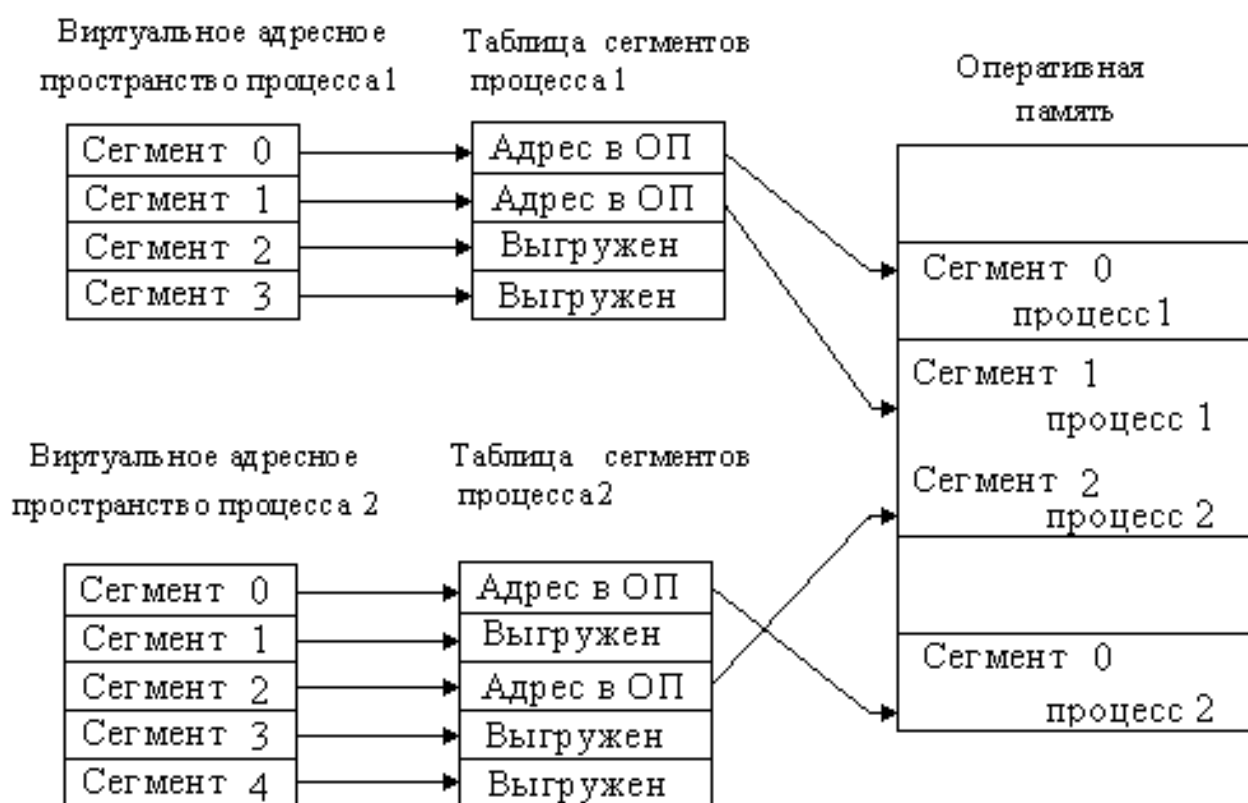


Рисунок 15 – Сегментное распределение памяти



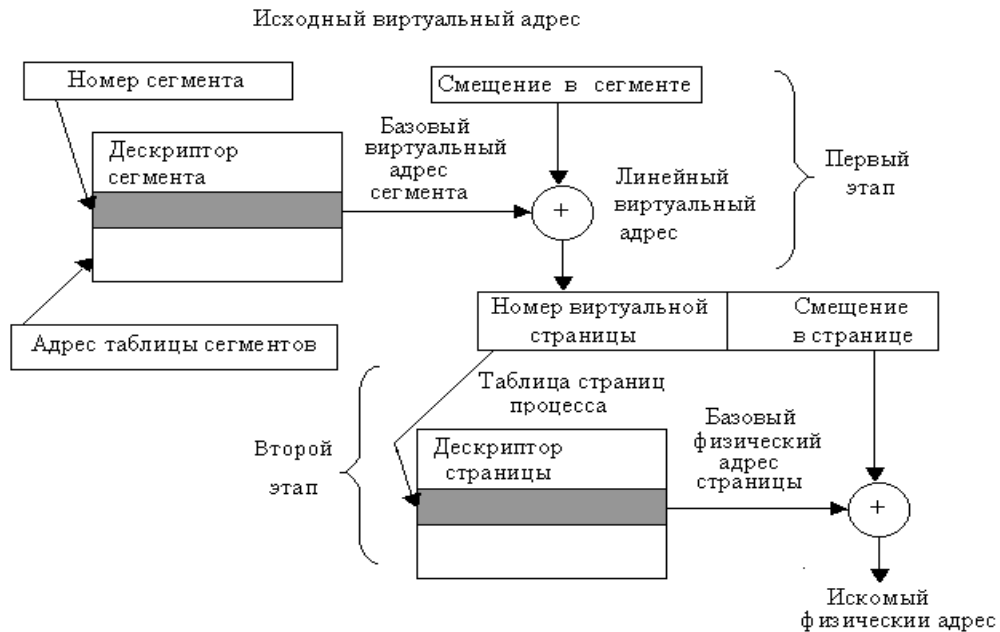


Рисунок 16 – Схема преобразования виртуального адреса в физический при сегментно-страничной организации памяти

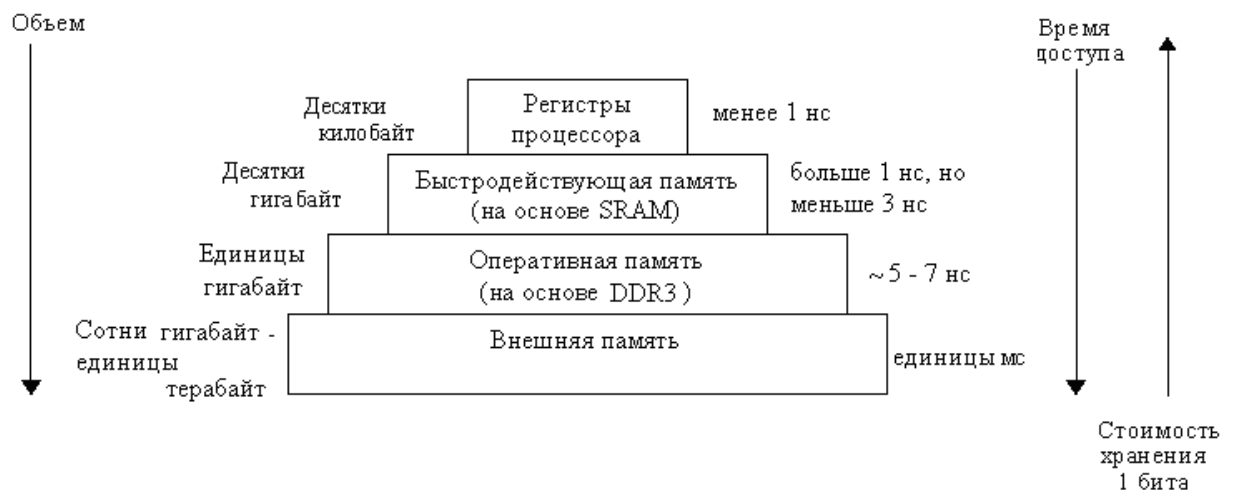


Рисунок 17 – Иерархия запоминающих устройств

*Кэш-память* – способ совместного функционирования двух типов запоминающих устройств, отличающихся временем доступа и стоимостью хранения данных, который за счет динамического копирования в «быстрое» запоминающее устройство наиболее часто используемой информации из «медленного» позволяет уменьшить среднее время доступа к данным и экономить более дорогую быстродействующую память. Неотъемлемым свойством кэш-памяти является ее прозрачность для программ и пользователей.

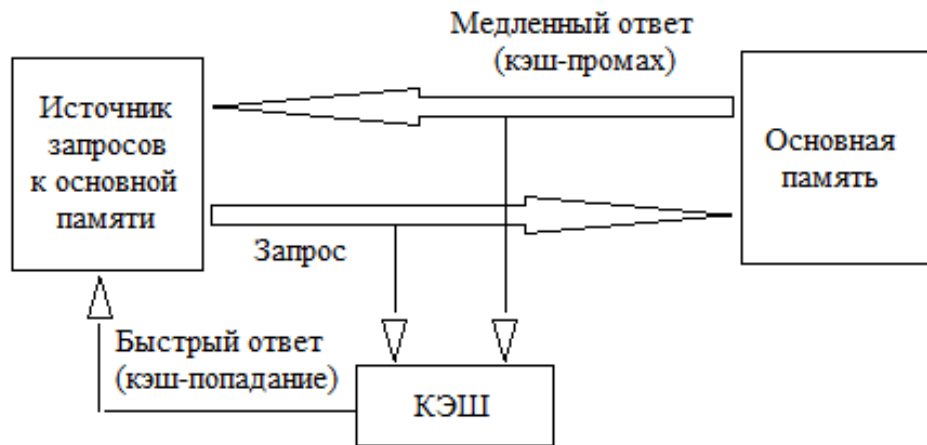


Рисунок 18 – Принцип действия кэш-памяти

При каждом обращении к основной памяти по физическому адресу просматривается содержимое кэш-памяти. Если данные обнаруживаются в кэш-памяти, т.е. произошло кэш-попадание, они считываются, и результат передается источнику запроса. Если же нужные данные отсутствуют в кэш-памяти (кэш-промах), они считываются из основной памяти, передаются источнику запроса и одновременно с этим копируются в кэш-память.

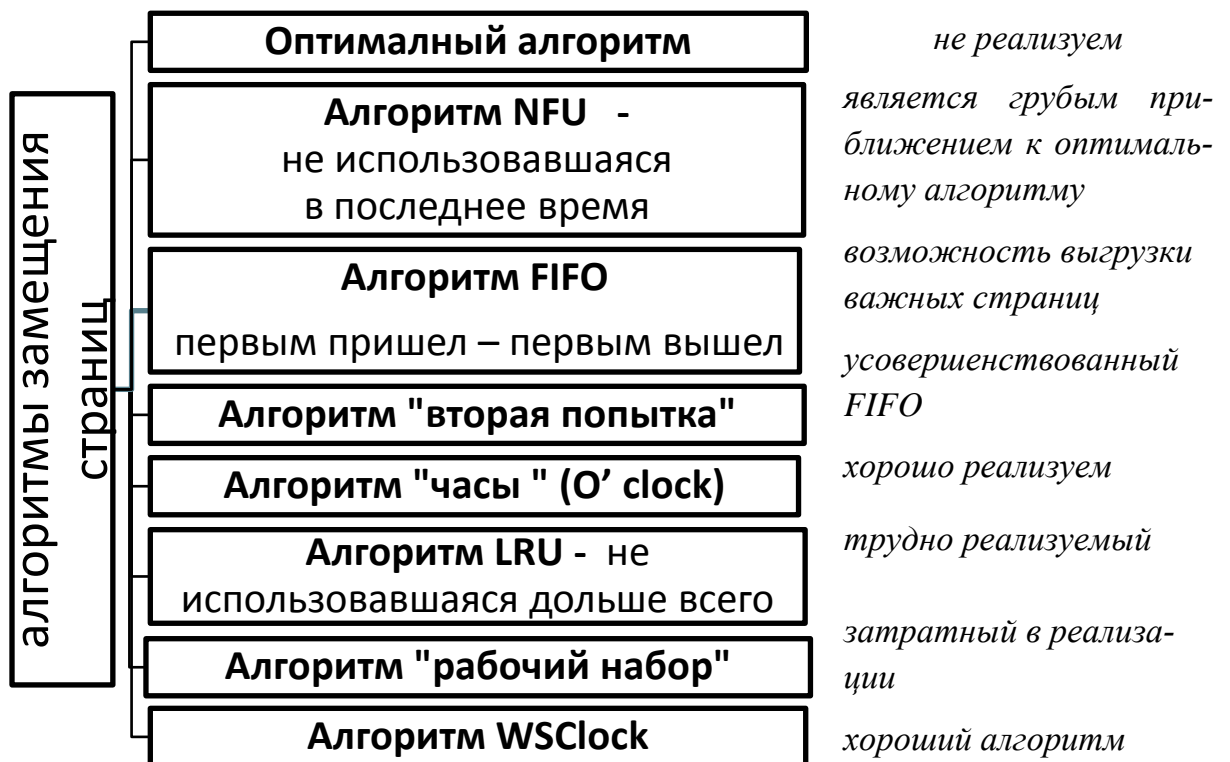


Рисунок 19 – Особенности алгоритмов замещения страниц

## Раздел 5. Подсистема ввода-вывода

Существуют разделяемые и неразделяемые устройства ввода-вывода. Примерами разделяемого устройства могут служить накопитель на магнитных дисках, устройство чтения компакт-дисков. Это устройства с прямым доступом. Примеры неразделяемых устройств – принтер, накопитель на магнитных лентах. Это устройства с последовательным доступом.



Рисунок 20 – Режимы управления вводом-выводом

Существует два основных режима ввода-вывода: *режим обмена с опросом готовности* устройства ввода-вывода и *режим обмена с прерываниями*.

В первом случае центральный процессор посылает команду устройству управления. Устройство управления исполняет команду, транслируя сигналы, понятные ему и центральному устройству, в сигналы, понятные устройству ввода-вывода. После выполнения команды устройство ввода-вывода (или его устройство управления) выдает *сигнал готовности*, который сообщает процессору о возможности выполнения новой команды для продолжения обмена данными. Однако, поскольку быстродействие устройства ввода-вывода намного меньше быстродействия центрального, сигнал готовности приходится очень долго ожидать, постоянно опрашивая соответствующую

линию интерфейса, что приводит к нерациональному использованию процессорного времени. Гораздо выгоднее сигнал готовности трактовать как запрос на прерывание от устройства ввода-вывода, как и делается во втором случае.

Таблица 3 – Системные таблицы ввода-вывода

<b>Наименование</b>	<b>Назначение</b>	<b>Содержимое</b>
<i>таблица оборудования</i>	хранение информации обо всех устройствах системы	тип устройства, символьное имя, характеристики; способ подключения устройства (интерфейс, разъем, порты, линия запроса прерывания); номер и адрес канала; информацию о драйвере, адреса секции запуска и секции продолжения драйвера; имя или адрес буфера; установка тайм-аута и ячейки для счетчика тайм-аута; состояние устройства; поле указателя для связи задач, ожидающих устройство.
<i>таблица виртуальных логических устройств</i>	установление связи между виртуальными (логическими) устройствами и реальными устройствами	какое конкретно устройство и каким образом подключено к компьютеру
<i>таблица прерываний</i>	для организации обратной связи между центральной частью и устройствами ввода-вывода	для каждого сигнала запроса на прерывание указывает на элемент таблицы оборудования, который сопоставлен нужному устройству

Любые операции по управлению вводом-выводом объявляются привилегированными и могут выполняться только кодом самой операционной системы.

Управление вводом-выводом осуществляется компонентом операцион-

ной системы, который часто называют супервизором ввода-вывода.

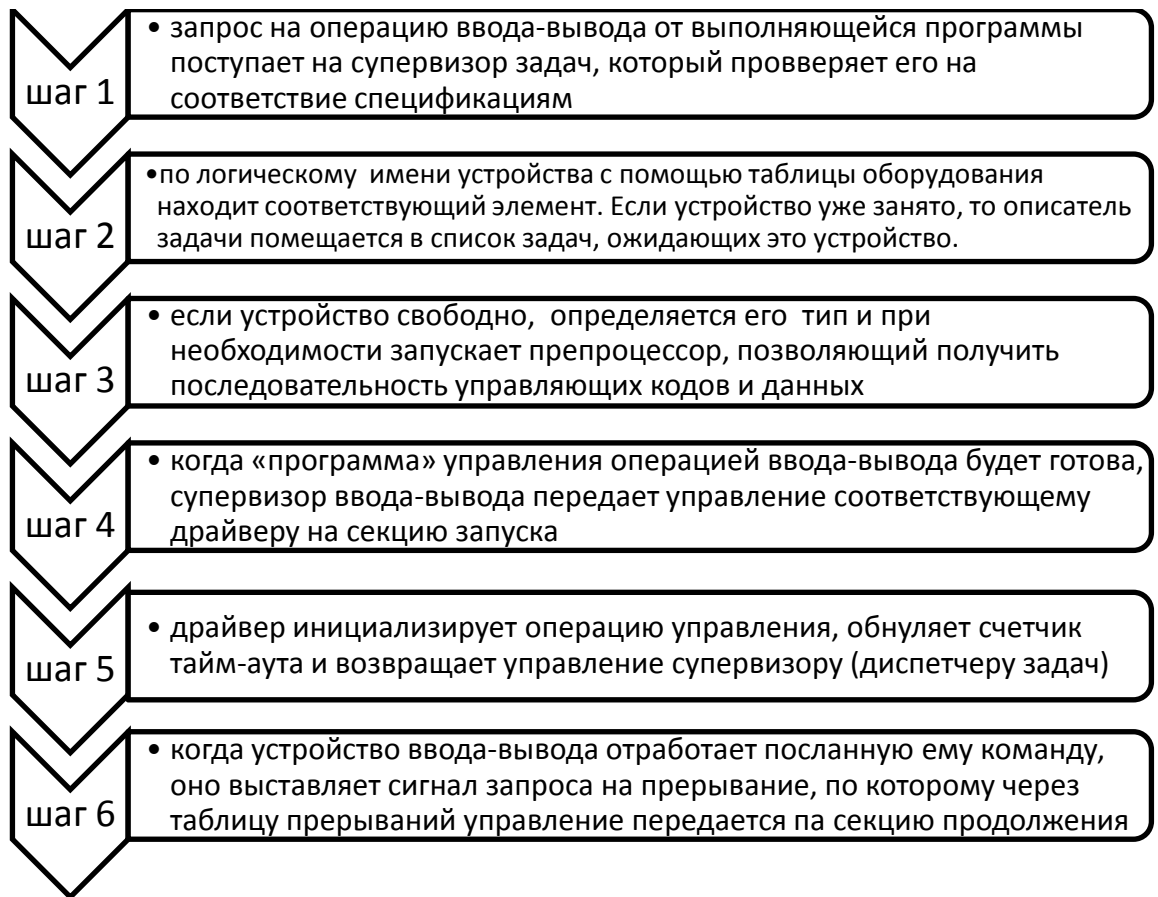
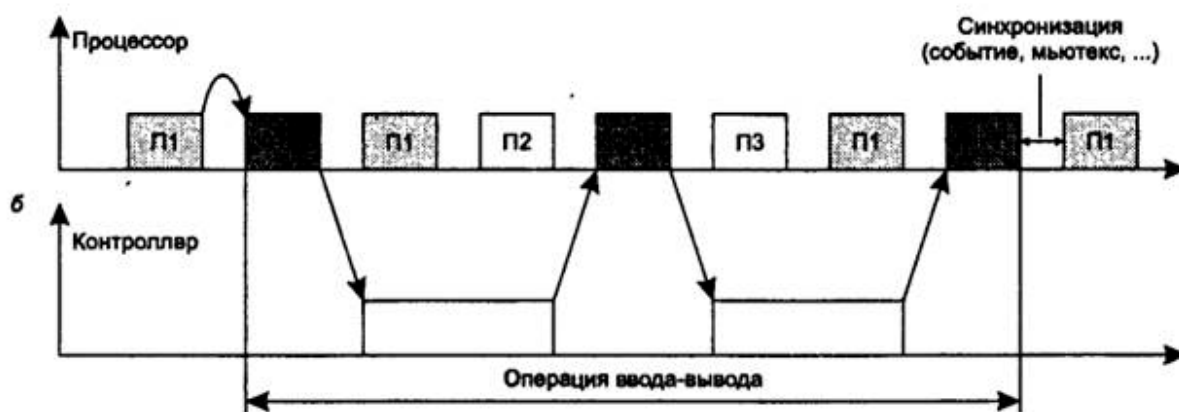
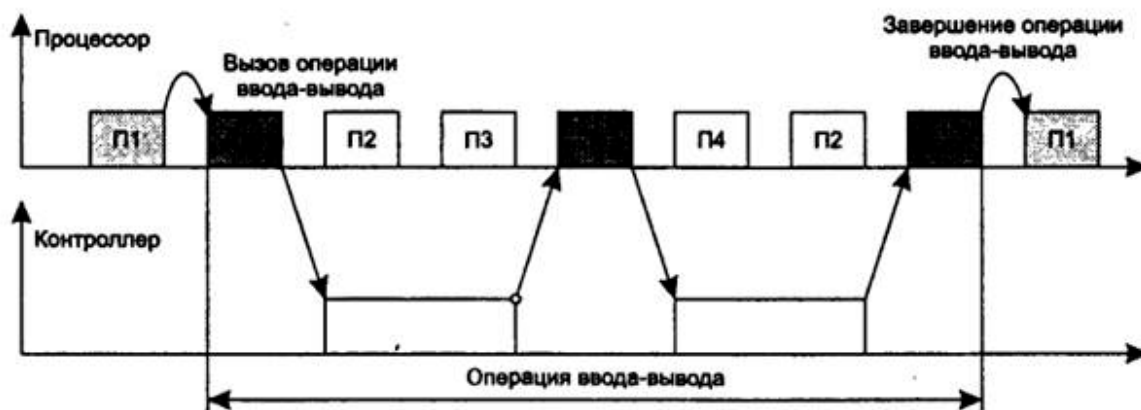


Рисунок 21 – Шаги работы супервизора ввода-вывода

Задача, выдавшая запрос на операцию ввода-вывода, переводится супервизором в состояние ожидания завершения заказанной операции. Когда супервизор получает от секции завершения сообщение о том, что операция завершилась, он переводит задачу в состояние готовности к выполнению, и она продолжает выполняться. Эта ситуация соответствует *синхронному вводу-выводу*. Синхронный ввод-вывод является стандартным для большинства операционных систем. Для увеличения скорости выполнения приложений, можно использовать *асинхронный ввод-вывод*.



- п1 — Вызывающая процедура
- Процедура ввода-вывода
- п2 — Другие процессы

Рисунок 22 – Синхронный и асинхронный ввод-вывод

## Раздел 6. Файловая система

Одной из основных задач ОС является предоставление пользователю удобств при работе с данными, хранящимися на дисках. Для этого ОС подменяет физическую структуру хранящихся данных некоторой удобной для пользователя логической моделью. Логическая модель файловой системы материализуется в виде дерева каталогов.

Главная функция файловой системы – это распределение пространства внешней памяти и управление ее работой, в частности работой дисковой памяти.

NTFS (New Technology File System) – файловая система новой технологии. Действительно, файловая система NTFS по сравнению FAT32 содержит ряд значительных усовершенствований и изменений, хотя с точки зрения пользователей файлы по-прежнему хранятся в каталогах, называемых папками.

NTFS делит все полезное дисковое пространство тома на кластеры (блоки данных, адресуемые как единицы данных). Поддерживаются размеры кластеров от 512 байт до 64 Кбайт; неким стандартом же считается кластер размером 2 или 4 Кбайтов. При увеличении размера кластера свыше 4 Кбайт становится невозможным сжимать файлы и каталоги.

Все дисковое пространство в NTFS делится на две неравные части. Первые 12 % диска отводятся под так называемую зону MFT (Master File Table — главная таблица файлов). Остальные 88 % тома представляют собой обычное пространство для хранения файлов.

Зона MFT	Зона размещения файлов и каталогов	Копия первых 16 записей MFT	Зона размещения файлов и каталогов
----------	------------------------------------	-----------------------------	------------------------------------

Рисунок 23 – Структура тома NTFS

## Литература

Гордеев А.В. Операционные системы. – СПб.: Питер, 2009 – 416 с.

Олифер В.Г., Олифер Н.А. Сетевые операционные системы. – СПб.: Питер, 2009 – 669 с.

Таненбаум Э.С., Бос Х. Современные операционные системы – СПб.: Питер, 2015 – 1120 с.

Татьяна Алексеевна Галаган, *доцент кафедры ИиУС АмГУ*  
(*составитель*)

**Операционные системы. Учебно-методическое пособие**

---