



Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Институт повышения квалификации
и переподготовки

Кафедра «Профессиональная переподготовка»

Е. И. Гридина

АРХИТЕКТУРА ОПЕРАЦИОННЫХ СИСТЕМ

ПРАКТИКУМ

**по одноименной дисциплине
для слушателей специальности 1-40 01 73
«Программное обеспечение
информационных систем»
заочной формы обучения**

Гомель 2016

УДК 004.65(075.8)
ББК 32.973-018.2я73
Г83

*Рекомендовано кафедрой «Профессиональной переподготовки»
ИПКиП ГГТУ им. П. О. Сухого
(протокол № 4 от 14.12.2015 г.)*

Рецензент: зав. каф. «Информатика» ГГТУ им. П. О. Сухого
канд. физ.-мат. наук, доц. *Т. В. Тихоненко*

Гридина, Е. И.

Г83 Архитектура операционных систем : практикум по одной дисциплине для слушателей специальности 1-40 01 73 «Программное обеспечение информационных систем» заоч. формы обучения / Е. И. Гридина. – Гомель : ГГТУ им. П. О. Сухого, 2016. – 34 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

Данное издание представляет собой пособие по практическому освоению основ проектирования операционных систем, предполагающих знание структур данных и алгоритмов работы компонентов операционной системы. Учебное издание написано доступным языком и содержит множество примеров, позволяющих в кратчайшие сроки овладеть основными принципами функционирования операционных систем.

Для слушателей специальности 1-40 01 73 «Программное обеспечение информационных систем» ИПКиП.

**УДК 004.65(075.8)
ББК 32.973-018.2я73**

© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2016

Содержание

Лабораторная работа № 1	4
Лабораторная работа № 2	14
Лабораторная работа № 3	22
Лабораторная работа № 4	27
Список использованных источников	34

Лабораторная работа № 1

Тема: Определение состава операционных систем

Цель работы: Научиться получать данные, используя операционную систему.

Теоретические сведения

Операцио́нная систе́ма, сокр. ОС (*operating system, OS*) — комплекс взаимосвязанных программ, предназначенных для управления ресурсами вычислительного устройства и организации взаимодействия с пользователем.

Основные функции операционной системы:

- исполнение запросов программ (ввод и вывод данных, запуск и остановка других программ, выделение и освобождение дополнительной памяти и др.);
- загрузка программ в оперативную память и их выполнение;
- стандартизованный доступ к периферийным устройствам (устройства ввода-вывода);
- управление оперативной памятью (распределение между процессами, организация виртуальной памяти);
- управление доступом к данным на энергонезависимых носителях (таких как жёсткий диск, оптические диски и др.), организованным в той или иной файловой системе;
- обеспечение пользовательского интерфейса;
- сохранение информации об ошибках системы.

Под архитектурой операционной системы понимают структурную и функциональную организацию ОС на основе некоторой совокупности программных модулей. В состав ОС входят исполняемые и объектные модули стандартных для данной ОС форматов, программные модули специального формата (например, загрузчик ОС, драйверы ввода-вывода), конфигурационные файлы, файлы документации, модули справочной системы и т.д.

Большинство современных ОС представляют собой хорошо структурированные модульные системы, способные к развитию, расширению и переносу на новые платформы. Какой-либо единой унифицированной архитектуры ОС не существует, но известны

универсальные подходы к структурированию ОС. Принципиально важными универсальными подходами к разработке архитектуры ОС являются:

- модульная организация;
- функциональная избыточность;
- функциональная избирательность;
- параметрическая универсальность;
- концепция многоуровневой иерархической вычислительной системы, по которой ОС представляется многослойной структурой;
 - разделение модулей на две группы по функциям: ядро – модули, выполняющие основные функции ОС, и модули, выполняющие вспомогательные функции ОС;
 - разделение модулей ОС на две группы по размещению в памяти вычислительной системы: резидентные, постоянно находящиеся в оперативной памяти, и транзитные, загружаемые в оперативную память только на время выполнения своих функций;
 - реализация двух режимов работы вычислительной системы: привилегированного режима (режима ядра – Kernel mode), или режима супервизора (supervisor mode), и пользовательского режима (user mode), или режима задачи (task mode);
 - ограничение функций ядра (а следовательно, и количества модулей ядра) до минимального количества необходимых самых важных функций.


Задание

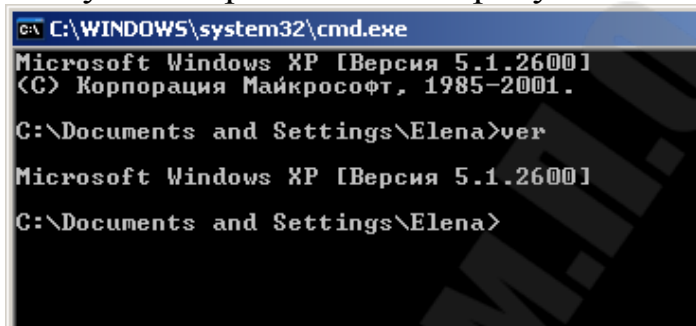
Все задания необходимо выполнить в текстовом интерфейсе командной строки Windows. Выполнение каждой команды необходимо подтвердить в отчете снимком окна командной оболочки с самой командой и результатом ее выполнения.

Получить сведения:

1. о версии операционной системы
2. получить сведения о ревизии операционной системы
3. описать все переменные среды (и их параметры) операционной системы
4. указать всех пользователей, зарегистрированных в операционной системе
5. указать исходный и максимальный размер файла подкачки, заданный в операционной системе
6. указать какие файловые системы используются на логических дисках.

Пояснения к выполнению работы


Версию операционной системы можно узнать несколькими способами. Для нашей операционной системы мы либо можем воспользоваться в режиме командной строки командой VER, либо запустить приложение Winver. Приведем результат, полученный обоими способами. Чтобы вызвать окно с командной строкой нажмем кнопку Пуск , затем выбираем команду Выполнить, вводим cmd и нажимаем клавишу Enter. В появившемся окне вводим команду VER и нажимаем Enter. Результат представлен на рисунке 1.1.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\Elena>ver
Microsoft Windows XP [Версия 5.1.2600]
C:\Documents and Settings\Elena>
```

Рисунок 0.1 – Результат выполнения команды ver

Чтобы воспользоваться приложением Winver, нажмите кнопку Пуск , выберите команду Выполнить, введите winver и нажмите клавишу Enter. Результат представлен на рисунке 1.2

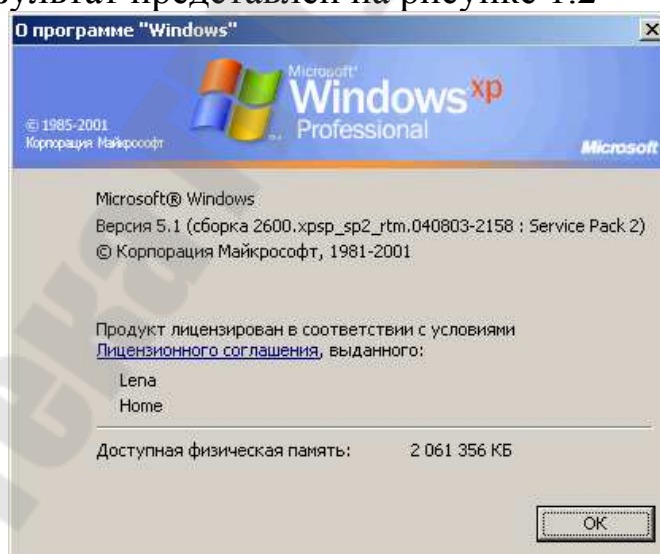


Рисунок 0.2 – Результат выполнения приложения Winver

Как видно из результатов работы команд, на компьютере установлена операционная система Windows XP версии 5.1.

Сведения о ревизии операционной системы

Чтобы определить сборку операционной системы воспользуемся информацией, полученной выше. Ревизия операционной системы: 2600 Service Pack 2.

Переменные среды операционной системы

Переменная среды (переменная окружения) в Windows используется для хранения текстовых строк пользователя и информации о настройках операционных систем.

С переменными среды можно работать, нажав кнопку «Переменные среды» на вкладке «Дополнительно» в диалоговом окне «Свойства системы» (контекстное меню значка «Мой компьютер» → пункт «Свойства»). Данное окно отображено на рисунке 1.3.

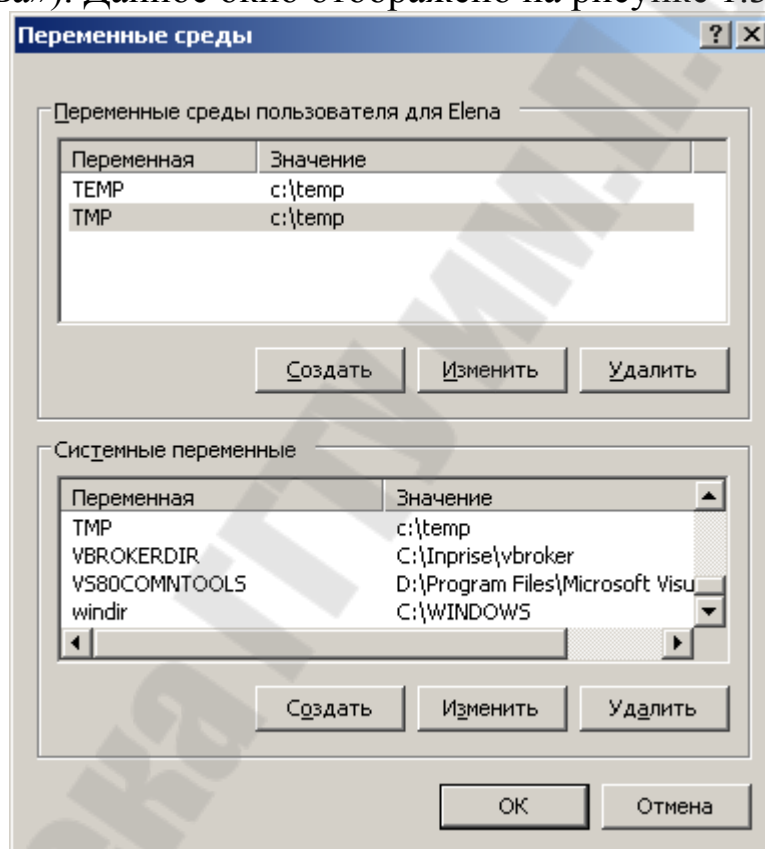


Рисунок 0.3 – Окно для работы с переменными среды

В используемой нами операционной системе определены следующие переменные среды (в скобках будем указывать значение для нашей ОС):

COMSPEC (C:\Windows\system32\cmd.exe) – путь к исполняемому файлу командного процессора (shell).

NUMBER_OF_PROCESSORS (2) – Количество процессоров в системе (фактически, количество ядер).

OS (Windows_NT) – Название операционной системы. Windows XP и Windows 2000 отображаются как Windows_NT Path (C:\Program

Files\Common Files\Pervasive Software Shared\pvsware;C:\PROGRA~1\Borland\CBUILD~1\Bin;C:\PROGRA~1\Borland\CBUILD~1\Projects\Bpl;C:\Inprise\vbroke\bin;C:\Program Files\Borland\Delphi7\Bin;C:\Program Files\Borland\Delphi7\Projects\Bpl; C:\WINDOWS\system32; C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program Files\MySQL\MySQL Server 5.5\bin) – указывает путь поиска исполняемых файлов.

PATHEXT (.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH) – Возвращает список расширений файлов, которые рассматриваются операционной системой как исполняемые.

PROCESSOR_ARCHITECTURE (x86) – Архитектура процессора. Возможные варианты: x86, IA64, AMD64.

PROCESSOR_IDENTIFIER (x86 Family 6 Model 23 Stepping 6, GenuineIntel) – Описание процессора.

PROCESSOR_LEVEL (6) – Номер модели процессора.

PROCESSOR_REVISION (1706) – Ревизия процессора.

TMP и TEMP (c:\temp) – Возвращает временные каталоги, по умолчанию используемые приложениями, которые доступны пользователям, выполнившим вход в систему. Некоторые приложения требуют переменную TEMP, другие — переменную TMP. Потенциально TEMP и TMP могут указывать на разные каталоги, но обычно совпадают.

Windir (C:\WINDOWS) – Каталог, в котором установлена Windows.

Пользователи, зарегистрированные в операционной системе

Учётная запись пользователя – это запись, которая содержит сведения, необходимые для идентификации пользователя при подключении к системе, а также информацию для авторизации и учёта. Это имя пользователя и пароль (или другое аналогичное средство аутентификации — например, биометрические характеристики). Пароль или его аналог, как правило, хранится в зашифрованном или хэшированном виде (в целях его безопасности).

Учетные записи пользователей можно просмотреть несколькими способами:

При помощи диалога «Учетные записи пользователей». Нажмем на кнопку «Пуск» для открытия меню, откроем «Панель управления» и из списка компонентов панели управления выбираем «Учетные записи пользователей». Результат представлен на рисунке 1.4.

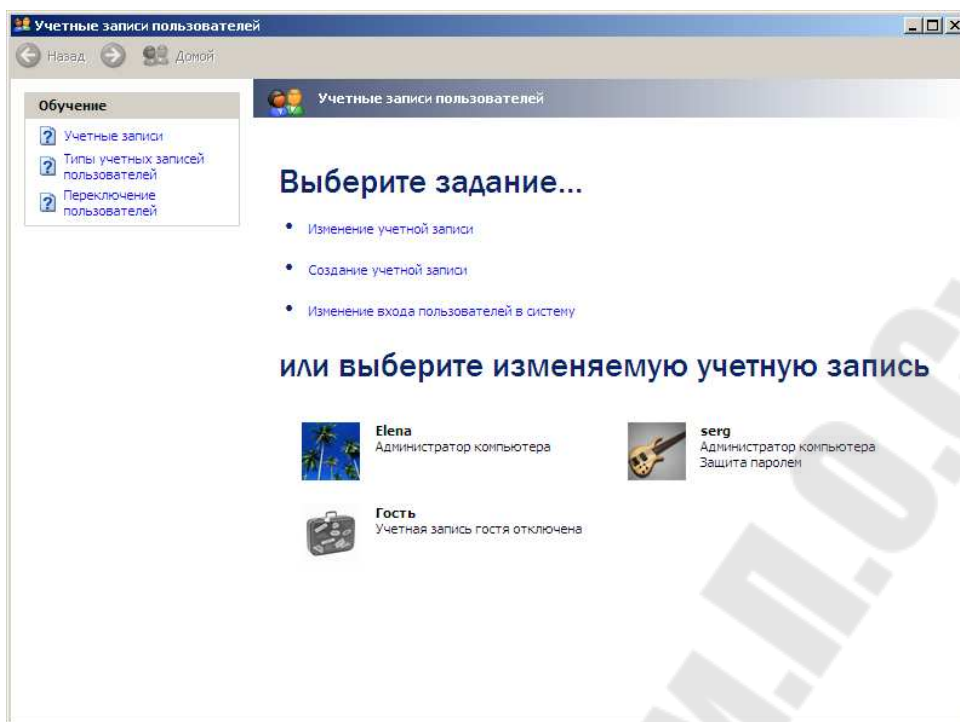


Рисунок 0.4 – Окно с учетными записями пользователей

Второй способ – оснастка «Локальные пользователи и группы»

Оснастка «Локальные пользователи и группы» расположена в компоненте «Управление компьютером», представляющем собой набор средств администрирования, с помощью которых можно управлять одним компьютером, локальным или удаленным. Оснастка «Локальные пользователи и группы» служит для защиты и управления учетными записями пользователей и групп, размещенных локально на компьютере. Можно назначать разрешения и права для учетной записи локального пользователя или группы на определенном компьютере (и только на этом компьютере).

Использование оснастки «Локальные пользователи и группы» позволяет ограничить возможные действия пользователей и групп путем назначения им прав и разрешений. Право дает возможность пользователю выполнять на компьютере определенные действия, такие как архивирование файлов и папок или завершение работы компьютера. Разрешение представляет собой правило, связанное с объектом (обычно с файлом, папкой или принтером), которое определяет, каким пользователям, и какой доступ к объекту разрешен.

Нажмем на кнопку «Пуск» для открытия меню, откроем «Панель управления» и из списка компонентов панели управления выбираем «Администрирование», затем откроем компонент «Управление компьютером». В «Управлении компьютером» откроем «Локальные пользователи и группы».

Результат представлен на рисунке 1.5.

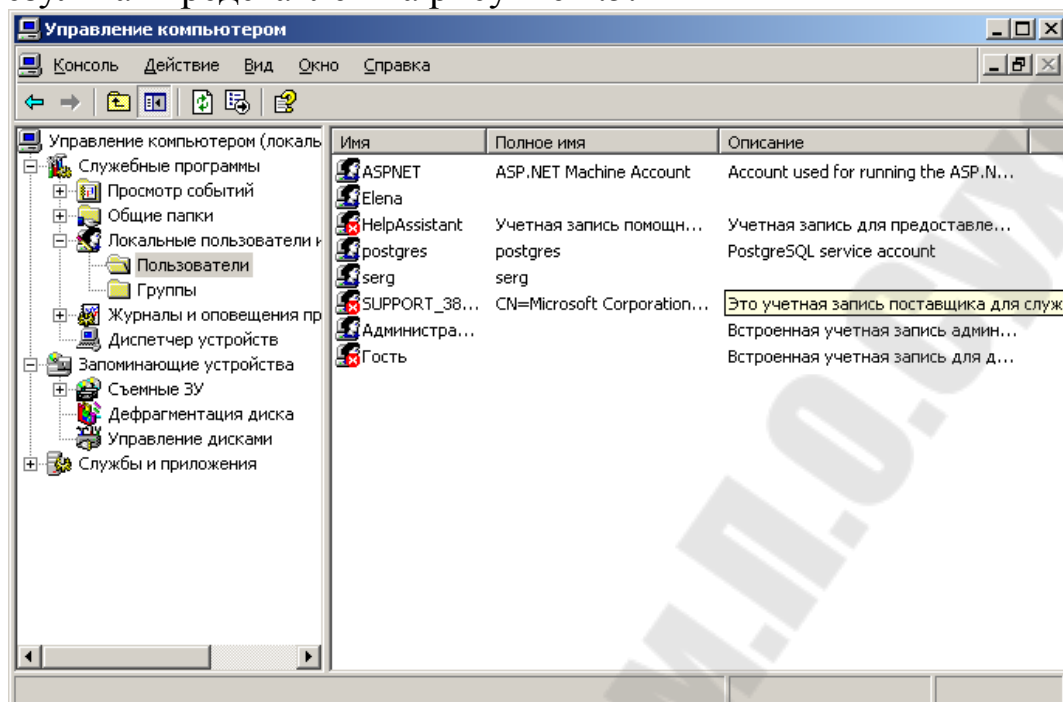


Рисунок 0.5 – Окно оснастки «Локальные пользователи и группы»

Таким образом, мы видим, что помимо встроенных учетных записей (Администратор, Гость) в системе зарегистрированы пользователи Serg и Elena.

Размер файла подкачки

Файл подкачки в Microsoft Windows XP расширяет объем доступной для приложений и служб памяти, позволяя им не ограничиваться физически установленной на компьютере RAM.

Чтобы просмотреть параметры имеющегося файла подкачки в системе Windows XP необходимо сделать нижеследующее: Кликнуть кнопку «Пуск» → пункт меню «Панель управления» → раздел «Система». В свойствах системы необходимо щёлкнуть на вкладку под названием «Дополнительно» и в разделе, именуемом «Быстродействие», выбрать пункт «Параметры». В раскрывшемся окне также щёлкнуть на кнопку под названием «Дополнительно» и в последующей вкладке выбрать виртуальную память «изменить». Результат выполненных действий представлен на рисунке 1.6.

Исходя из полученной информации можно сказать, что файл подкачки располагается на диске D:, и его размер фиксирован и равен 2000 Мб.

Файловые системы логических дисков

Дисковое пространство делится на сектора, размер каждого из которых составляет примерно 512 байт. Сектора группируются в кластеры. Кластеры, называемые также блоками размещения, имеют размеры от 512 байт до 64 килобайт, поскольку обычно состоят из нескольких секторов. Кластеры представляют собой непрерывные блоки дискового пространства.

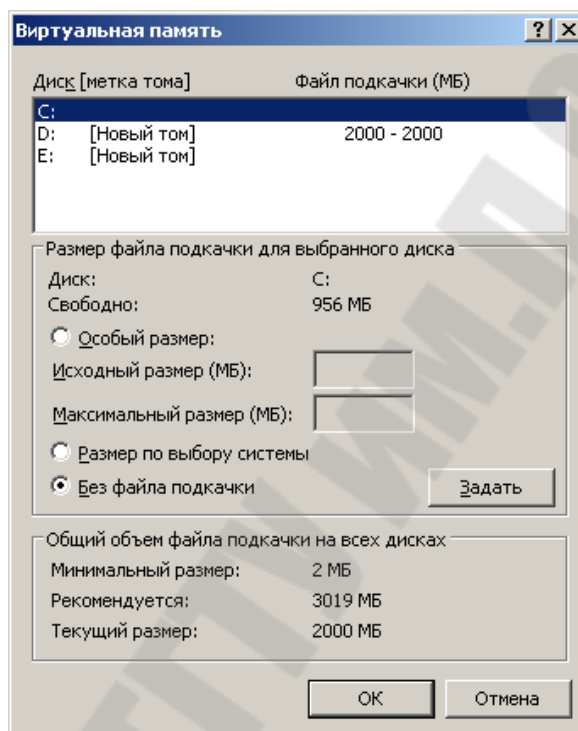


Рисунок 0.6 – Окно управления виртуальной памятью

Операционные системы опираются на файловую систему для организации кластерного хранилища информации. Файловая система представляет собой базу данных, в которой содержится информация о состоянии каждого кластера. По сути, файловая система как бы сообщает операционной системе о том, в каком кластере (или кластерах) хранится файл, а куда можно записать новые данные.

Файловая система указана в свойствах диска. Зайдем в Мой компьютер. В разделе Жесткие диски можно увидеть все жесткие диски, которые присутствуют в системе. Результат показан на рисунке 1.7.

Как видно из этого рисунка, в системе присутствует три логических диска (C, D и E).



Рисунок 0.7 – Окно «Мой компьютер»

Кликнем правой кнопкой мыши на нужном диске и далее в контекстном меню выберем Свойства. Во вкладке Общие в строке Файловая система мы увидим тип файловой системы.

На рисунках 1.8 представим характеристики одного из логических дисков.

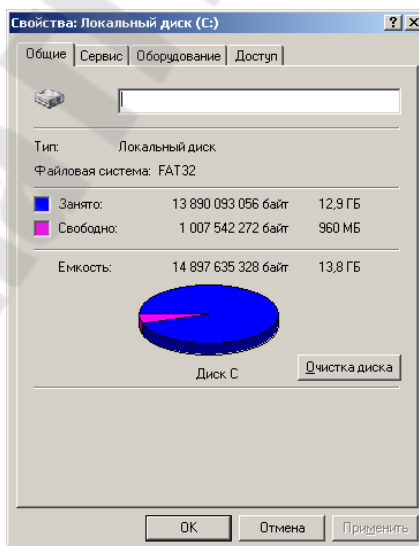


Рисунок 0.8 – Окно свойств логического диска C

На диске C: используется файловая система FAT32.

Контрольные вопросы

1. Что такое операционная система?
2. Что относится к основным функциям операционной системы?
3. Что понимают под архитектурой операционной системы?
4. Что входит в состав операционной системы?
5. Какие принципиально важные подходы необходимо применять к разработке ОС ?
6. Как можно узнать версию операционной системы?
7. Что представляет собой переменная среды операционной системы ?
8. Что представляет собой учётная запись пользователя?
9. Для чего в операционной системе используется файл подкачки?
10. Для чего используются файловые системы?

Лабораторная работа № 2

Тема: Управление процессами

Цель работы: освоить основы организации мультипрограммного режима и режима разделения времени работы операционных систем, оценить преимущества мультипрограммирования и режима разделения времени.

Теоретические сведения

Ключом к пониманию функционирования средств управления процессорами (процессами) современных ОС является диаграмма состояний процесса. Модули ОС обеспечивают изменение состояний процессов в соответствии с диаграммой, переключение контекста процессора для запуска текущего процесса, вызов подсистемы ввода-вывода для инициирования операций на внешних устройствах.

Основными состояниями процесса являются:

- представление;
- готовность;
- выполнение;
- блокировка;
- завершение.

В состоянии представления запущенному процессу выделяются требуемые ресурсы; в состоянии готовности процесс обладает всеми необходимыми для выполнения ресурсами за исключением центрального процессора (ЦП); в состоянии выполнения программный код процесса фактически выполняется на процессоре; в состоянии блокировки процесс ожидает события (например, завершения ввода-вывода); в состоянии завершения процесс освобождает ресурсы ОС.

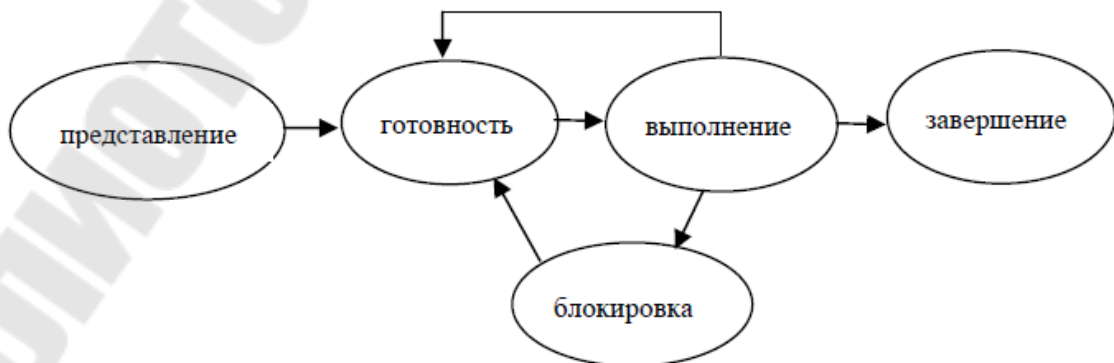


Рисунок 2.1 – Абстрактная диаграмма состояний

При проектировании ОС абстрактная диаграмма дополняется множеством состояний, отображающих различные причины блокировок и этапы выделения (освобождения) ресурсов.

Первые вычислительные системы (ВС) работали в однопрограммном режиме. ОС начинала выполнение нового задания лишь в случае полного завершения текущего задания.

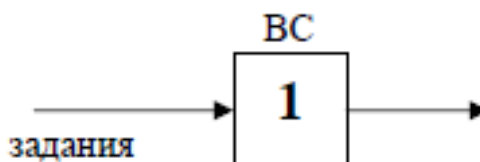


Рисунок 2.2 – Однопрограммный режим работы

Это приводило к неэффективному использованию ресурсов вычислительной системы в связи с простоем оборудования. Только одно из многочисленных устройств (процессор, магнитный диск, принтер, магнитная лента) работало в конкретный момент времени.

Мультипрограммные ОС позволяют загрузить множество устройств вычислительной системы за счёт одновременного выполнения нескольких заданий, что приводит к сокращению суммарного времени выполнения смеси заданий.

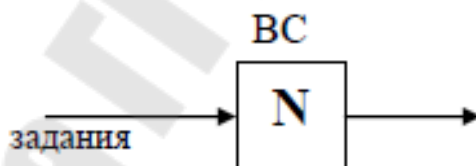


Рисунок 2.2 – Многопрограммный режим работы

Число N называют коэффициентом мультипрограммирования. Сокращение суммарного времени выполнения достигается за счёт совмещения работы процессора с работой внешних устройств. В то время как Процесс2 выполняется на процессоре, Процесс1 может обслуживаться на магнитном диске, как показано на рисунке 2.3.

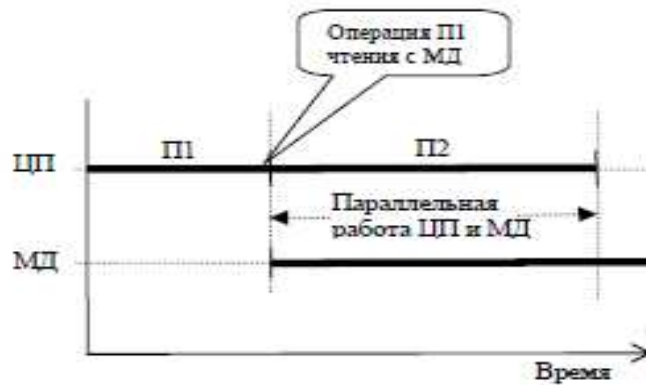


Рисунок 2.3 – Совмещении работы процессора с работой внешних устройств

Следует отметить, что мультипрограммный режим работы требует соответствующей аппаратной поддержки. Оборудование должно обеспечивать автономную работу внешних устройств, управляемых своим контроллером (каналом), а также систему прерываний для сигнализации внешних устройств о завершении операции.

Первые мультипрограммные ОС функционировали в *пакетном режиме*, в котором пользователь отделялся от процесса выполнения его заданий. Предварительно подготовленные задания организовывались в очереди на перфокартах либо магнитном диске. Затем смесь заданий подавалась на вход вычислительной системы:

Необходимость взаимодействия пользователя с выполняющимися программами, автоматизация процессов подготовки заданий с помощью текстовых редакторов, а также появление дисплеев с электронно-лучевой трубкой обусловили появление ВС с *диалоговым режимом* работы. В этом режиме несколько пользователей одновременно взаимодействуют с ВС с помощью терминалов.

Для обеспечения работы нескольких пользователей в диалоговом режиме особенно важным становится *время реакции ОС* на команды пользователя. При использовании классического мультипрограммирования один из процессов может захватить процессор на неопределённое время и, таким образом, заблокировать работу остальных пользователей. Эффективный диалоговый режим возможен при гарантированном времени реакции ОС не превышающем несколько секунд.

Для обеспечения диалоговой работы был предложен *режим разделения времени*, обеспечивающий совместную работу нескольких

пользователей. Основой для обеспечения режима разделения времени является *циклическое квантование времени* центрального процессора. Каждому процессу при получении процессора выделяется определённый интервал времени – *квант*, в течение которого он выполняется на процессоре; затем процесс возвращается в конец очереди готовых процессов. Таким образом, процессор циклически переключается между готовыми процессами и каждый из пользователей считает, что ОС работает только с ним одним; далее представлен фрагмент диаграммы состояний процесса, обеспечивающий циклическое квантование.

Следует отметить, что организация квантования влечёт за собой некоторое снижение производительности ВС. Переключение между процессами требует определённого времени, в течение которого работают программы ОС, и представляет собой *накладные расходы* ресурсов ОС при организации режима разделения времени. При реализации циклического квантования встаёт вопрос о выборе *оптимального размера кванта*. Большой размер кванта снижает время реакции ОС, малый – увеличивает накладные расходы из-за частого переключения. Реальные ОС применяют адаптивные механизмы выбора размера кванта .

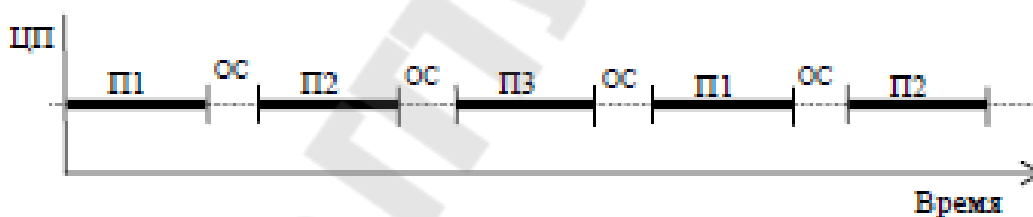


Рисунок 2.4 – Адаптивный механизм выбора размера кванта

Для реализации квантования времени используется аппаратный таймер. Таймер устанавливается на продолжительность кванта при выделении процессора процессу; прерывание таймера по истечении времени кванта инициирует переключение.

Задание

I Выполнить ручную трассировку работы средств управления процессами мультипрограммной ОС. Оценить эффективность работы мультипрограммной ОС.

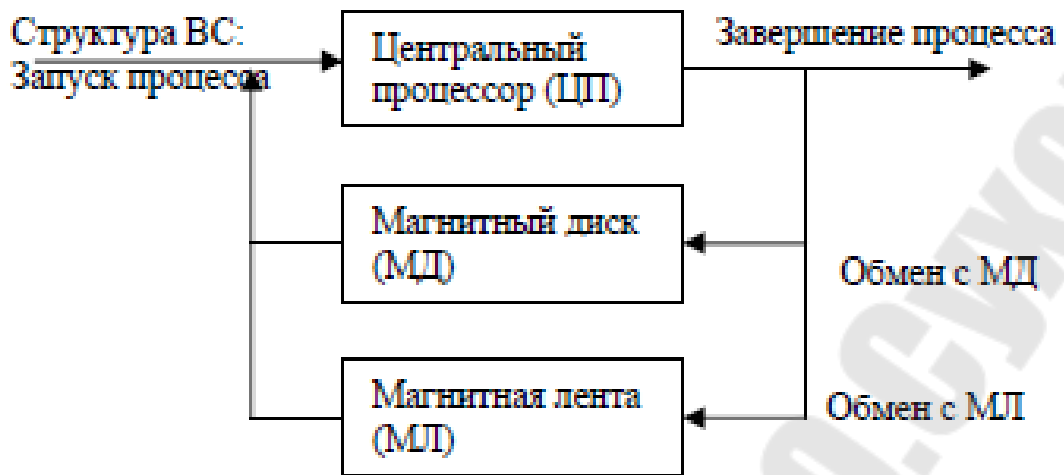


Рисунок 2.5 – Структура вычислительной системы

Структура процессов:

31: С1ЦП-D1МД-C2ЦП

32: С3ЦП-L1МЛ-C4ЦП

33: С5ЦП-D2МД-C6ЦП-D3МД-C7ЦП 34: С3ЦП-L2МЛ-C4ЦП-D2МД-C1ЦП

Варианты заданий

$$C_i = (((n+i) \bmod 5) + 1) * 10$$

$$D_i = (((n+i) \bmod 4) + 1) * 100$$

$$L_i = (((n+i) \bmod 3) + 1) * 1000$$

n – номер слушателя в журнале

33: 50ЦП-400МД-10ЦП-100МД-20ЦП 34: 30ЦП-2000МЛ-40ЦП-400МД-10ЦП

31: 10ЦП-300МД-20ЦП 32: 30ЦП-1000МЛ-40ЦП

Таблица 2.1 – Исходные данные для варианта №29

№	C	D	L
1	10	300	1000
2	20	400	2000
3	30	100	
4	40		
5	50		
6	10		
7	20		

II Выполнить ручную трассировку работы средств управления процессами. Заполнить трассировочную таблицу.

Характеристики ОС: циклическое квантование времени, мультипрограммирование

Варианты заданий

$$C_i = (((n+i) \bmod 5) + 1) * 10$$

$$D_i = (((n+i) \bmod 4) + 1) * 100$$

$$L_i = (((n+i) \bmod 3) + 1) * 1000$$

Размер кванта:

$$dt = ((n \bmod 2) + 1) * 10$$

Время переключения:

$$tos = ((n \bmod 4) + 2)$$

n – номер слушателя в журнале.

Пояснения к выполнению работы

Задание I

1. Выполнить ручную трассировку выполнения указанной смеси процессов:

а) в однопрограммной вычислительной системе;

б) в мультипрограммной вычислительной системе.

2. Заполнить трассировочные таблицы.

3. Выполнить анализ эффективности мультипрограммного режима:

3.1. Оценить ускорение выполнения смеси процессов;

3.2. Рассчитать и сравнить коэффициенты загрузки устройств.

4. Сформулировать преимущества мультипрограммного режима.

Пример выполнения

Таблица 2.2 – Характеристики выполняемых процессов

Номер	Время поступления	Последовательность действий
1	0	ЦП(40)-МД(100)-ЦП(20)
2	10	ЦП(20)-МД(100)-ЦП(80)
3	20	ЦП(60)-МД(100)-ЦП(10)

Таблица 2.3 – Пример трассировки

Время	Очередь готовых процессов	Процессор	Очередь к МД	МД
0		П1(40)		
10	П2(20)	П1(30)		
20	П3(20), П2(20)	П1(20)		
40	П3(20)	П2(20)		П1(100)
60		П3(60)	П2(100)	П1(80)
120			П3(100), 2(100)	П1(20)
140		П1(20)	П3(100)	П2(100)
160			П3(100)	П2(80)
240		П2(80)		П3(100)
320				П3(20)
340		П3(10)		
350				

Ускорение выполнения смеси процессов

Время выполнения смеси в однопрограммном режиме:

$$T_1 = (40 + 100 + 20) + (20 + 100 + 80) + (60 + 100 + 10) = 530$$

Время выполнения смеси в мультипрограммном режиме:

$$T_m = 350$$

$$\text{Ускорение } A = T_1 / T_m = 530 / 350 = 1,51$$

Коэффициенты загрузки устройств

$$K = T_{\text{устр}} / T_{\text{общ}}$$

Таблица 2.4 – Расчет коэффициентов загрузки устройств

Однопрограммный	Мультипрограммный
$K_{\text{проц}} = 230 / 530 = 0,45$	$K_{\text{проц}} = 230 / 350 = 0,66$
$K_{\text{мд}} = 300 / 530 = 0,57$	$K_{\text{мд}} = 300 / 350 = 0,86$

Задание II

1. Выполнить ручную трассировку выполнения указанной смеси процессов.
2. Заполнить трассировочные таблицы.
3. Выполнить анализ эффективности режима разделения времени:
 - а) при мгновенном переключении процессов;
 - б) при времени переключения равном 1.
4. Сформулировать преимущества и недостатки режима

разделения времени.

Пример выполнения

Размер кванта – 10.

Таблица 2.5 – Характеристики выполняемых процессов

Номер	Время поступления	Последовательность действий
1	0	ЦП(40)-МД(100)-ЦП(20)
2	10	ЦП(20)-МД(100)-ЦП(80)
3	20	ЦП(60)-МД(100)-ЦП(10)

Таблица 2.6 – Пример заполнения трассировочной таблицы

Время	Очередь готовых процессов	Процессор	Очередь к МД	МД
0		П1(40)		
10	П1(30)	П2(20)		
20	П2(10), П1(30)	П3(60)		
30	П3(50),П2(10)	П1(30)		
40	П1(20),П3(50)	П2(10)		
50	П1(20)	П3(50)		П2(100)
60	П3(40)	П1(20)		П2(90)
70	П1(10)	П3(40)		П2(80)
80	П3(40)	П1(10)		П2(70)
90		П3(40)	П1(100)	П2(60)
100

Контрольные вопросы

1. В чём состоит мультипрограммный режим работы ОС?
2. За счёт чего сокращается время выполнения смеси процессов в мультипрограммном режиме?
3. В чём состоит режим разделения времени?
4. Как влияет режим разделения времени на производительность ВС?
5. Каковы требования к аппаратным средствам компьютера для организации квантования?
6. Какие критерии используют при выборе оптимального размера кванта?

Лабораторная работа № 3

Тема: Управление памятью

Цель работы: освоить основы организации управления памяти динамическими разделами, оценить эффективность управления памяти динамическими разделами

Теоретические сведения

Оперативная память (ОП) является вторым после ЦП ресурсом по своей значимости для исполнения процесса. Мно-жество способов управления памятью конкретных ОС можно классифицировать по следующим основным признакам:

- полное либо частичное размещение процесса в ОП;
- связанное либо несвязанное выделение оперативной памяти;
- выделение участков памяти фиксированной либо переменной длины.

Многие сложные методы управления ОП, обуслов-ленные исторически высокой стоимостью и ограничениями объёма физической памяти, например, оверлейные структуры, практически не используются в настоящее время в связи с развитием концепции виртуальной памяти. Стандартом де-факто для современных ОС является виртуальная сегментно-страничная память.

Управление оперативной памятью (ОП) *динамическими разделами* применялось в ОС ранних поколений; однако и в настоящее время этот способ управления памятью остаётся актуальным, так как используется для управлением пулом памяти базы данных ОС, в котором размещаются управляющие блоки ОС: блок управления процессом, блок управления устройством, блок управления файлом и другие .

В ранних поколениях ОС оперативная память рассматривалась как одномерный массив байтов (слов). Для выделения процессу ЦП требовалось выполнение двух условий: полное размещение процесса (исполнимого файла) в оперативной памяти; связанное выделение памяти процесса в виде одного непрерывного участка. В этом случае текущее распределение памяти полностью описывается массивом границ (начальный и конечный адреса либо начальный адрес и длина) памяти процессов. Аналогичный описатель можно использовать и для оставшихся свободных участков ОП.

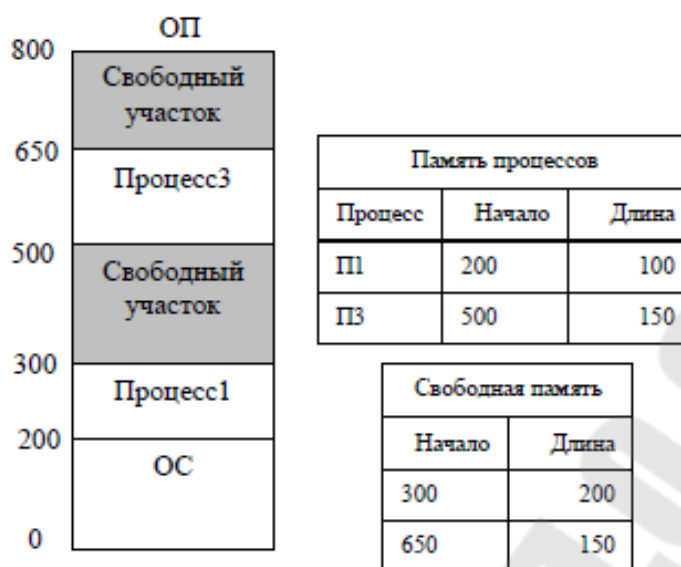


Рисунок 3.1 – Схема размещения процессов в оперативной памяти в ранних операционных системах

Первоначально загруженные процессы занимают один непрерывный участок памяти вслед за кодом ОС.

Однако ввиду случайного времени их завершения, свободные и занятые участки становятся разбросанными в массиве байтов ОП.



Рисунок 3.2 – Схема размещения процессов в оперативной памяти в современных операционных системах

Такую ситуацию называют *фрагментацией памяти*. При выборе требуемого участка памяти для нового процесса ОС может применять критерии первого подходящего участка либо наиболее подходящего участка. Однако возможна ситуация, при которой

общий объем свободной памяти удовлетворяет требованиям процесса, но составлен суммой размеров несмежных свободных участков. В этом случае был предложен специальный метод реорганизации ОП, называемый *перемещением*. Запускается специальный модуль управления памяти ОС, который сдвигает (копирует) все процессы в ОП для устранения фрагментации и образования одного непрерывного участка свободной ОП:



Рисунок 3.3 – Дефрагментация оперативной памяти

Следует отметить, что перемещение неизбежно влечёт за собой накладные расходы ОС в виде дополнительного времени копирования процессов в ОП. Хотя архитектура многих процессоров предлагает специальные операции быстрого копирования блоков памяти, эти расходы могут быть существенными для общей производительности ВС.

Мультипрограммный режим делает актуальными аппаратные средства защиты памяти для предотвращения (ошибочного либо намеренного) изменения кодов одного процесса некоторым другим процессом. Заметим, что средства защиты памяти крайне необходимы и однопрограммным ОС для защиты раздела памяти самой ОС от вмешательства исполняемого процесса. В большинстве ОС с управлением памятью динамическими разделами эта проблема решается путём использования *аппаратных регистров границ* текущего процесса. При обращении к адресам памяти вне границ генерируется аппаратное прерывание по защите памяти, которое затем обрабатывается ОС.

Задание

Выполнить ручную трассировку средств управления оперативной памятью. Заполнить трассировочную таблицу. Оценить эффективность управления памятью.

Характеристики ОС: динамические разделы, мультипрограммирование, приоритетная дисциплина.

Варианты заданий

Размер ОП: $V = ((n \bmod 4) + 7) * 10$

n – номер слушателя в журнале.

Таблица 3.1 – Характеристики выполняемых процессов

Номер (i)	Время поступления (t_i)	Время выполнения	Размер в ОП	Приоритет
1	0	$((n+i) \bmod 4)+2) * 10$	$((n+i) \bmod 3) + 3) * 10$	$((n+i) \bmod 3) + 1$
2	$t_{i-1} + ((n+i) \bmod 2) + 1) * 10$
3
4

Пояснения к выполнению работы

1. Выполнить ручную трассировку работы средств управления ОП.
2. Заполнить трассировочные таблицы.
3. Выполнить анализ накладных расходов на перемещение процессов.
4. Сформулировать преимущества и недостатки управления памяти динамическими разделами.

Пример выполнения

Размер ОП – 70

Таблица 3.2 – Характеристики выполняемых процессов

Номер	Время поступления	Время выполнения	Размер в ОП	Приоритет
1	0	30	30	2
2	20	40	20	1
3	30	50	40	3

Таблица 3.3 – Пример заполнения трассировочной таблицы

Время	Состояние ОП				Примечание
0	Процесс	П1 (акт.)		Своб.	
	Начало	0		30	
	Длина	30		40	
20	Процесс	П1 (акт.)	П2	Своб.	
	Начало	0	30	50	
	Длина	30	20	20	
Время	Состояние ОП				Примечание
30	Процесс	Своб.	П2	Своб.	Фрагментация
	Начало	0	30	50	
	Длина	30	20	20	
30	Процесс	П2		Своб.	Перемещение
	Начало	0		20	
	Длина	20		50	
30	Процесс	П2	П3 (акт.)	Своб.	
	Начало	0	20	60	
	Длина	20	40	10	
80

Контрольные вопросы

1. Каким образом описывается распределение памяти при использовании динамических разделов?
2. Почему возникает фрагментация памяти при использовании динамических разделов?
3. Каким образом ликвидируется фрагментация памяти?
4. Какие аппаратные средства применяются для защиты оперативной памяти?

Лабораторная работа № 4

Тема: Файловая структура диска

Цель работы: освоить основы организации управления информацией, выполнить сравнительную оценку различных способов организации файловой структуры диска

Теоретические сведения

Древовидная (иерархическая) файловая система, образованная такими элементами как *том*, *каталог*, *файл*, является стандартной для современных ОС. Файл представляет собой поименованную единицу хранения информации; файлы объединяются в каталоги, причём каталог может содержать как файлы, так и другие каталоги; том представляет собой устройство с файловой системой. Единственная разница между семействами доминирующих в настоящее время ОС Unix и MS Windows состоит в представлении томов: Unix использует общую иерархию всех томов, MS Windows представляет файловую систему с разбивкой по томам (устройствам):

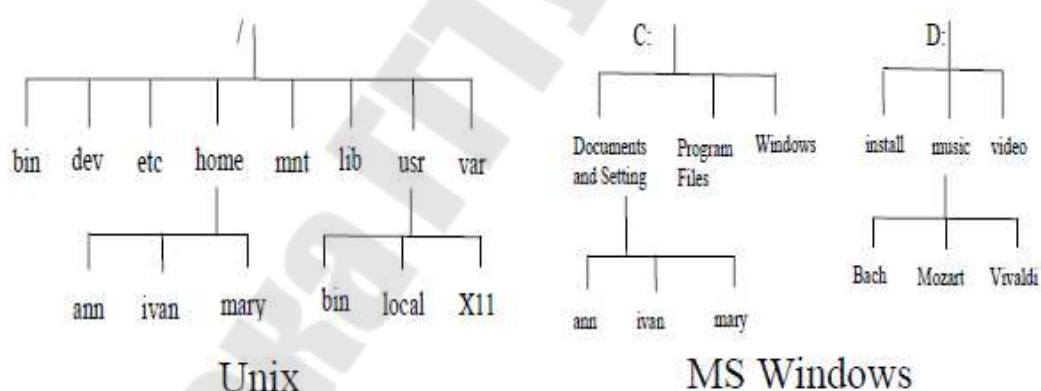


Рисунок 4.1 – Иерархия данных ОС Unix и Windows

Как правило, современные ОС реализуют такие основные файловые операции как чтение-запись и позиционирование внутри файла, обеспечивая *последовательный и прямой доступ* к информации. Более сложные методы доступа реализуются СУБД на основе указанных операций ОС.

Логическая структура диска может быть представлена одномерным массивом секторов образованным занумерованными физическими секторами всех поверхностей, цилиндров и дорожек.

Возникает *проблема отображения* файловой системы на логическую структуру диска для обеспечения доступа к файлам.

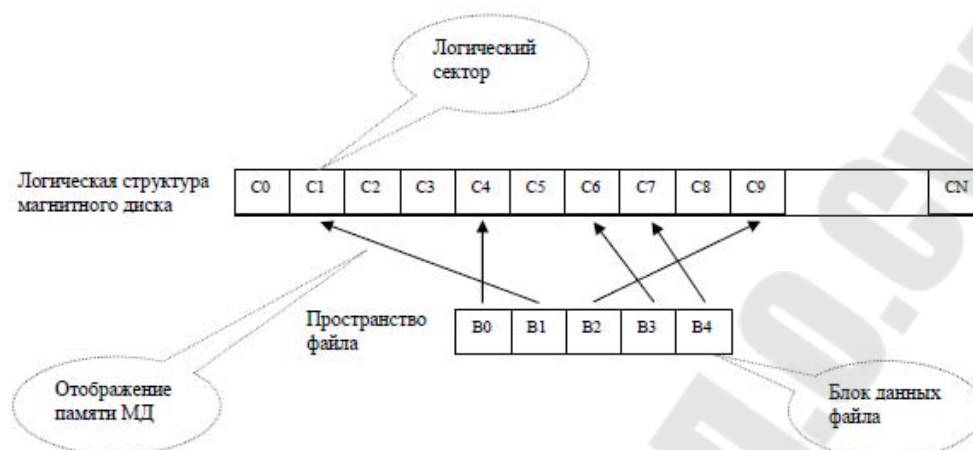


Рисунок 4.2 – Организация доступа к файлам

Для обеспечения отображения создаются специальные структуры данных, описывающие отображение и невидимые для конечного пользователя. Способ организации таких данных принято называть *файловой структурой*. Файловая структура обеспечивает решение двух основных задач: нахождение логических секторов заданного файла/каталога на диске; учёт свободных/занятых секторов.

Большинство файловых структур использует специальный блок – *дескриптор* для каждого из своих объектов: том, каталог, файл. Дескриптор файла/каталога содержит имя владельца, права доступа, даты создания/корректировки, длину. Дескриптор тома содержит также указатель на дескриптор корневого каталога. Дескрипторы файла/каталога имеют фиксированную длину. ОС используют два основных подхода к размещению дескрипторов: в ОС семейства Unix дескрипторы файлов/каталогов размещаются в общем одномерном массиве – *индексе*, в этом случае каталог содержит список указателей на дескрипторы его файлов/каталогов; в ОС семейства MS Windows дескрипторы файлов/каталогов размещаются внутри содержащего их каталога

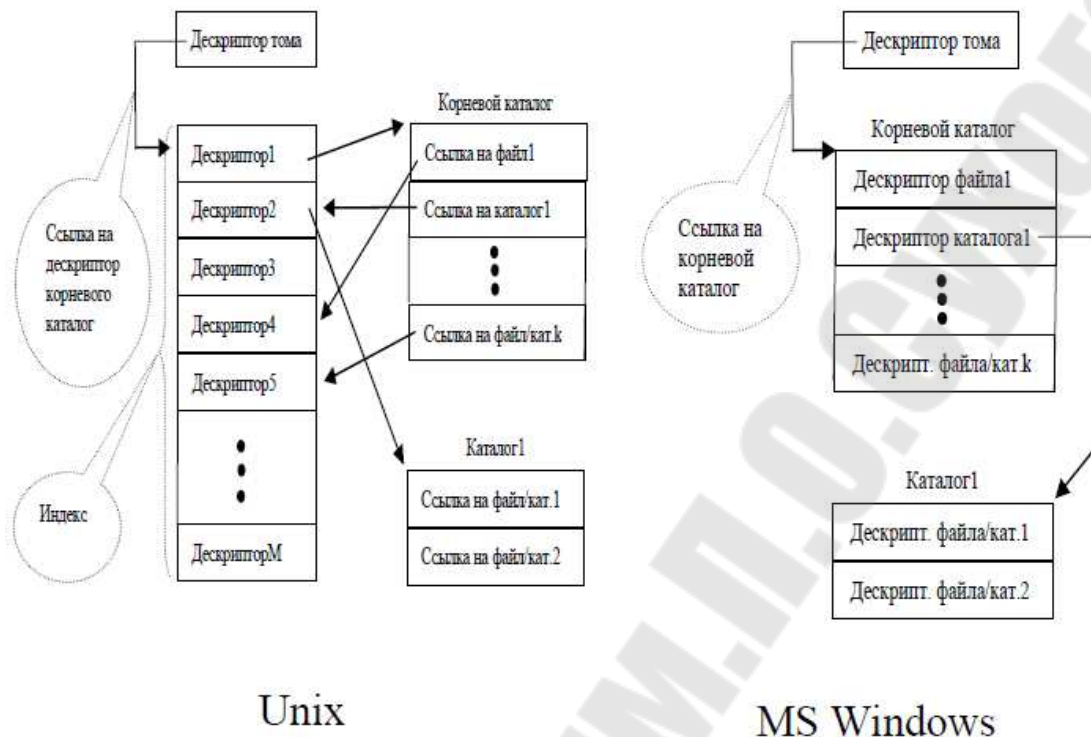


Рисунок 4.3 – Организация дескрипторов файлов в ОС

При решении задачи отображения для уменьшения размера структур данных применяют *кластеризацию*. Кластер состоит из нескольких блоков (секторов) и является единицей выделения пространства диска. В настоящее время доминируют два подхода к описанию размещения файла на диске: использование общей таблицы размещения для всех кластеров (в ОС семейства MS Windows); использование массива описателей связанных участков в дескрипторе файла (в ОС семейства Unix).

Общая таблица размещения содержит по одной записи для каждого кластера. Запись равняется нулю, если соответствующий кластер свободен. Если кластер принадлежит некоторому файлу, то соответствующая запись содержит номер следующего кластера; запись последнего кластера файла имеет специальное значение – признак конца цепочки (например, все двоичные единицы).

Использование *таблицы связанных участков*, описанных указанием начального кластера участка и длины в дескрипторе файла, обеспечивает более быстрое отображение. Однако существенным недостатком этого метода является фиксированное число участков в дескрипторе файла. Размещение больших фрагментированных файлов решается в ОС Unix за счёт выделения дополнительных дескрипторов.

0	1	2	3	4	5	6	7	8	9
0	0	3	4	8	0	-	0	9	6

Рисунок 4.4 – Таблица размещения файлов (File Allocation Table)

Нач	2	8	6
Дл.	3	2	1

Рисунок 4.5 – Таблица связанных участков (в дескрипторе)

Использование единой таблицы размещения обеспечивает также решение задачи *учёта свободных/занятых кластеров*: записи свободных кластеров имеют нулевое значение в таблице. К дополнительным способам учёта свободного/занятого пространства можно отнести битовые карты и специальный (фиктивный) файл, занимающий все свободные кластеры. В битовой карте каждый бит соответствует кластеру; значение 0 – кластер свободен, значение 1 – занят. Применение специального файла обеспечивает преимущества быстрого поиска наиболее подходящего участка при выделении пространства и способствует уменьшению фрагментации.

Задание

Выполнить ручную трассировку работы файловой системы. Заполнить трассировочную таблицу.

I Характеристики ОС: несвязное выделение дискового пространства, таблица связанных участков в дескрипторе файла, специальный файл учёта свободного пространства, мультипрограммирование.

II Характеристики ОС: несвязное выделение дискового пространства, отдельная таблица размещения файлов, учёт свободного пространства в таблице размещения, мультипрограммирование

Последовательность операций:

Создать файл 1. Записать $((n \bmod 2) + 2)$ блока в файл 1. Создать файл 2. Записать $((n \bmod 2) + 1)$ блока в файл 2. Допisać $((n \bmod 2) + 1)$ блока в файл 1. Допisać $((n \bmod 3) + 1)$ блока в файл 2. Удалить файл 1.

Доступное пространство диска: $V = (n \bmod 3) + 10$
 n – номер слушателя в журнале.

Пояснения к выполнению работы

1. Выполнить ручную трассировку работы сетевых средств ОС.
2. Заполнить трассировочные таблицы.
3. Оценить дополнительный объем информации заголовков пакетов.
4. Сформулировать основные принципы адресации процессов в сети.

Пример выполнения

Последовательность операций:

Создать файл 1. Записать 3 блока в файл 1.

Создать файл 2. Записать 2 блока в файл 2.

Дописать 1 блок в файл 1.

Дописать 2 блока в файл 2.

Удалить файл 1.

Доступное пространство диска – 10.

Задание I

Таблица 4.1 – Пример заполнения трассировочной таблицы

Операция	Дескрипторы файлов		
	Файл 0	Файл 1	Файл 2
	A, (0,10)		
Создать файл 1	A, (0,10)	F1	
Записать 3 блока в файл 1	A, (3,7)	F1, (0,3)	
Создать файл 2	A, (3,7)	F1, (0,3)	F2
Записать 2 блока в файл 2	A, (5,5)	F1, (0,3)	F2, (3,2)
Дописать 1 блок в файл 1	A, (6,4)	F1, (0,3),(5,1)	F2, (3,2)
...

Задание II

Таблица 4.2 – Пример заполнения трассировочной таблицы

Операция	Код операции
Создать файл 1	O1
Записать 3 блока в файл 1	O2
Создать файл 2	O3
Записать 2 блока в файл 2	O4
Дописать 1 блок в файл 1	O5

Таблица 4.3 – Пример заполнения трассировочной таблицы

Код операции	Дескрипторы файлов		
	Файл 0	Файл 1	Файл 2
	A, (0,10)		
O1	A, (0,10)	F1	
O2	A, (3,7)	F1, (0,3)	
O3	A, (3,7)	F1, (0,3)	F2
O4	A, (5,5)	F1, (0,3)	F2, (3,2)
O5	A, (6,4)	F1, (0,3),(5,1)	F2, (3,2)
...

Таблица 4.4 – Пример заполнения трассировочной таблицы

Код операции	Дескрипторы файлов			Таблица размещения файлов							
	0	1	2	0	1	2	3	4	5	6	7
				0	0	0	0	0	0	0	0
O1	F1										
O2	F1, 0			1	2	-	0	0	0	0	0
O3	F1, 0	F2		1	2	-	0	0	0	0	0
O4	F1, 0	F2, 3		1	2	-	4	-	0	0	0
O5	F1, 0	F2, 3		1	2	5	4	-	-	0	0
...

Контрольные вопросы

1. Каковы основные элементы файловой системы?
2. Для чего необходима специальная файловая структура диска?
3. Какую информацию содержит дескриптор файла (тома, каталога)?
4. Как организуются дескрипторы для описания древовидной файловой системы?
5. Что такое кластер?
6. Каким образом описывается фактическое размещение файла на диске?
7. Какие средства используются для учёта свободного/занятого пространства диска?
8. Каким образом обеспечивается защита файловой системы?

Список использованных источников

1. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. – Питер, 2001. – 544 с.
2. Дейтел Г. Введение в операционные системы: В 2-х т. – М.: Мир, 1987. – 756 с.
3. Краковяк С. Основы организации и функционирования ЭВМ. – М.: Мир, 1988. – 480 с.
4. Мэдник С., Донован Дж. Операционные системы. – М.: Мир, 1978. – 640с.
5. Иртегов Д. Введение в операционные системы. – ВНУ: Санкт-Петербург, 2002. – 624 с.
6. Столлингс В., Операционные системы. – Вильямс, 2002.– 848 с.
7. Партыка Т.Л., Попов И.И. Операционные системы, среды и оболочки. – Форум, 2003. – 400 с.
8. Гордеев А.В. Молчанов А.Ю Системное программное обеспечение. СПб Питер, 2001.
9. Робачевский А.М. ОС UNIX. ВНУ: С-Петербург, 2001.
10. Шоу А. Логическое проектирование операционных систем. – М.: Мир, 1981. – 360 с.
11. Соловьев Г.Н. Операционные системы ЦВМ. – М.: Машиностроение, 1977. – 136 с.
12. Татцан Г. Операционные системы.- М.: Мир, 1976.- 471с.
13. Лихачева Г.Н., Медведев В.Д. Операционные системы. – М.: Статистика, 1980. – 231с.
14. Зелковиц М., Шоу А., Гэннон Дж. Принципы разработки программного обеспечения. – М.: Мир, 1982. – 368 с.
15. Коэн Л.Дж. Анализ и разработка операционных систем. – М.: Наука, 1975. – 190 с.

Гридина Елена Ивановна

**АРХИТЕКТУРА
ОПЕРАЦИОННЫХ СИСТЕМ**

ПРАКТИКУМ

**по одноименной дисциплине
для слушателей специальности 1-40 01 73
«Программное обеспечение
информационных систем»
заочной формы обучения**

Подписано к размещению в электронную библиотеку
ГГТУ им. П. О. Сухого в качестве электронного
учебно-методического документа 27.09.16.

Рег. № 88Е.

<http://www.gstu.by>