Федеральное агентство по образованию РФ
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ГОУВПО «АмГУ»)

УТВЕРЖДАЮ
Зав. кафедрой ИУС
_____А.В.Бушманов
«_____»_____

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

по дисциплине «Операционные системы»

для студентов специальности 230201 – Информационные системы и технологии

Составитель: доцент кафедры ИУС Галаган Т.А.

Факультет математики и информатики

Кафедра информационных и управляющих систем

2007

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

Т.А. Галаган

Учебно-методический комплекс по дисциплине «Операционные системы». Для студентов специальности 230201 «Информационные системы и технологии» очной формы обучения.- Благовещенск: Амурский гос. ун-т, 2007.

Пособие содержит рабочую программу, курс лекций, методические рекомендации по проведению и выполнению лабораторных работ. Составлено в соответствии с требования государственного образовательного стандарта.

© Амурский государственный университет, 2007

I. ПРИМЕРНАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ, УТВЕРЖДЕННАЯ МИНОБРАЗОВАНИЯ РФ

Государственный образовательный стандарт высшего профессионального образования

Направление подготовки дипломированного специалиста 654700 - информационные системы и технологии

Образовательная программа – 230201 Информационные системы и технологии

наименование дисциплины – Операционные системы

блок общеобразовательных дисциплин ОПД.Ф.11

Всего часов - 102

Содержание разделов:

Принципы построения ОС, вычислительный процесс и его реализации с помощью ОС, основные функции ОС, обзор современных ОС и операционных оболочек, стандартные сервисные программы, машинно-зависимые свойства ОС, управление вычислительными процессами, вводом-выводом, реальной памятью; управлению виртуальной памятью; машинно-независимые свойства ОС; способы планирования заданий пользователей; динамические, последовательные и параллельные структуры программ; способы построения ОС; сохранность и защита программных систем; интерфейсы и основные стандарты в области системного программного обеспечения.

II. РАБОЧАЯ ПРОГРАММА

Курс 2 Семестр 3

Лекции 36 (час.) Экзамен 3 семестр

Лабораторные работы 18 (час.)

Самостоятельная работа 48 (час.)

Всего часов 102 час.

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1.Цель курса – изучение принципов построения, назначения, теоретических основ функционирования и практического использования операционных систем как эффективного средства управления процессами обработки данных в современных ЭВМ.

1.2. По завершению курса «Операционные системы» студент должен:

- владеть такими понятиями, как вычислительный процесс, мультипрограммирование, файловая система, и их реализация с помощью операционной системы;
- знать назначение и функции операционной системы, современное состояние теории операционных систем и методы, используемые при их разработке;
- знать теоретические основы планирования процессов и потоков и управление памятью;
- иметь устойчивые практические навыки работы с операционными системами MS DOS, Windows98/NT/2000/XP;
- уметь создавать программы, расширяющие возможности операционных систем.

1.3. Преподавание курса «Операционные системы» связано с изучением курса государственного образовательного стандарта «Информатика» и является основой для изучения дальнейших дисциплин, использующих ЭВМ и программирование.

2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. ФЕДЕРАЛЬНЫЙ КОМПОНЕНТ

Программа курса «Операционные системы» составлена в соответствие с требованиями государственного образовательного стандарта специализации – Интегрированные системы автоматизированного управления, специализации 230201, блок общеобразовательных дисциплин ОПД.Ф.11

2.2. ЛЕКЦИИ (36 часов)

2.2.1. Эволюция операционных систем (2 часа)

2.2.2. Принципы построения современных операционных систем (4 часа): назначение, основные функции; классификация операционных систем.

2.2.3. Архитектура операционной системы (4 часа):
ядро и вспомогательные модули операционной системы; многослойная структура операционной системы; микроядерная архитектура операционной системы; машинно-зависимые и машинно-независимые свойства, способы построения.

2.2.4. Управление вычислительными процессами (6 часов):
мультипрограммирование; понятие процесса и потока; алгоритмы планирования процессов на основе квантования и приоритетов; мультипрограммирование на основе прерываний.

2.2.5. Управление памятью (6 часов):
функции операционной системы по управлению памятью; алгоритмы распределения памяти; свопинг и виртуальная память; кэширование данных.

2.2.6. Управление вводом-выводом (8 часов):
понятие файловой системы, ее основные свойства; реализации файловой системы; дескриптор файла; физическая организация файловой системы, файловые системы FAT, NTFS.

2.2.7. Сохранность программных систем, методы защиты от сбоев и несанкционированного доступа (2 часа)

2.2.8. Обзор современных операционных систем, операционных оболочек, интерфейсы и основные стандарты в области системного программного обеспечения (4 часа): краткая характеристика и сравнительный анализ операционных систем: MS DOS, Windows, Windows NT, UNIX, LINUX.

2.3 ПРАКТИЧЕСКИЕ ЗАНЯТИЯ –

2.4. ЛАБОРАТОРНЫЕ РАБОТЫ (18 часов)

2.4.1. Создание файлов конфигураций операционной системы MS DOS (4 часа)

2.4.2: Создание командных файлов операционной системы MS DOS (2 часа)

2.4.3. Загрузка операционной системы WINDOWS98/2000/XP. Структура оперативной памяти. Распределение ресурсов компьютера(4 часа)

2.4.4. Работа с дисками и дисководами. (2 часа)

2.4.5. Служебные программы WINDOWS98/2000/XP. (2 часа)

2.4.6. Антивирусные программные средства (2 часа)

2.4.7. Архивация и кодирование данных (2 часа)

2.5. КУРСОВАЯ РАБОТА –

2.6. САМОСТОЯТЕЛЬНАЯ РАБОТА (48 часов):

2.6.1. Знакомство с публикациями в периодических изданиях

2.6.2. Поиск информации в Internet www:/amursu.ru/citforum

2.6.3. Физическая организация файловых систем FAT16, FAT32, NTFS

2.6.4. Динамические, последовательные и параллельные структуры программ

Контроль за выполнением самостоятельной работы в экзаменационных вопросах.

2.7. ПЕРЕЧЕНЬ И ТЕМЫ ПРОМЕЖУТОЧНЫХ ФОРМ КОНТРОЛЯ ЗНАНИЙ

2.7.1. Вариант тестовых заданий к темам 2.2.2 – 2.2.5

1. Выберите верные утверждения:

Операционная система

a) загружается автоматически при запуске компьютера.

b) это несвязный набор программных средств, предназначенных для управления загрузкой, запуском и выполнением других программ.

c) это комплекс взаимосвязанных программ, неоднородный по характеру и многоплановый по уровню.

d) комплекс взаимосвязанных программ, из которого нельзя удалять и добавлять части.

2. Источниками прерываний могут быть

a) аппаратура и программы b) только аппаратура c) только программы

3. *Утилиты - это*

a) специальные варианты пользовательского интерфейса

b) программы, решающие отдельные задачи управления и сопровождения компьютерной системы, программы сжатия дисков, архивирования данных

c) библиотеки процедур различного назначения, упрощающие разработку приложений

4. К ядру операционной системы относятся

a) библиотеки процедур различного назначения, упрощающие разработку приложений

b) текстовые и графические редакторы

c) функции, решающие внутрисистемные задачи, такие как переключение контекстов, обработка прерываний

d) отладчики, компиляторы, компоновщики

5. Восстановите последовательность функционирования компьютера после включения питания

a) инициализация основных аппаратных блоков компьютера и регистров процессора
b) загрузка ядра операционной системы
c) запуск программы первоначальной загрузки

6. К сетевым средствам операционной системы относятся:

a) распределение оперативной памяти
b) планирование и диспетчеризация процессов
c) средства предоставление локальных ресурсов в общее пользование
d) средства запроса к удаленным данным
e) управление внешней памятью
f) транспортные средства, обеспечивающие передачу сообщений в сети

7. Поставьте соответствия между свойствами операционных систем и их определениями

1) расширяемость	a) код системы позволяет изменения, связанные с поддержкой новых типов устройств или технологий
2) переносимость	b) наличие средств поддержки для выполнения программ, написанных для других операционных систем
3) совместимость	c) наличие нескольких вариантов реализаций для разных платформ

8. В оперативной памяти хранятся постоянно

a) утилиты, системные обрабатывающие программы и ядро операционной системы
b) только ядро операционной системы
c) утилиты, системные обрабатывающие программы

9. Поставьте соответствие между слоями ядра ОС и их функциями

1) менеджеры ресурсов
2) интерфейс системных вызовов
3) базовые механизмы ядра
4) машинно-зависимые компоненты
5) средства аппаратной поддержки ОС

a) программные модули, в которых отражается специфика аппаратной платформы компьютера
b) средства, непосредственно участвующие в организации вычислительных процессов

(система прерываний средства переключения процессов, защиты областей памяти)
c) примитивные операции ядра, исполняющие переключение контекстов процессов, диспетчеризацию прерываний, перемещение страниц из памяти на диск.
d) мощные функциональные модули, реализующие стратегические задачи по управлению ресурсами системы
e) функции предоставления доступа к системным ресурсам

10. Информация о процессе, необходимая ядру операционной системы в течение всего его жизненного цикла, содержится в
a) контексте процесса b) в дескрипторе процесса

11. Выберите все правильные положения, используемые в алгоритмах планирования, основанных на квантовании
a) кванты, выделяемые потокам, могут быть одинаковы для всех потоков или различными
b) кванты, выделяемые потокам, должны быть одинаковы для всех потоков
c) кванты, выделяемые одному потоку, фиксированной величины
d) кванты, выделяемые одному потоку, могут быть фиксированной величины, а могут изменяться в периоды жизни потока
e) потоки, получая квант времени, всегда используют его полностью

12. Приоритеты делятся на:
a) назначаемые и неназначаемые
b) изменяемые и неизменяемые
c) вытесняемые и невытесняемые

13. В системах с относительными приоритетами выполнение активного потока продолжается (укажите все возможные варианты):
a) пока он сам не покинет процессор, перейдя в состояние ожидания
a) пока не произойдет ошибка
b) пока поток не завершится
c) пока в очереди готовых процессов не появится поток, приоритет которого выше приоритета активного потока

2.7.2. Вариант тестовых заданий к темам 2.2.6 – 2.2.7
1. Выберите из перечисленных методы распределения памяти без использования внешней памяти
a) распределения памяти фиксированными разделами
b) страничное распределение
c) распределения памяти динамическими разделами
d) сегментное распределение
e) распределения памяти перемещаемыми разделами
f) странично-сегментное распределение

2. Выберите правильные утверждения при распределении памяти динамическими разделами
a) память заранее не делится на разделы фиксированного размера
b) если достаточный свободный раздел памяти, для помещения всего процесса, отсутствует процесс все же создается
c) существенный недостаток данного метода - дефрагментация

3. Сжатие данных, выполняемое либо при завершение процесса, либо когда вновь созданному процессу недостаточно места, присуще методу
a) с фиксированными разделами
b) с динамическими разделами
c) с перемещаемыми разделами

4. При свопинге
a) между оперативной памятью и диском перемещаются части (сегменты, страницы) образов процессов
b) образы виртуальных процессов выгружаются на диск целиком

5. Восстановите иерархию запоминающих устройств в порядке снижения времени доступа
a) регистры процессора
b) оперативная память
c) быстродействующая память
d) внешняя память

6. Выберите верные утверждения при страничном распределении памяти
a) размер физической страницы оперативной памяти выбирается равным степени двойки.
b) копия всего виртуального адресного пространства находится на диске.
c) смежные виртуальные страницы обязательно находятся на смежных физических сегментах.
d) таблицы страниц хранятся в оперативной памяти.
e) виртуальный и физический адреса представлены в виде пар – порядкового номера страницы и смещения в пределах страницы
f) достоинство страничной виртуальной памяти – сложность организации защиты данных.

7. Какое из утверждений верно при сегментно-страничной организации памяти
a) Виртуальное адресное пространство процесса разделено на сегменты, перемещение же данных между памятью и диском осуществляется страницами – т.е. участками заранее заданного размера.

b) Виртуальное адресное пространство процесса разделено на страницами – т.е. участки заранее заданного размера, перемещение же данных между памятью и диском осуществляется сегментами.

2.8.ЭКЗАМЕНАЦИОННЫЕ ВОПРОСЫ:

1. Первые ОС, их развитие в 65-75 гг.
2. Операционные системы и компьютерные сети
3. Развитие ОС в 80-е годы
4. Составные части ОС
5. Назначение и функции ОС
6. Классификация ОС
7. Требования, предъявляемые к современным ОС
8. Системы пакетной обработки, системы реального времени
9. Многослойная архитектура ОС
10. Ядро и вспомогательные модули ОС
11. Микроядерная архитектура ОС
12. Понятия процесса и потока
13. Состояния потока
14. Вытесняющие и невытесняющие алгоритмы
15. Алгоритмы планирования, основанные на квантовании
16. Алгоритмы планирования, основанные на приоритетах
17. Понятия «прерывание», «вектор прерываний», «обработчик прерываний»
18. Механизм обработки прерывания
19. Классификация методов распределения памяти
20. Идентификация переменных и программ, типы адресов
21. Распределение памяти фиксированными разделами
22. Распределение памяти динамическими разделами
23. Перемещаемые разделы
24. Свопинг и виртуальная память
25. Страничное распределение памяти
26. Сегментное распределение памяти
27. Странично-сегментное распределение памяти
28. Разделяемые сегменты памяти
29. Иерархия запоминающих устройств
30. Кэш-память, принцип действия
31. Понятие файловой системы
32. Цели и задачи файловой системы
33. Дескриптор файла
34. Задачи операционной системы по управлению файлами и устройствами
35. Файловая системы FAT16
36. Файловая система FAT32
37. Файловая система NTFS
38. Команды файла Config
39. Команды файла Autoexec
40. Классификация вирусов
41. Виды и классификация антивирусных программ
42. Создание bat-файлов
43. Динамические структуры программ
44. Последовательные и параллельные структуры программ

КРИТЕРИИ ОЦЕНОК ЗНАНИЙ СТУДЕНТОВ

Отлично

Студент дает полные ответы на теоретические вопросы билета, показывая глубокое знание учебного материала, свободное владение основными понятиями и терминологией; ответ на дополнительный вопрос.

Хорошо

Студент дает ответы на теоретические вопросы билета, показывая прочное знание учебного материала, владение основными понятиями и терминологией; ответ на дополнительный вопрос.

Удовлетворительно

Студент дает неполные ответы на теоретические вопросы билета, показывая поверхностное знание учебного материала, владение основными понятиями и терминологией; при неверном ответе на билет ответы на наводящие вопрос.

Неудовлетворительно

Студент не дает полные ответы на теоретические вопросы билета, показывая лишь фрагментарное знание учебного материала, незнание основных понятий и терминологии; наводящие вопросы остаются без ответа.

3. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ:

3.1. ЛИТЕРАТУРА

ОСНОВНАЯ

1. Гордеев А.В. Операционные системы. Учебник для вузов. 2-е издание. СПб: Питер, 2004. 416 с. (Допущено Министерством образования РФ)
2. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. СПб.: Питер, 2001. 544 с.

ДОПОЛНИТЕЛЬНАЯ

1. Информатика / Под ред. Н. Макаровой. М.: Финансы и статистика, 1997.
2. Фигурнов В.Э. IBM для пользователя от начинающего – до опытного 7-е издание. М.: Инфра-М, 1998
3. Данкин Р. Профессиональная работа с MS DOS: М.: Мир, 1993

3.2. МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

3.2.1. Галаган Т.А., Соловцова Л.А. Операционные системы. Практикум на ПЭВМ. Благовещенск: АмГУ, 2002. (электронный вариант)

ТЕХНИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ

ДИСЦИПЛИНЫ:

Компьютерный класс ПЭВМ

Операционная система MS DOS, WINDOWS 98/2000/XP/NT

4. УЧЕБНО-МЕТОДИЧЕСКАЯ (ТЕХНОЛОГИЧЕСКАЯ) КАРТА ДИСЦИПЛИНЫ

Номер недели	Вопросы, изучаемые на лекции	Занятия (номера)		Используемые нагляд. и метод. пособия	Самостоятельная работа студентов		Форма контроля
		Практич (семин.)	Лаборат.		Содержание	часы	
1	2.2.1		2.4.1	3.2.1	2.6.1 - 2.6.2	2	отчет
2	2.2.2				2.6.1 - 2.6.2	2	
3	2.2.2		2.4.1	3.2.1	2.6.1 - 2.6.2	2	отчет
4	2.2.3				2.6.1 - 2.6.2	2	
5	2.2.3		2.4.2	3.2.1	2.6.1 - 2.6.2	2	отчет
6	2.2.4				2.6.1 - 2.6.2	2	
7	2.2.4		2.4.3	3.2.1	2.6.1 - 2.6.2	2	отчет
8	2.2.4				2.6.1 - 2.6.2	2	
9	2.2.5		2.4.3	3.2.1	2.6.1 - 2.6.2	2	отчет
10	2.2.5				2.6.1 - 2.6.3	3	
11	2.2.5		2.4.4	3.2.1	2.6.1 - 2.6.3	3	отчет, 2.7.1
12	2.2.6				2.6.1 - 2.6.3	3	
13	2.2.6		2.4.5	3.2.1	2.6.1 - 2.6.3	3	отчет
14	2.2.6				2.6.1 - 2.6.3	3	
15	2.2.6		2.4.6	3.2.1	2.6.1 - 2.6.3	3	отчет
16	2.2.7				2.6.1 - 2.6.2	3	
17	2.2.8		2.4.7	3.2.1	2.6.1 - 2.6.2	3	отчет, 2.7.2
18	2.2.8				2.6.1 - 2.6.2	3	

III. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ И ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Лабораторные работы проводятся по подгруппам в компьютерном классе. Задания к лабораторным работам выполняются парой студентов на одном компьютере соответствии с вариантом.

Выполняя задание, студенты пользуются материалом, изложенным в тексте лабораторной работы; готовят письменный отчет, включающий краткое изложение проделанных действий, ответы на контрольные вопросы, выводы.

Преподаватель, принимая лабораторную работу, проверяет навыки, полученные студентами при выполнении задания, отчет, задает дополнительные вопросы по теоретическому материалу.

IV. ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ И ЗАДАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ

ОПЕРАЦИОННАЯ СИСТЕМА MS DOS

Длительное время операционная система MS DOS явно господствовала на подавляющем большинстве персональных компьютеров. С наступлением эры графических операционных средств DOS отошла в тень. Однако ее роль в обеспечении процессов управления ЭВМ по–прежнему остается ощутимой. Знание основ DOS способствует более глубокому пониманию того, как функционирует программное обеспечение последующих поколений.

Лабораторная работа №1
НАЧАЛЬНАЯ ЗАГРУЗКА DOS

Операционная система состоит из следующих частей: базовая система ввода-вывода, загрузчик операционной системы, дисковые файлы IO.SYS и MSDOS.SYS, командный процессор DOS.

Базовая система ввода-вывода (BIOS) находится в постоянной памяти компьютера (ПЗУ). Эта часть операционной системы встроена в компьютер и занимается выполнением наиболее простых и универсальных услуг операционной системы, связанных с осуществлением ввода-вывода. В BIOS включены аппаратные функции системы, используемые функциями операционной системы для получения доступа к ресурсам персонального компьютера.

Загрузчик операционной системы – это очень короткая программа, состоящая из двух частей. Первая часть загрузчика находится в первом секторе диска, она выбирает, с какого из разделов следует продолжить загрузку. Вторая часть загрузчика находится в первом секторе этого раздела, она считывает в память модули DOS и передает им управление.

Командный процессор DOS обрабатывает команды, вводимые пользователем. Он находится в дисковом файле COMMAND.COM на диске, с которого загружается операционная система. Некоторые команды пользователя, например TYPE, DIR, COPY, командный процессор выполняет сам. Такие команды называются внутренними. Для выполнения внешних командный процессор ищет на дисках программу с соответствующим именем. По окончанию работы программы командный процессор удаляет программу из памяти и выводит сообщение о готовности к выполнению команд.

Для выполнения начальной нагрузки DOS необходимо, чтобы компьютер имел жесткий диск с записанной на нем операционной системой DOS. В начале загрузки работают программы проверки оборудования, находящиеся в постоянной памяти компьютера. Если они находят ошибки, то код ошибки выводится на экран. Если ошибка не критическая, то пользователю предоставляется возможность продолжить процесс загрузки, нажав клавишу F1. Если неисправность критическая, то процесс загрузки прекращается.

На большинстве компьютеров перед тестированием оборудованияна экран выводится сообщение о комбинации клавиш, которую надо нажать для входа в программу конфигурирования компьютера, например DEL. При ее нажатии пользователь попадает в экран программы, в котором можно задать текущие дату, время, типы дисководов, назначить носитель, с которого будет выполняться загрузка операционной системы и др. параметры компьютера. Вход в программы конфигурирования можно защитить паролем.

После того, как с диска прочитана BOOT RECORD - программа-загрузчик операционной системы, эта программа считывает в память компьютера модули операционной системы IO.SYS и MSDOS.SYS и передает им управление.

Далее с того же диска читается файл конфигурации системы CONFIG.SYS и в соответствие с указаниями, содержащимися в этом файле, загружаются драйверы устройств и устанавливаются параметры операционной системы. Если файл CONFIG отсутствует, все параметры устанавливаются по умолчанию.

После этого с диска, с которого загружается операционная система, читается командный процессор (файл COMMAND.COM) и ему передается управление. Командный процессор выполняет командный файл AUTOEXEC.BAT, если этот файл имеется в корневом каталоге диска, с которого зпгружается операционная система. В файле AUTOEXEC.BAT указываются программы и команды, выполняемые при каждом запуске компьютера. Если этот файл отсутствует, то DOS запрашивает у пользователя текущие дату и время.

После выполнения файла AUTOEXEC.BAT процесс загрузки операционной системы заканчивается.

Контрольные вопросы

1. Из каких частей состоит операционная система DOS?
2. Что представляет собой базовая система ввода-вывода?
3. Чем отличаются внешние команды от внутренних?
4. Какие действия выполняются при отсутствии файла CONFIG.SYS?
5. Выполнится ли загрузка операционной системы при отсутствии дисковых файлов IO.SYS и MSDOS.SYS?
6. Каким образом можно изменить текущую дату при загрузке операционной системы?

Задание

1. Проследить и описать предшествующую загрузке операционной системы информацию.

Лабораторная работа №2

НАСТРОЙКА КОНФИГУРАЦИИ DOS

Основную роль в установлении конфигурации DOS играют файлы CONFIG.SYS и AUTOEXEC.BAT. При начальной загрузке операционная система DOS считывает из корневого каталога загрузочного диска файлы CONFIG.SYS и AUTOEXEC.BAT и выполняет содержащиеся в них команды.

Файл CONFIG.SYS является текстовым файлом, в котором содержатся специальные команды для настройки конфигурации DOS: подключения различных драйверов, определение размеров системных таблиц и т.д. Задаваемые в нем настройки действуют в течении всего сеанса работы (то есть могут быть изменены только после перезагрузки компьютера).

Если файл CONFIG.SYS отсутствует, то параметры MS DOS будут установлены по умолчанию.

После завершения выполнения файла CONFIG.SYS автоматически выполняется командный файл AUTOEXEC.BAT, если он имеется в корневом каталоге загрузочного диска. Как правило, в файл AUTOEXEC.BAT записывают команды для запуска резидентных программ и других программ, которые целесообразно запускать при каждой загрузке операционной системы, а также команды для задания списка каталогов, установки формата приглашения DOS и др.

Файл конфигурации CONFIG.SYS

CONFIG.SYS создается и редактируется в любом текстовом редакторе, при этом необходимо соблюдать следующие правила:

- файл CONFIG.SYS состоит из специальных команд операционной системы MS DOS по настройке аппаратуры. Формат этих команд имеет особый вид: Имя команды=значение;
- каждая команда располагается на отдельной строке;
- при вводе команд могут использоваться как строчные, так и прописные буквы;
- после ввода команды необходимо нажать клавишу ENTER;
- после создания (редактирования файла) CONFIG.SYS необходимо перезагрузить операционную систему для установки новой конфигурации настройки;
- если в команде файла CONFIG.SYS перед знаком “=” поставить знак вопроса, то при выполнении DOS на экран выведет команду и спросит, надо ли ее выполнять «[Y/N]?»

Наиболее часто используемые команды:

- BREAK=ON или OFF – установка/отмена режима возможности прерывания процесса выполнения программ по нажатию комбинации клавиш CTRL+BREAK (или CTRL+C);
- BUFFERS=<число> - выделение объема оперативной памяти в соответствии с заданным в команде числом буферов (ячеек стандартной длины). Рекомендуется использовать не менее 40 буферов;
- DEVICE=<имя драйвера>[параметры] – подключение драйвера для управления каким-либо устройством;

- DEVICEHIGH=<имя драйвера>[параметры] – загружает драйвер устройства в верхнюю память (UMB), т.е. память с адресами от 640 Кбайт до 1 Мбайта;

Порядок записи команд DEVICE и DEVICEHIGH в файле CONFIG.SYS **очень важен:** сначала устанавливаются драйверы, позволяющие работать с разными видами памяти, которые дополняют стандартную память объемом 640 Кбайт, а затем устанавливаются драйверы, использующие эту память;

- FILES=<число> - устанавливает максимальное количество одновременно открытых файлов. Обычно не менее 20;
- DOS=<параметры> - устанавливает режим использования области высокой памяти (HMA) и обеспечения доступа к блокам верхней памяти (UMB);
- REM или ; - комментарии;
- LASTDRIVE – установка количества букв логических дисков;
- INSTALL=полное имя программы (параметры)- установка резидентной программы;

Начиная с MS DOS версии 6.0 в CONFIG.SYS появилась возможность создания описания различных вариантов загрузки, из которых пользователь может в режиме диалога выбирать нужный на текущий сеанс работы. Для этого файл рассматривается не просто как последовательность строк, а как совокупность блоков, каждый из которых начинается со строки, в которой приводится (начиная с первой колонки) имя блока, заключенного в квадратные скобки. Затем следуют команды, относящиеся к данному блоку. Для организации меню в файле CONFIG.SYS необходимо следовать следующим правилам:

- если в файле CONFIG.SYS нет заголовочных блоков, то команды этого файла выполняются по очереди , в том порядке, в котором они указаны в файле CONFIG.SYS, при этом строки комментариев игнорируются;
- если имеются заголовки блоков , то в блоке [MENU] должно задаваться стартовое меню, выводимое в начале выполнения файла CONFIG.SYS. Пользователь должен выбрать один из пунктов данного меню, после чего начинается выполнение файла CONFIG.SYS, соотвествующего выбранному пользователем пункту меню, а также из блока [COMMON], команды которого выполняются всегда. Содержимое остальных блоков игнорируется ;
- если в начале файла CONFIG.SYS нет заголовка блока, считается , что там стоит блок [COMMON], то есть строки файла CONFIG.SYS до первого заголовка выполняются всегда;
- блоков с именем [COMMON] может быть несколько;
- для организации меню используются следующие команды:

MENUITEM= имя-блока, сообщение – определение пункта меню;

SUBMENU= имя-блока, сообщение – определение пункта меню, при выборе которого выводится подменю (эта возможность используется редко);

MENUDEFAULT= имя-блока, число секунд – определение пункта меню, выполняемого «по умолчанию»: при нажатии клавиши ENTER или по истече-

нии заданной в команде задержки. Как правило, данные команды записываются в блок [MENU] . При выполнении блока меню DOS выведет указанные в предложениях MENUITEM и SUBMENU сообщения и предложит выбрать номер пункта.

Пример файла CONFIG.SYS:

```
[COMMON]
;     загрузка драйверов расширенной и внешней памяти
DEVICE= C:\EXE\MSDOS\HIMEN.SYS /TESTMEN:OFF
DEVICE= C:\EXE\MSDOS\EMM386.SYS RAM
[MENU]
MENUITEM=DOSCONF,  Работа в DOS
MENUITEM=WINDOWS, Работа в WINDOWS
MENUDEFAULT==WINDOWS, 15
[DOSCONF]
;      установка размеров внутренних структур DOS
BUFFERS=10
LASTDRIVE= D
[WINDOWS]
……
[COMMON]
……
```

Командный файл автонастройки AUTOEXEC.BAT

Основное назначение файла AUTOEXEC.BAT состоит в сохранении и оперативной установке значений параметров операционной среды для конкретного пользователя конкретного компьютера. Наиболее часто используемыми командами в составе AUTOEXEC.BAT являются:

- REM – комментарии;
- PAUSE – прерывание выполнения командного файла до тех пор, пока не будет нажата любая клавиша;
- CLS – очистка экрана;
- ECHO – вывод сообщений на экран;
- ECHO OFF – позволяет вывод на экран всех стоящих после нее команд;
- ECHO ON – поддерживает вывод на экран всех сообщений;
- @ - запрет на индикацию одной команды, перед которой стоит этот символ;
- VERIFY ON или OF – установка или отключение режима контрольного чтения при операциях записи файлов (если данный режим включен, то он позволяет выявлять возможные ошибки, но, естественно, замедляет работу);

- PATH [путь] – задание последовательности каталогов, в которых будут последовательно искаться выполняемые файлы, если пользователь при запуске не задает их полное имя;
- PROMT [параметры] – установка формата приглашения в командной строке DOS (по умолчанию используется текущее логическое устройство и символ >, например C:>);
- SET <имя><значение> - устанавливает имя и значение так называемой системной среды.

К одной из главных функций файла автозагрузки относится запуск резидентных программ, которые после выполнения должны остаться в памяти машины для реализации тех или иных служебных функций.

Очень часто из AUTOEXEC.BAT запускаются:

- антивирусные программы (например, VSAFE – резидентная команда слежения за появлением вирусов);
- программы-оболочки (NORTON COMMANDER);
- драйверы клавиатуры, принтера, и т.п.

При использовании меню в файле CONFIG.SYS часто требуется при выборе варианта в меню выполнить соответствующие этому варианту команды и в файле AUTOEXEC.BAT. Это легко сделать, поскольку имя выбранного в меню файла CONFIG.SYS, автоматически присваивается переменной окружения CONFIG. Поэтому в файле AUTOEXEC.BAT можно задавать различные действия в зависимости от значения этой переменной.

Пример файла AUTOEXEC.BAT

```
rem    Фрагмент файла, позволяющий выполнять команды в зависимости от
rem    выбора, сделанного в стартовом меню приведенного примера файла rem
       CONFIG.SYS
@echo off
rem установка пути поиска программ
path C:\EXE\MSDOS; C:\WINDOWS
GOTO %CONFIG%
 : DOSCONF
rem команды, выполняемые при работе DOS
rem установка вида приглашения DOS
promt $p$q
GOTO CONTINUE
 : WINDOWS
rem команды, выполняемые при работе в WINDOWS
rem установка переменных окружения
set TEMP= C:\ WINDOWS\TEMP
set TMP= C:\ WINDOWS\TEMP
GOTO CONTINUE
 : CONTINUE
```

Задание

Дополните имеющиеся на компьютере файлы CONFIG.SYS и AUTOEXEC.BAT таким образом, чтобы при загрузке компьютера появлялось меню, предоставляющие возможность выбора запуска по существующему сценарию или созданному вами, в котором обязательно предусмотреть:

- автоматическую загрузку операционной системы MS-DOS,
- вызов антивирусной программы при загрузке,
- запуск программы кэширования дисков,
- загрузку драйверов в обычную память и верхнюю память,
- задание размеров внутренних структур DOS, таких как число буферов, число одновременно открываемых файлов, последней буквы дисководов и др.,
- установку списка каталогов, в которых производится поиск программ,
- функции управления начальной загрузкой DOS.

Контрольные вопросы

1. В чем состоит назначении файла конфигурации CONFIG.SYS?
2. Перечислите основные команды файла AUTOEXEC.BAT?
3. В течнии какого времени действуют настройки, задаваемые в файле CONFIG.SYS?
4. Перечислите известные вам драйверы, устанавливаемые в файле CONFIG.SYS?

Лабораторная работа №3

ПАКЕТНЫЕ КОМАНДНЫЕ ФАЙЛЫ

Операционная система позволяет записать последовательность команд, которую необходимо повторять, чтобы осуществить периодически выполняемые действия, в специальный файл, называемый командным файлом. Командный файл должен иметь расширение .bat. Последовательность команд, записанную в файл, можно выполнить набрав имя командного файла.

Часто приходится выполнять одни и те же команды или последовательность команд с небольшими отличиями. Например, имеется командный файл Type.bat для вывода на экран содержимого двух файлов. Фрагмент этого командного файла имеет вид:

```
Type имя_файла_1
Type имя_файла_2
```

Если имена выводимых файлов необходимо изменить, то пользователь вынужден корректировать командный файл. Чтобы не переписывать командный файл используют параметры. Всего может использоваться до 9 парамет-

ров, обозначаемых символами %1 - %9. Предыдущий фрагмент можно записать в виде:

Type %1
Type %2

При выполнении командного файла Typ.bat символ %1 будет заменен на значение первого параметра, а %2 - на значение второго параметра. Например, если ввести команду

Type abc.txt xyz.txt

то вместо %1 будет подставлено abc.txt, а вместо 2% - xyz.txt и тем самым будут выполнены следующие команды

Type abc.txt
Type xyz.txt

В командном файле можно использовать символ %0, значение которого - имя выполняемого командного файла.

С помощью средств перенаправления вывода можно выводить сообщения не на экран, а в файл. Это дает возможность протоколировать работу командного файла. Формат команды:

ECHO сообщение >> имя файла

(добавление строки с сообщением в конец файла, если файл не существует, то он создается)

ECHO сообщение > имя файла

(создание файла и запись в него строки с сообщением, если такой файл уже существует, то его старое содержимое будет потеряно).

Командный файл может содержать метки и команды перехода. Это позволяет управлять порядком выполнения команд в файле. Строка командного файла, начинающаяся с двоеточия «:» воспринимается как метка, имя которой набор символов следующих за двоеточием до первого пробела или конца строки. Формат команды перехода:

GOTO метка

Если метка в команде не указана, то процесс пакетной обработки прекращается.

Пример:

```
GOTO met
…
:met
REM продолжение выполнения командного файла.
```

В этом примере после выполнения команды GOTO met выполнение командного файла продолжается со строки, следующей за меткой met, т.е. со строки

REM продолжение выполнения командного файла.

Команда IF позволяет в зависимости от выполнения некоторых условий выполнять или не выполнять команды в командном файле. Формат команды:

IF условие команда

команда – любая допустимая команда, которая выполняется, если условие в команде IF истинно, иначе команда игнорируется;

условие – это одно из следующих выражений:

ERRORLEVEL число – условие истинно тогда, когда код завершения предыдущей выполненной программы больше заданного числа или равен ему, код завершения устанавливается программами при окончании их работы и по умолчанию равен 0;

строка1 == строка2 – условие истинно тогда, когда строка1 и строка2 полностью совпадают, если в этих строках имеются символы %0 - %9, то вместо этих символов подставляются параметры командного файла;

EXIST имя-файла – условие истинно тогда, когда указанный файл уже существует;

NOT условие - истинно тогда, когда условие ложно.

Пример. Рассмотрим создание командного файла Typ.bat, который выводит на экран два файла. Формат вызова имеет вид:

TYP имяфайла1 имяфайла2

В командном файле должно проверяться заданы ли при вызове параметры и существуют ли указанные файлы. Его вид может быть следующим:

```
ECHO OFF
IF -%1== - GOTO no_parametr1
IF -%2== - GOTO no_parametr2
IF NOT EXIST %1 GOTO no_file1
IF NOT EXIST %2 GOTO no_file2
TYPE %1
TYPE %2
GOTO  exit
: no_parametr1
ECHO не задан первый параметр
GOTO exit
: no_parametr2
ECHO не задан второй параметр
GOTO exit
:no_file1
ECHO не найден файл %1
GOTO exit
:no_file2
ECHO не найден файл %2
:exit
```

В строке 2 проверяется задан ли первый параметр командного файла. Если параметр не задан, то происходит переход на метку no_parametr1 и выдается сообщение

Не задан первый параметр

и выполнение командного файла прекращается. Тоже проделывается для второго параметра.

В 4-ой строке проверяется, что указанный в первом параметре файл существует. Если он не существует, то выдается сообщение :

Не найден файл…

и выполнение командного файла прекращается. Следующая строка выполняет аналогичные действия для второго файла.

Иногда в командном файле нужно выполнить различные действия по выбору пользователя. Это можно сделать с помощью функции ASK программы BE из комплекса NORTON UTILITIES или программы CHOICE из MS DOS версии 6. Формат вызова BE ASK следующий:

BE ASK “сообщение”, список символов

Формат вызова программы CHOICE:

CHOICE /C: список-символов сообщение

Обе программы выводят указанное сообщение и ждут, пока пользователь не введет один из указанных в списке символов. Значение переменной ERRORLEVEL устанавливается равным номеру введенного символа в списке.

Пример:

```
BE ASK “Запустить программу ALFA [Y/N]?” YN
CHOICE /C:YN “ Запустить программу ALFA ?”
```

В обоих случаях при ответе “N” значение переменной ERRORLEVEL устанавливается равным 2, при ответе “Y” – равным 1. Пример. Программа ALFA запускается , если пользователь на соответсвующий запрос ответит “Y”

```
BE ASK “Запустить программу ALFA [Y/N]?” YN
IF ERRORLEVEL 2 GOTO continue
ALFA.EXE xxx
:continue
```

С помощью команды BE ASK можно осуществить выбор из меню. Например, создается командный файл, который позволяет выбрать одну из трех игр:Digger, Tetris, Cat.

```
ECHO OFF
ECHO Выберите игру:
ECHO D- Digger
ECHO T- Tetris
ECHO C- Cat
BE ASK “введите D, T или C”, DTC
IF ERRORLEVEL 3 GOTO cat
IF ERRORLEVEL 2 GOTO tetris
DIGGER
GOTO exit
:tetris
TETRIS
GOTO exit
:cat
CAT
:exit
```

Проверки значений переменной ERRORLEVEL следует располагать в порядке убывания значений : сначала проверять на самое большое значение, затем на следующее по убыванию.

Обе программы позволяют указать максимальное время ожидания нажатия клавиши и ответ, принимаемый по умолчанию. Для программы CHOICE это задается параметром /T:символ, число-секунд, параметр указывается в командной строке до сообщения. Для BE ASK - параметрами /TIMEOUT=число-секунд, /DEFAULT=символ. Например, параметры /T:Y,2 (для CHOICE) или /TIMEOUT=2, /DEFAULT=Y (для BE ASK) указывают, что если пользователь в течении 2 секунд не ответит на запрос, будет принят ответ “Y”.

Контрольные вопросы

1. Что такое командный файл?
2. Каким образом можно запротоколировать работу командного файла?
3. Как обозначаются и используются параметры?
4. Какая программа MS DOS позволяет создать диалоговый командный файл?
5. Какие выражения используются для записи условия в команде IF?

Задание

1. Написать командный файл, который выполняет следующие действия:

Вариант 1

Форматирование дискеты;
Копирование на дискету файла, заданного при запуске командного файла;
Переименование скопированного на предыдущем шаге файла, новое имя должно быть задано при запуске командного файла как параметр;
Удаление копии файла с диска С.

Вариант 2

Удаление всех файлов с дискеты;
Копирование файлов из каталога DN, имеющих расширение .exe;
Переименование всех файлов на дискете таким образом, чтобы они начинались на букву «а»;
Вывести на экран содержимое файла, имя которого задано как параметр.

Вариант 3

Создание директории в корневом каталоге, имя задать как параметр;
Перенос в созданную директорию всех файлов из папки Windows с расширением .doc, с изменением расширения на .txt;
Удаление из этой директории всех файлов, начинающихся на букву «с»;
Открыть для редактирования документ, имя которого задается как второй параметр.

Вариант 4

Удаление с дискеты директории, имя которой задано как первый параметр;
Создание на дискете файла, имя которого задано как второй параметр;
Создание двух копий этого файла в корневом каталоге жесткого диска, с именами, заданными как параметры;
Удаление с дискеты файла, созданного на втором шаге.

Вариант 5

Удаление из директории, имя которой задано как параметр, все файлов, имеющих расширение .bak;
Просмотр файл с именем, заданным как параметр;
Копирование этого файла на дискету во вновь созданную директорию с именем заданным как второй параметр.

Вариант 6

Создание текстового файла в корневом каталоге с именем, заданным как параметр;
Копирование из директории «Мои документы» на дискету всех файлов с расширением .doc и .txt;
Удаление с дискеты всех файлов, начинающихся на букву, заданную как второй параметр;
Редактирование документа, созданного на первом шаге.

2. Написать диалоговый командный файл, который выполняет следующие действия по выбору пользователя:

а) Запуск антивирусной программы;

б) Запуск программы-оболочки;

в) Запуск одного из редакторов текста;

г) Запуск Паскаля или Си++.

Указать время ожидания и ответ, принимаемый по умолчанию

ОПЕРАЦИОННАЯ СИСТЕМА WINDOWS

Лабораторная работа №4

ЗАГРУЗКА ОПЕРАЦИОННОЙ СИСТЕМЫ WINDOWS 98

Появление в 1986 г. графической многооконной операционной оболочки Windows стало важным шагом в развитии операционных систем. После своего возникновения она пережила ряд модификаций, причем не все из них бы-

ли удачными. Перечислим основные преимущества операционной системы WINDOWS 98:

- Интегрированная операционная система, ядро которой, загружаемое в момент включения компьютера, активизирует графический интерфейс пользователя и обеспечивает полную совместимость с операционной системой MS DOS;
- Вытесняющая многозадачность – свойство, позволяющее операционной системе самостоятельно в зависимости от внутренней ситуации передавать или забирать управление у того или иного приложения, не позволяющее одному приложению занять все аппаратные ресурсы;
- Многопоточность – свойство, позволяющее выполнять одновременно несколько 32-битовых приложений, называемых процессами;
- Использование технологии Plug and Play обеспечивает помощь при распознавании устройств для их установки и настройки, динамическое изменение состояния и конфигурации системы, интеграцию драйверов устройств и системных компонентов;
- Использование мастеров, служебных программ и ресурсов, обеспечивающих бесперебойную работу системы.

Задание

1. Проследить и описать предшествующую загрузке операционной системы информацию.
2. Описать процедуру загрузки операционной системы WINDOWS 98.
3. Описать модули операционной системы по адресам оперативной памяти.
4. Получить отчет об устройствах компьютера и распределении совместно используемых ими ресурсов
5. Составить полный отчет в тетради.

СЕРВИСНЫЕ ПРОГРАММНЫЕ СРЕДСТВА ОПЕРАЦИОННОЙ СИСТЕМЫ WINDOWS

Сервисные программы – это вспомогательные инструменты, расширяющие и дополняющие функциональность операционных систем. С одной стороны, они не входят в состав операционных систем, а с другой – предназначены для реализации служебных функций по управлению компьютером.

Лабораторная работа №5

СЛУЖЕБНЫЕ ПРОГРАММЫ

Форматирование дисков

Накопители на магнитной основе (гибкие, жесткие диски и т. п.) перед тем, как их можно будет использовать в качестве носителей информации, должны пройти специальную операцию — форматирование. Для форматирова-

ния дисков обычно используется стандартная программа Windows, которая доступна при вызове контекстного меню соответствующего устройства. Можно форматировать диски доступных floppy-устройств, жесткие диски и диски других устройств, допускающих эту процедуру. Исключение составляет системное устройство, с которого была произведена загрузка системы. Обычно это устройство с именем C:. Форматировать можно как новые диски, так и уже бывшие в употреблении, принимая в расчет то, что при форматировании информация, записанная на диске, теряется.

В окне Форматирование необходимо указать параметры форматирования. Обычно, если диск новый, его необходимо полностью форматировать. Если диск уже был форматирован (часто диски форматируются уже при изготовлении), то можно воспользоваться *быстрым форматированием*. В этом случае область физического размещения файлов на дискете не изменяется, очищается лишь заголовочная часть диска. Тогда при случайном форматировании возможно восстановление файлов отформатированного диска при условии, что на диск не копировалось ни одного файла. Обычно такой способ форматирования используется, если необходимо удалить все файлы с дискеты, а их очень много и процесс их удаления обычным способом занимает значительное время.

Полное форматирование целесообразно применять для дисков, на которых появились «сбойные» участки. Эта процедура в некоторых случаях позволяет восстановить работоспособность диска.

При форматировании диска можно сделать его *системным*. В последующем с него можно загрузить компьютер. При загрузке системы с системного диска, созданного средствами Windows, графическая оболочка не загружается и рассмотренная выше команда недоступна. Если на компьютере не установлена операционная система Windows (например, на новом компьютере), то для форматирования диска можно воспользоваться DOS-утилитой Format. Справку по формату ее командной строки можно получить, введя на приглашение DOS-команду: Format /h .

Дефрагментация дисков

Процедура дефрагментации диска связана с особенностями используемой файловой системы FAT. Системы FAT 16 или FAT32 размещают файлы в кластерах, необязательно смежных. Доступ к файлу, расположенному в одном месте диска, когда файл размещается последовательно в смежных кластерах, занимает меньше времени, чем доступ к файлу, фрагменты которого разбросаны по всему диску. Поэтому для увеличения быстродействия системы диск необходимо периодически дефрагментировать. Эта стандартная служебная программа запускается в результате выполнения последовательности команд Пуск -> Программы -> Стандартные -> Служебные -> Дефрагментация диска.

Дефрагментация — это длительная процедура по переносу информации из одних кластеров в другие, в результате которой файлы будут размещаться преимущественно в одном месте диска. Кроме этого файлы передвигаются ближе к началу диска, что также улучшает быстродействие всей системы. Так-

же при выполнении дефрагментации предусмотрены дополнительные возможности исправления ошибок записи на диск.

Проверка диска на наличие ошибок

В процессе эксплуатации диска возможно появление ошибок, связанных со сбоями в процессе записи на него. Некоторые из них можно исправить, не прибегая к процедуре форматирования. Для этого используется стандартная служебная программа проверки диска. Ее запуск осуществляется из главного системного меню с помощью последовательности команд Пуск –> Программы –> Стандартные –> Служебные – > Проверка диска. Она позволяет проверить целостность файловой системы и поверхность диска.

Целостность файловой системы определяется:

- правильностью имен файлов (файлы с неправильными именами, например, нельзя открыть);
- правильностью даты и времени создания файла (неправильные дата и время могут влиять на работу многих программ, обрабатывающих такие файлы);
- уникальностью имен файлов;
- отсутствием файлов с общими кластерами;
- отсутствием кластеров, не принадлежащих ни одному файлу.

Окно настройки параметров процесса проверки диска вызывается по нажатии кнопки Дополнительно... .

Проверка поверхности диска на наличие повреждений и аппаратных ошибок занимает больше времени, чем проверка целостности файловой системы. Для сокращения времени работы программы можно выбрать проверку не всего диска, а только системной области или области данных. При нахождении поврежденных секторов их данные обычно переносятся в другие сектора. Некоторые системные файлы нельзя переносить, поэтому рекомендуется исправление ошибок в секторах, в которых располагаются такие файлы, отменить. Для большей надежности проверки поверхности диска необходимо использовать режим записи на диск (при этом данные на диске не теряются). Для сокращения времени тестирования от этого режима можно отказаться.

Обнаруженные ошибки могут исправляться автоматически в соответствии с заданными параметрами. Для исправления других ошибок файловой системы необходимо использовать другие программы, например известные утилиты Norton Utilities.

Очистка дисков

В процессе эксплуатации дисковых носителей свободное пространство заполняется файлами. Для создания новых файлов необходимо освободить место, занимаемое уже ненужными файлами. Эти файлы удаляются. Удаленные файлы помещаются в корзину на рабочем столе, то есть файлы меняют «прописку», но не удаляются с диска. Для освобождения места, занятого удаленными файлами, не обходимо очистить корзину на рабочем столе.

«Ненужность» тех или иных файлов в конечном итоге определяется пользователем. В помощь пользователю предоставляется специальная программа

очистки диска, которая запускается последовательностью Пуск –> Программы –> Стандартные –> Служебные –> Очистка диска.
Программа очистки диска может найти и освободить место, занимаемое файлами следующих категорий:
• Temporary Internet Files — файлы, которые копируются из Интернета для быстрого просмотра;
• Downloaded Program Files — файлы, которые временно сохраняют элементы ActiveX и приложения Java, автоматически загружаемые из Интернета при просмотре некоторых страниц;
• корзина — удаленные файлы;
• временные файлы — файлы, создаваемые программами при своей работе. Обычно эти файлы удаляются после окончания работы программ, но по тем или иным причинам эти файлы могут остаться на диске.

При недостатке свободного места на диске можно удалить неиспользуемые компоненты Windows либо неиспользуемые программы. Данная возможность предоставляется программой очистки диска. Следует помнить, что при этом освобождается место на том диске, на котором установлено выбранное программное обеспечение.

Дополнительное свободное место можно получить, установив FAT32 вместо FAT16. Эта процедура запускается последовательностью Пуск –> Программы –> Стандартные –> Служебные –> Преобразование диска в FAT32.

Контрольные вопросы

1. Какие программы относятся к служебным?
2. Чем отличается быстрое форматирование от полного?
3. Чем определяется целостность файловой системы?
4. Какие действия выполняются при дефрагментации дисков?
5. Каково назначение программы очистки дисков

Задание

Научитесь выполнять следующие виды работ:
1. форматирование дискет (все возможные способы);
2. отслеживание производительности системы;
3. увеличение эффективного размера диска с помощью утилиты уплотнения диска DriveSpace (теоретически);
4. улучшение производительности жесткого диска путем его дефрагментации;
5. очистка диска;
6. поиск и устранение ошибок с помощью утилиты проверки диска Scandisk.

Лабораторная работа №6
АРХИВАЦИЯ ДАННЫХ

Наиболее простым и эффективным способом восстановления потерянных или испорченных данных на компьютере является создание копий. Для уменьшения места, необходимого для содержания копий, и удобства эксплуатации создано достаточно много программ, позволяющих работать с архивными файлами. Архивный файл может содержать один и более файлов в сжатом виде. Можно как извлекать из этого файла исходные файлы в их первоначальном виде, так и добавлять в него новые.

Среди популярных программ архивации, работающих в среде Windows, можно назвать такие, как WinZip, WinRar, WinArj, NetZip и др.

Архиватор WinZip

После установки этой программы создается не только меню WinZip, но и ярлыки на рабочем столе и в кнопке Пуск, в контекстное меню для папок и файлов добавляются пункты, связанные с архивацией.

Работа с архивом начинается с создания файла архива при помощи последовательности команд File –> New Archive (как в любых программах Windows, любая команда или операция может быть осуществлена разными способами, в частности, клавишами Ctrl + N или кнопкой New на панели инструментов). Файл архива будет создан после задания его имени. При установке флажка Add dialog будет открыто дополнительное диалоговое окно Add, в котором можно указать имена и параметры файлов, добавляемых в архив. В противном случае (если флажок Add dialog не установлен) будет создан пустой архивный файл.

При добавлении файлов в уже существующий архив необходимо открыть выбранный файл архива. Для этого используется команда открытия File –> Open Archive.

Для включения файла в архив используется команда добавления Actions –> Add. В окне Add кроме указания файла (файлов) для добавления в архив задаются параметры включения в архив:

• Action

o Add (and replace) files — добавляет файлы в архив. Если файл в архиве уже есть, то он перезаписывается;

o Freshen existing files — обновляются только те файлы, которые уже есть в архиве. Новые файлы не добавляются. Обновление осуществляется при условии, что обновляемые файлы имеют более старую дату создания;

o Move files — добавляемые файлы удаляются с диска, то есть файлы перемещаются с диска в архив;

o Update (and add) files — обновляются файлы, которые уже есть в архиве, и добавляются новые.

• Compresion — параметр, значение которого определяет степень сжатия файла. Время архивации и степень сжатия имеют обратную зависимость.

• Include subfolders — добавляются все файлы, находящиеся в папках открытой папки.

• Save full path info — запоминается полный путь файла. При извлечении файлов из архива файлы будут помещаться в соответствующие папки. Если папок нет, они будут созданы. В архив можно поместить целое дерево папок с файлами.
• Include only if archive attribute is set — в архив добавляются только те файлы, которые еще не помещались в архив (соответствующий атрибут файла указывает на необходимость архивации файла).
• Reset archive attribute — добавленный в архив файл помечается как уже подвергнутый архивации.
• Include system and hidden files — в архив можно помещать файлы с указанными команд Actions –> Extract.
В появившемся окне необходимо в поле Extract to: ввести имя папки, в которую будут помещаться извлекаемые из архива файлы, а в поле Files: — имена извлекаемых файлов (Если вводится список из нескольких имен, то они разделяются запятыми).
Выбор существующей папки можно осуществить в поле Folders/drivers, для создания новой папки следует воспользоваться кнопкой New folder….

Чтобы извлечь все файлы из архива, необходимо установить параметр All files. Параметр Selected files используется для извлечения файлов, выбранных в поле Folders/drivers перед вызовом команды извлечения. Дополнительные параметры позволяют управлять процессом извлечения файлов из архива:
• Overwrite existing files — при существовании на диске файла с именем, совпадающим с именем извлекаемого файла, первый из них удаляется.
• Skip older files — при существовании на диске более нового файла с именем, совпадающим с именем извлекаемого файла, последний не извлекается из архива.
• Use folder name — извлекаемый из архива файл помещается в папку, имя которой было записано в архив вместе с файлом.

Команда извлечения запускается нажатием кнопки Extract.
В рабочем окне программы WinZip располагается информация о хранимых файлах:
• имя файла:
• дата изменения:
• исходный размер файла;
• размер файла в архиве (в процентах к исходному размеру);
• экономия на размере файла (в байтах);
• полный путь файла.

Программа WinZip позволяет работать с файлами без извлечения их из архива. При этом файл автоматически извлекается из архива в специальную рабочую папку программы WinZip и обрабатывается стандартным образом. Например, документ редактора Word можно открыть прямо в архиве, щелкнув мышью на его имени. Более того, изменения, которые вносятся в документ, фиксируются и в файле архива.

Если документ состоит из нескольких файлов, то для нормальной работы с ним необходимо предварительно извлечь все необходимые документы, так как WinZip не способен отследить внутреннюю логику связей между этими файлами.

При добавлении файлов можно указать пароль для недопущения несанкционированного доступа к файлам архива.

Команда добавления запускается нажатием кнопки Add.

При извлечении файла из архива извлекаемый файл из архива не удаляется. Для извлечения файлов из архива необходимо открыть файл архива и запустить команду извлечения.

Контрольные вопросы

1. Перечислите известные программы архивации данных.
2. В каких целях используется сжатие данных?

Задание

1. Изучите принципы работы архиваторов WinZip и WinRar.
2. Сравните работу архиваторов WinZip и WinRar по степени сжатия данных и скорости работы. Для этого произведите архивацию файлов различного типа (текстовых, графических, программных). Результаты работы поместите в таблицу. Сделайте выводы по полученным результатам.

Лабораторная работа №7

КОМПЬЮТЕРНЫЕ ВИРУСЫ И АНТИВИРУСНЫЕ ПРОГРАММЫ

Компьютерным вирусом называется программа (некоторая совокупность выполняемого кода/инструкций), которая способна создавать свои копии (не обязательно полностью совпадающие с оригиналом) и внедрять их без ведома пользователя в различные объекты/ресурсы компьютерных систем, сетей и т.д. При этом копии сохраняют способность дальнейшего распространения.

Вирусы можно разделить на классы по следующим признакам:

- по среде обитания вируса;
- по способу заражения среды обитания;
- по деструктивным возможностям
- по особенностям алгоритма вируса

По ***среде обитания*** вирусы делятся на сетевые, файловые и загрузочные. *Сетевые* вирусы распространяются по компьютерной сети, *файловые* – внедряются в выполняемые файлы, *загрузочные* – в загрузочный сектор диска (Boot-сектор) или в сектор, содержащий системный загрузчик винчестера (Master Boot Record). Также существуют различные сочетания перечисленных классификаций, например, файлово-загрузочные вирусы, заражающие как файлы, так и загрузочные сектора дисков. Такие вирусы, как правило, имеют довольно сложный алгоритм работы и часто применяют оригинальные методы проникновения в систему.

Способы заражения делятся на *резидентный* и *нерезидентный*. *Резидентный* вирус при инфицировании компьютера оставляет в оперативной памяти свою резидентную часть, которая затем перехватывает обращение операционной системы к объектам заражения и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера. *Нерезидентные* вирусы не заражают память компьютера и являются активными ограниченное время. Некоторые из них оставляют в оперативной памяти небольшие резидентные программы, которые не распространяют вирус.

По деструктивным возможностям вирусы делят на:

· *безвредные,* т.е. не влияющие на работу компьютера (кроме уменьшения свободной памяти на диске в результате своего распространения);

· *неопасные*, влияние которых ограничивается уменьшением свободной памяти на диске и графическими, звуковыми и прочими эффектами;

· *опасные* вирусы, которые могут привести к серьезным сбоям в работе компьютера;

· *очень опасные*, которые могут привести к потере программ, уничтожить данные, стереть необходимую для работы компьютера информацию, записанную в системных областях памяти.

По особенностям алгоритма можно выделить

компаньон-вирусы (companion) - вирусы, не изменяющие файлы. Алгоритм работы которых состоит в том, что они создают для EXE-файлов файлы-спутники, с тем же именем, но с расширением .COM, например, для файла XCOPY.EXE создается файл XCOPY.COM. Вирус записывается в COM-файл и никак не изменяет EXE-файл. При запуске такого файла DOS первым обнаружит и выполнит COM-файл, т.е. вирус, который затем запустит и EXE-файл;

вирусы-"черви" (worm) - вирусы, распространение которых в компьютерной сети , так же как и компаньон-вирусов, не изменяет файлы или сектора на дисках. Они проникают в память компьютера из компьютерной сети, вычисляют сетевые адреса других компьютеров и рассылают по этим адресам свои копии. Такие вирусы иногда создают рабочие файлы на дисках системы, но могут вообще не обращаться к ресурсам компьютера (за исключением оперативной памяти).

"паразитические" - все вирусы, которые при распространении своих копий обязательно изменяют содержимое дисковых секторов или файлов. В эту группу относятся все вирусы, которые не являются "червями" или "компаньон".

"студенческие" - крайне примитивные вирусы, часто нерезидентные и содержащие большое число ошибок;

"стелс"- вирусы (вирусы-невидимки), представляющие собой весьма совершенные программы, которые перехватывают обращения DOS к пораженным файлам или секторам дисков и "подставляют" вместо себя незараженные участки информации. Кроме этого, такие вирусы при обращении к файлам используют достаточно оригинальные алгоритмы, позволяющие "обманывать" резидентные антивирусные мониторы.

“полиморфик”-вирусы (самошифрующиеся или вирусы-призраки) – достаточно трудно обнаруживаемые вирусы, не имеющие сигнатур, т.е. не содержащие ни одного постоянного участка кода. В большинстве случаев два образца одного и того же полиморфик-вируса не будут иметь ни одного совпадения. Это достигается шифрованием основного тела вируса и модификациями программы-расшифровщика.

“макро-вирусы” – вирусы этого семейства используют возможности макро-языков, встроенных в системы обработки данных (текстовые редакторы, электронные таблицы и т.д.). В настоящее время наиболее распространены макро-вирусы заражающие текстовые документы редактора Microsoft Word.

Хотя вирусные атаки случаются не очень часто, общее число вирусов слишком велико, а ущерб от “хулиганских” действий вируса в системе может оказаться значительным. Существуют вирусы, которые могут привести к потере программ, уничтожить данные, стереть необходимую для работы компьютера информацию, записанную в системных областях памяти, привести к серьезным сбоям в работе компьютера. В результате этих действий Вы можете навсегда потерять данные, необходимые для работы и понести существенный моральный и материальный ущерб. “Эпидемия” компьютерного вируса в фирме (неважно - большой или маленькой) может полностью дестабилизировать ее работу. При этом может произойти сбой в работе, как отдельных компьютеров, так и компьютерной сети в целом, что повлечет за собой потерю информации, необходимой для нормальной работы и потерю времени, которое будет затрачено на восстановление данных и приведением компьютеров и/или сети в рабочее состояние.

Основные симптомы вирусного поражения следующие:

· замедление работы некоторых программ;
· увеличение размеров файлов (особенно выполняемых);
· появление не существовавших ранее “странных” файлов;
· уменьшение объема доступной оперативной памяти (по сравнению с обычным режимом работы);
· внезапно возникающие разнообразные видео и звуковые эффекты.

Способы защиты от вирусов

Одним из основных последствий деятельности вирусов является потеря или порча информации. Поэтому для обеспечения устойчивой и надежной работы необходимо (или, по крайней мере, желательно) всегда иметь чистые, незараженные (эталонные) копии используемой информации и программного обеспечения.

При заражении вирусом нарушается целостность информации. Использование специальных утилит, которые позволяют отслеживать информацию о системных областях и файлах (контрольные суммы, размеры, даты создания и др.), позволяет относительно быстро обнаружить проникновение вирусов.

Использование специальных антивирусных средств, в большинстве случаев, позволяет не только выявить вирусное вторжение, но и оперативно обезвредить обнаруженные вирусы и восстановить испорченную информацию.

Для защиты от вирусов необходимо применять специализированные программы. Эти программы можно разделить на несколько видов: детекторы, доктора (фаги), ревизоры, доктора-ревизоры, фильтры и вакцины.

Программы-детекторы позволяют обнаруживать файлы, зараженные одним из нескольких известных вирусов.

Программы-доктора или фаги – лечат зараженные программы или диски, выкусывая из зараженных программ тело вируса, т.е. восстанавливая программу в том состоянии, в котором она находилась до заражения вирусом.

Программы-ревизоры сначала запоминают сведения о состоянии программ и системных областей дисков, а затем сравнивают их состояния с исходным. При выявлении несоответствий сообщается пользователю.

Доктора-ревизоры - это гибриды ревизоров и докторов, т.е. программы, которые не только обнаруживают изменения в файлах и системных областях, но и могут в случае изменения автоматически вернуть их в исходное состояние.

Программы-фильтры располагаются резидентно в оперативной памяти компьютера и перехватывают те обращения к операционной системе, которые используются для размножения и нанесения вреда, и сообщают о них пользователю. Пользователь может разрешить или запретить выполнение соответствующей операции.

Программы-вакцины или иммунизаторы модифицируют программы и диски таким образом, что это не отражается на работе программ, но тот вирус, от которого производится вакцинация, считает эти программ или диски уже зараженными. Эти программы неэффективны и редко используются.

Среди популярных и эффективных антивирусных программных систем можно выделить:

- антивирус Касперского — AntiViral Toolkit Pro (AVP);
- Doctor Web;
- Panda Antivirus;
- F-Secure Anti-Virus;
- Sophos Anti-Virus;
- Norton AntiVirus.

К сожалению, не существует антивирусных программ, которые бы обеспечивали абсолютную защиту от вирусов. Это связано, прежде всего, с постоянным развитием отрасли написания вирусов, в ответ на которое создатели антивирусов, в свою очередь, стремятся как можно быстрее создать нужную вакцину. Поэтому необходимо пользоваться антивирусами, версии которых постоянно обновляются.

Антивирус Касперского — AntiViral Toolkit Pro

Лаборатория Касперского предлагает достаточно широкий набор антивирусных средств AntiViral Toolkit Pro (AVP) для большинства известных опе-

рационных систем. После установки антивирусной программы монитор запускается при загрузке компьютера (см. папку Автозагрузка). Работа антивируса несколько замедляет работу компьютера.

Монитор антивируса работает постоянно, проверяя файлы, к которым осуществляется текущий доступ. Файлы, которые должны быть проверены, определяются вкладкой Объекты при помощи указания соответствующего типа файлов и файлов, имеющих вложения. Во вкладке Действия определяется процедура, которая задействуется при обнаружении вируса.

В антивирусе AVP предусмотрена возможность сканирования всех файлов, находящихся на локальных дисках. Соответствующая кнопка есть в мониторе, но рекомендуется использовать специальную программу, входящую в комплект поставки. Здесь можно выбрать устройство для сканирования, например floppy-дисковод, отдельную папку, файлы которой необходимо сканировать. Антивирус позволяет сканировать любые доступные сетевые устройства

Задание

1. Выполните установку с диска антивирусной программы.
2. Установите автоматическое сканирование загружаемых дисков антивирусной программы
3. Выполните проверку на вирус диска C:

Контрольные вопросы

1. По каким признакам можно классифицировать вирусы?
2. Перечислите возможные симптомы вирусного поражения.
3. Почему необходимо бороться с компьютерными вирусами?
4. Перечислите известные антивирусные программы.

V. КОНСПЕКТ ЛЕКЦИЙ

ЭВОЛЮЦИЯ ОПЕРАЦИОННЫХ СИСТЕМ

Появление первых операционных систем

Рождение цифровых вычислительных машин произошло после окончание второй мировой войны. В середине 40-х созданы первые вычислительные устройства. Операционные системы еще не существовали, и все задачи организации вычислительного процесса решались вручную каждым программистом с пульта управления, который представлял собой примитивное устройство, состоящее из кнопок, переключателей и индикаторов.

С середины 50-х начался новый период развития вычислительной техники, связанный с появлением полупроводниковых элементов. К библиотекам математических и служебных подпрограмм добавился новый тип программного обеспечения – трансляторы.

Выполнение каждой программы стало включать:

загрузку нужного транслятора,

запуск транслятора и получение результирующей программы в машинных кодах,

связывание программы с библиотечными подпрограммами,

загрузку программы в оперативную память,

запуск программы,

вывод результатов на периферийное устройство.

Для выполнения этих задач в штат вычислительных центров были введены должности операторов. Они стали первыми системными программами, предназначенными для *управления вычислительным процессом*. Оператор составлял пакет заданий, которые в дальнейшем запускались на выполнение управляющей программой – монитором. Монитор также был способен самостоятельно обрабатывать наиболее часто встречающиеся при работе пользовательских программ аварийные ситуации, такие как отсутствие исходных данных, переполнение регистров, деление на ноль, обращение к несуществующей области памяти и др.

Системы пакетной обработки повысили эффективность использования компьютера, но при этом программисты-пользователи лишились непосредственного доступа к компьютеру, что снижало эффективность их работы – внесение любого исправления требовало значительно больше времени, чем при интерактивной работе за пультом машины.

Появление мультипрограммных операционных систем для майнфреймов

Следующий важный период развития ОС относится к 1965 – 1975 годам. В технической базе вычислительных машин произошел переход от отдельных полупроводниковых элементов типа транзисторов к интегральным микросхе-

мам.

В этот период были реализованы практически все основные механизмы, присущи современным ОС: мультипрограммирование, мультипроцессирование, поддержка многотерминального многопользовательского режима, виртуальная память, файловые системы, разграничение доступа, сетевая работа.
В эти годы начинается расцвет системного программирования. Теперь процессор уже не простаивал, пока одна из программ выполняла операцию ввода-вывода, а переключалась на другую готовую к выполнению программу, но пользователь по-прежнему был лишен возможности интерактивного взаимодействия со своими программами.

Системы разделения времени были рассчитаны на многотерминальные системы, когда пользователь работает за своим терминалом. В числе первых ОС разделения времени, разработанных в середине 60-х, были TSS (компании IBM), GTSS и MULTIS (Массачусетский технологический институт совместно с Bell Labs и компанией General Electric). В таких системах эффективность использования оборудования ниже, чем в системах пакетной обработки, что являлось платой за удобство пользователя.

Многотерминальный режим использовался и в системах пакетной обработки. При этом не только оператор, но и все пользователи получали возможность формировать свои задания и управлять их выполнением со своего терминала. Такие операционные системы получили название систем удаленного ввода заданий. Такие вычислительные системы с удаленными терминалами, сохраняя централизованный характер обработки данных, в какой-то степени являлись прообразом современных сетей, а соответствующее программное обеспечение – прообразом сетевых операционных систем.

Еще одной важной тенденцией этого периода является создание семейств программно-совместимых машин и операционных систем для них. Примерами которых являются серии машин IBM/360 и IBM/370 (аналоги этих семейств советского производства – машины серии ЕС), PDP-11 (советские аналоги СМ-3, СМ-4, СМ-1420).

Несмотря на необозримые размеры и множество проблем, OS/360 и другие ей подобные операционные системы этого поколения удовлетворяли большинству требований потребителей. За это десятилетие был сделан огромный шаг вперед и заложен прочный фундамент для создания современных операционных систем.

Операционные системы и глобальные сети

В начале 70-х появились первые сетевые операционные системы, которые в отличие от многотерминальных ОС позволяли не только рассредоточить пользователей, но и организовать распределенное хранение и обработку данных между несколькими компьютерами, связанными электрическими связями. Значимые практические результаты по объединению компьютеров в сети были получены в конце 60-х, когда с помощью глобальных сетей и техники коммутации пакетов удалось реализовать взаимодействие машин класса мэйнфреймов

и суперкомпьютеров.

В 1969 году Министерство обороны США инициировало работы по объединению суперкомпьютеров оборонных и научно-исследовательских центров в единую сеть. Эта сеть получила название ARPANET и явилась отправной точкой для создания самой известной глобальной сети – Интернета.

В 1974 году компания IBM объявила о создании собственной сетевой архитектуры для своих мэйнфреймов, получившей название SNA (System Network Architecture). Эта многоуровневая архитектура, во многом подобная стандартной модели OSI, появившейся несколько позже.

В это же время в Европе велись активные работы по созданию и стандартизации сетей X.25. Эти сети с коммутацией пакетов не были привязаны к какой-либо конкретной операционной системе. После получение статуса международного стандарта в 1974 году протоколы X.25 стали поддерживаться многими операционными системами. С 1980 года компания IBM включила поддержку протоколов X.25 в архитектуру SNA и в свои операционные системы.

Операционные системы мини-компьютеров и первые локальные сети

К середине 70-х наряду с мэйнфреймами широкое распространение получили мини-компьютеры, такие как PDP-11, Nova, HP. Мини-компьютеры первыми использовали преимущества больших интегральных схем, позволившие реализовать достаточно мощные функции при сравнительно невысокой стоимости компьютера.

Их архитектура была значительно упрощена, что нашло отражение в их операционных системах. Важной вехой в истории мини-комппьютеров и вообще в истории операционных систем явилось создание ОС UNIX, которая первоначально предназначалась для поддержания режима разделения времени. С середины 70-х годов началось массовое использование ОС UNIX. К этому времени 90% программного кода UNIX было написано на языке высокого уровня C.

Первые локальные сети строились с использованием нестандартого коммуникационного оборудования. Программное обеспечение также было нестандартным и реализовывалось в виде пользовательских приложений. Первое сетевое приложение для UNIX – программа UUCP (UNIX-to-UNIX Copy Program) появилось в 1976 году и начала распространяться с версией 7AT&T UNIX с 1978 года. Эта программа позволяла копировать файлы с одного компьютера на другой в пределах локальной сети через различные аппаратные интерфейсы, могла работать через глобальные связи, например модемные.

Развитие операционных систем в 80-е годы

К наиболее важным событиям этого десятилетия можно отнести разработку стека TCP/IP, становление Интернета, стандартизацию технологий ЛВС, появление персональных компьютеров и операционных систем для них.

Рабочий вариант стека протоколов TCP/IP был создан в конце 70-х годов. В 1983 году стек протоколов TCP/IP был принят Министерством обороны США в качестве военного стандарта.

Внедрение протоколов TCP/IP в ARPANET придало этой сети все основные черты современного Интернета.

Начало 80-х связано с появлением персональных компьютеров, что явилось знаменательным событием для истории операционных систем.

Однако первая версия наиболее популярной операционной системы раннего этапа развития персональных компьютеров – MS-DOS. Разработчики первых персональных компьютеров считали, что при индивидуальном использовании компьютера и ограниченных возможностях аппаратуры нет смысла в поддержке мультипрограммирования.

Недостающие функции для MS-DOS и подобных ей ОС компенсировались внешними программами, предоставляющими пользователю удобный графический интерфейс (например, Norton Commander) или средства тонкого управления дисками (например, PC Tools). Наибольшее влияние на развитие программного обеспечения для персональных компьютеров оказала операционная система Windows, представлявшая собой настройку над MS-DOS.

1. ОПЕРАЦИОННАЯ СИСТЕМА

1.1. ОСНОВНЫЕ ПОНЯТИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ

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.

Важно отметить, что операционная система это комплекс взаимосвязанных программ, неоднородный по характеру и многоплановый по уровню. Этот комплекс программ динамичен по своему составу: из него можно удалять и в него добавлять определенные части.

К числу основных ресурсов, управление которыми осуществляет ОС, относятся процессоры, основная память, таймеры, наборы данных, диски, накопители на магнитных лентах, принтеры, сетевые устройства и другие. Ресурсы компьютера распределяются между процессами. Для решения задач управления ресурсами разные ОС используют различные алгоритмы, особенности которых, в конечном счете, и определяют облик ОС.

Большинство современных ОС представляют собой хорошо структурированные модульные системы, способные к развитию, расширению и переносу на новые платформы.

1.2. НАЗНАЧЕНИЕ И ФУНКЦИИ ОС

ОС представляет собой комплекс взаимосвязанных программ, действующий как интерфейс между приложениями и компьютером с одной стороны, и

аппаратурой с другой. В соответствии с этим она выполняет две группы функций:

- Предоставление пользователю или программисту вместо реальной аппаратуры компьютера расширенной виртуальной машины, с которой удобней работать и легче программировать,
- Повышение эффективности использования компьютера путем рационального использования и управления его ресурсами в соответствии с некоторым критерием.

Управление ресурсами включает решение следующих задач, не зависящих от типа ресурса задач:

- планирование ресурса, т.е. определение, какому процессу, когда и в каком количестве следует выделить данный ресурс,
- удовлетворение запросов на ресурсы,
- отслеживание состояния и учет использования ресурса – т.е. поддержание оперативной информации о том, занят или свободен ресурс, какая доля ресурса распределена,
- разрешение конфликтов между процессами.

Для решения этих общих задач разные ОС используют различные алгоритмы, которые и определяют облик ОС в целом. В отличие от функции виртуальной расширенной машины большинство функций управления ресурсами выполняются ОС автоматически и прикладному программисту недоступны.

Функции ОС автономного компьютера обычно группируются либо в соответствии с типами локальных ресурсов, которыми управляет ОС, либо в соответствии со специфическими задачами, применимыми ко всем ресурсам. 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 функции операционной системы.

1. Прием от пользователя заданий или команд и их обработка. Как правило, это команды запуска и приостановки программ, операции с файлами.

2. Загрузка в оперативную память, подлежащих исполнению программ.

3. Распределение памяти и организация виртуальной памяти.

4. Запуск программ.

5. Идентификация всех программ и данных.

6. Прием и выполнение различных запросов от выполняющих приложений.

7. Обслуживание всех операций ввода-вывода.

8. Обеспечение работы систем управления файлами.

9. Обеспечение режима мультипрограммирования, т.е. организация па-

раллельного выполнения двух и более программ на одном процессоре, создающая видимость их одновременного выполнения.

10. Планирование и диспетчеризация задач в соответствии с заданными стратегией.

11. Организация механизма обмена сообщениями и данными между выполняющимися программами.

12. Обеспечение сохранности данных, защита приложений друг от друга, защита самой ОС от исполняемых приложений.

13. Аутентификация и авторизация пользователя (проверка правильности имени и пароля пользователя и назначение определенных прав).

14. Предоставление услуг на случай частичного сбоя системы.

1.3. КЛАССИФИКАЦИЯ ОПЕРАЦИОННЫХ СИСТЕМ.

Традиционно различают ОС общего и специального назначения. ОС специального назначения подразделяют на ОС организации и ведения баз данных, решения задач реального времени и т.п.

По режиму обработки задач различают однопрограммные и мультипрограммные. Под мультипрограммирование понимается способ организации вычислений, когда на одном процессоре поочередно выполняется несколько задач, создавая видимость их одновременного выполнения.

Современные ОС реализуют и мультипрограммный и мультизадачный режимы.

Выделяют также системы многопроцессорной обработки. Мультипроцессирование приводит к усложнению всех алгоритмов управления ресурсами. (Windows NT, NetWare 4.1, OS/2, Open Server 3.x, Solaris 2.x)

По способу распределения процессорного времени между несколькими одновременно существующими в системе процессами выделяют две группы алгоритмов:

невытесняющая многозадачность (NetWare, Windows 3.x);

вытесняющая многозадачность (Windows NT, OS/2, Unix.

По способу взаимодействия с компьютером ОС делят на диалоговые системы и системы пакетной обработки.

Операционная система компьютерной сети аналогична ОС автономного компьютера. При организации сетевой работы она играет роль интерфейса экранирующего от пользователя все детали низкоуровневых программно-аппаратных средств сети.

В зависимости от того, какой виртуальный образ создает СОС, различают сетевые ОС и распределенные ОС.

1.4. ТРЕБОВАНИЯ, ПРЕДЪЯВЛЯЕМЫЕ К СОВРЕМЕННЫМ ОПЕРАЦИОННЫМ СИСТЕМАМ.

Современная ОС должна поддерживать мультипрограммную обработку, виртуальную память, своппинг, многооконный графический интерфейс пользователя и др. Кроме того, к ОС предъявляются эксплутационные требования:

Расширяемость.
Переносимость.
Совместимость.
Надежность и отказоустойчивость.
Безопасность.
Производительность.

1. 5. ПРЕРЫВАНИЯ

При включении компьютера операционная система автоматически загружается с диска в оперативную память.

Функционирование компьютера после включения питания начинается с запуска программы первоначальной загрузки – *Boot Track*. Программа Boot Track инициализирует основные аппаратные блоки компьютера и регистры процессора *(CPU)*, накопитель памяти, контроллеры периферийного оборудования. Затем загружается ядро ОС – *Operating System Kernel.* Дальнейшее функционирование ОС осуществляется как реакция на события, происходящие в компьютере. Наступление того или иного события сигнализируется прерываниями – *Interrupt.* Источниками прерываний могут быть как аппаратура *(HardWare),* так и программы (*SoftWare*).

Аппаратура «сообщает» о прерывании асинхронно (в любой момент времени) путем пересылки в CPU через общую шину сигналов прерываний. Программа «сообщает» о прерывании путем выполнения операции *System Call*. Примеры событий, вызывающих прерывания: попытка деления на ноль, запрос на системное обслуживание, завершение операции ввода/вывода, неправильное обращение к памяти.

Каждое прерывание обрабатывается соответственно обработчиком прерываний (*Interrupt handler*), входящим в состав ОС.

Переход от прерываемой программы к обработчику прерываний и обратно должен выполняться как можно быстрей. Одним из быстрых методов является использование таблицы, содержащей перечень всех допустимых для компьютера прерываний и адреса соответствующих обработчиков. Такая таблица называется *вектором прерываний* (*Interrupt vector*) и хранится в начале адресного пространства основной памяти (UNIX / MS DOS).

Для корректного возвращения к прерванной программе перед передачей управления обработчику прерываний, содержимое регистров процессора запоминается либо в памяти с прямым доступом либо в системном стеке – *System*

Stack.

Таким образом, прерывания – механизм, позволяющий координировать параллельное функционирование отдельных устройств вычислительной системы и реагировать на особые состояния, возникающие при работе процессора. Прерывание обязательно влечет за собой изменение порядка выполнения команд процессором.

Механизм обработки прерываний независимо от архитектуры вычислительной системы подразумевает выполнение некоторой последовательности шагов:

1. Установление факта прерывания и идентификация прерывания.
2. Запоминание состояния прерванного процесса вычислений, которое определяется значением счетчика команд, содержимым регистров процессора и другую информацию.
3. Передача управления аппаратно обработчику прерывания.
4. Сохранение информации о прерванной программе, которую не удалось сохранить аппаратными средствами на шаге 2.
5. Выполнение программы, связанной с обработкой прерывания.
6. Восстановление информации о прерванном программе (шаг обратный шагу 4)
7. возврат на прерванную программу.

Шаги 1 – 3 реализуются аппаратно, шаги 4 – 7 программно.

Прерывания можно разделить на два класса: внешние (асинхронные) и внутренние (синхронные).

Управление ходом выполнения задач со стороны операционной системы заключается в организации реакций на прерывания, в организации обмена информацией (данными и программами), в предоставлении необходимых ресурсов, в динамике выполнения задачи и организации сервиса. Причины прерываний определяет ОС (модуль, называемый супервизор прерываний), она же выполняет, требуемые в данной ситуации.

Супервизор прерываний прежде всего сохраняет в дескрипторе текущей задачи рабочие регистры процессора. Далее определяет подпрограмму, которая должна выполнить действия, связанные с обслуживанием текущего запроса и устанавливает необходимый режим обработки прерывания. После его обработки управление вновь передается ядру операционной системы модулю диспетчеризации задач. Он в соответствии с принятой дисциплиной распределения процессорного времени восстановит контекст задачи, которая выбрана на выполнение процессором.

Если бы контекст процесса сохранялся просто в стеке, как это обычно реализуется аппаратурой, а в не специальных структурах данных, называемых дескрипторами, не было бы возможности гибко подходит к выбору следующей задачи.

Был рассмотрен общий принцип. В конкретных процессорах и в конкретных ОС могут существовать некоторые отступления и дополнения. Например, в

современных процессорах часто имеются специальные аппаратные возможности для сохранения контекста прерываемого вычислительного процесса непосредственно в дескрипторе процессора.

1.6. Архитектура операционной системы

Сложность современных ОС приводит к сложности ее архитектуры, под которой понимают структурную организацию ОС на основе различных программных модулей. Какой-то единой архитектуры ОС не существует, но существуют универсальные подходы к структурированию ОС.

Наиболее общим из этих подходов является разделение всех ее модулей на две группы:

- Ядро – модули, выполняющие основные функции ОС;
- Модули, выполняющие вспомогательные функции ОС.

Модули ядра взаимодействуют с аппаратными средствами непосредственно и потому ядро ОС должно постоянно храниться в компьютере. В частности, программное обеспечение, входящее в состав ядра, отвечает за проверку работоспособности компьютера и выполнение элементарных (базовых) операций, связанных с работой монитора, клавиатуры, магнитных накопителей и т.п.

В состав ядра входят функции, решающие внутрисистемные задачи организации вычислительного процесса, такие как переключение контекстов, обработка прерываний. Эти функции недоступны для приложений. Другой класс функций ядра служит для поддержки приложений, создавая для них так называемую прикладную программную среду. Функции ядра, которые могут вызываться приложениями, образуют интерфейс прикладного программирования – **API**.

Функции, выполняемые модулями ядра, являются наиболее часто используемыми функциями ОС, поэтому скорость их выполнения определяет производительность всей системы в целом. Для обеспечения высокой скорости работы ОС все модули ядра или их большая часть постоянно находятся в оперативной памяти, то есть являются *резидентными.*

Остальные 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 математических функций, библиотека ввода-вывода и т.д.

Модули ОС, оформленные в виде утилит, системных обрабатывающих программ и библиотек, обычно загружаются в оперативную память только на выполнение своих функций, т.е. являются *транзитными.*

Вычислительную систему, работающую под управлением ОС на основе ядра, можно рассматривать как систему, состоящую из трех иерархически расположенных слоев: нижний слой образует аппаратура, промежуточный ядро, верхний слой – утилиты, обрабатывающие программы и приложения. Слоистую структуру вычислительной системы принято обозначать в виде концентрических окружностей. При такой организации Ос приложения не могут взаимодействовать непосредственно с аппаратурой, а только через слой ядра.

Поскольку ядро представляет собой сложный многофункциональный комплекс, то многослойный подход распространяется и на структуру ядра.

Ядро может состоять из следующих слоев:

Средства аппаратной поддержки ОС.

Машинно-зависимые компоненты ОС.

Базовые механизмы ядра.

Менеджеры ресурсов.

Интерфейс системных вызовов.

Приведенное разбиение ядра ОС на слои является достаточно условным. В реальной системе количество слоев и распределение между ними функций может быть иным.

Альтернативой классическому способу построения операционной системы является микроядерная архитектура, в которой в привилегированном режиме остается работать только очень небольшая часть ОС, называемая микроядром. В состав микроядра обычно входят машинно-зависимые модули, выполняющие базовые (но не все!) функции ядра по управлению процессами, обработке прерываний, управлению виртуальной памятью, пересылке сообщений. Набор функций микроядра обычно соответствует функциям слоя базовых механизмов обычного ядра. Все остальные высокоуровневые функции ядра оформляются в виде приложений, работающих в пользовательском режиме. Важнейшей задачей реализации микроядерной структуры является наличие удобного и эффективного механизма вызова процедур одного из процессов из другого. Микроядерные ОС удовлетворяют большинству требований, предъявляемым к современным ОС. Они обладают переносимостью, расширяемостью, надежностью, но за эти достоинства приходится платить снижением производительности.

1.7. Функциональные компоненты операционной системы

Функции операционной системы группируются в соответствии с типами ресурсов, которыми управляет ОС. Иногда такие группы функций называют подсистемами. Наиболее важными из них являются подсистемы управления памятью, файлами и внешними устройствами. Общими для всех ресурсов являются подсистемы пользовательского интерфейса, защиты данных и админи-

стрирования.

Управление процессами

Для каждого вновь созданного процесса ОС генерирует системные информационные структуры, содержащие данные о потребностях процесса в ресурсах и действительно выделенных ему ресурсах.

Процессы, порождаемые пользователями и их приложениями, называются *пользовательскими*, а инициированные самой ОС для выполнения своих функций – *системными*.

На протяжении своего существования процесс может быть многократно прерван и продолжен. Для его возобновления восстанавливается состояние его операционной среды, определяющееся состоянием регистров, программного счетчика, режимов работы процессора, указателями на открытые файлы, информацией о незавершенных операциях ввода-вывода, кодами ошибок выполняемых данным процессом системных вызовов и др. Данной подсистеме посвящена глава 2.

Управление памятью

Эта подсистема включает распределение физической памяти между существующими в данный момент процессами, загрузку кодов и данных процессов в отведенные им области памяти, настройку частей кодов программы на физические адреса памяти, защиту областей памяти каждого процесса. Данной подсистеме посвящена глава 3.

Управление файлами и внешними устройствами

Способность ОС к экранированию сложностей реальной аппаратуры проявляется в файловой системе. Особенностям файловых систем посвящена глава 4.

При выполнении своих функций файловая система взаимодействует с подсистемой управления внешними устройствами, которая осуществляет передачу данных между дисками и оперативной памятью.

Подсистема управления внешними устройствами, называемая также подсистемой ввода-вывода исполняет роль интерфейса ко всем устройствам, подключенным к компьютеру.

Программа, управляющая конкретной моделью внешнего устройства, называется драйвером этого устройства. Драйвер может управлять единственной моделью устройства или группой устройств определенного типа.

Созданием драйверов устройств занимаются как разработчики конкретной ОС, так и специалисты компании, выпускающей это устройство. ОС должна поддерживать хорошо определенный интерфейс между драйверами и остальной частью ОС.

Защита данных и администрирование

Безопасность данных обеспечивается средствами отказоустойчивости ОС. Первым этапом защиты от несанкционированного доступа является процедура

логического входа. Права пользователей при обращении к ресурсам и выполнение ими тех или иных действий определяет администратор.

Функции аудита заключаются в фиксации всех событий, от которых зависит безопасность системы. Список событий также назначается администратором.

Поддержка отказоустойчивости реализуется резервированием, т.е. хранением нескольких копий данных на разных дисках или накопителях, и использованием нескольких процессоров.

Поддержка отказоустойчивости также входит в обязанности системного администратора.

2. ПРОЦЕССЫ И ПОТОКИ

Важнейшей функцией ОС является организация рационального использования всех аппаратных и информационных ресурсов. Задачи управления ресурсами более сложны в мультипрограммных ОС, в которых за ресурсы конкурируют сразу несколько приложений. При этом программы попеременно выполняются на одном процессоре и совместно используют такие ресурсы компьютера, как оперативная и внешняя память, устройства ввода-вывода, данные.

В зависимости от выбранного критерия эффективности ОС делятся на системы пакетной обработки данных, системы разделения времени, системы реального времени. Каждый тип ОС имеет специфические внутренние механизмы и особые области применения. Некоторые Ос могут одновременно поддерживать несколько режимов.

2.1. ПОНЯТИЯ «ПРОЦЕСС» и «ПОТОК».

Чтобы поддерживать мультипрограммирование, ОС должна определить и оформить для себя те внутренние единицы работы, между которыми будет разделяться процессор и другие ресурсы компьютера. В настоящее время в большинстве ОС определены два типа единиц работы. Более крупная единица работы, носящая название *процесса*, или *задачи,* требует для своего выполнения нескольких более мелких работ, для обозначения которых используют термины *«поток»* или *«нить».*

Процесс рассматривается ОС как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени. Процессорное время распределяется ОС между другими единицами работы – *потоками,* представляющими собой последовательности команд. Потоки возникли в ОС как средство распараллеливания вычислений, облегчающее работу программиста. В ОС, не поддерживающей потоков, процесс всегда состоит из одного потока, а программисту приходится решать задачу синхронизации нескольких параллельных ветвей программы.

Создать процесс означает создать *описатель процесса,* в качестве которого выступает одна или несколько информационных структур, содержащих все

сведения о процессе, необходимые операционной системе для управления им.

При управлении процессами ОС использует два основных типа информационных структур дескриптор процесса и контекст процесса. *Дескриптор процесса* содержит информацию, необходимую ядру в течение всего жизненного цикла процесса, независимо от того находится он в пассивном или активном состоянии, находится его образ (совокупность его кодов и данных) в оперативной памяти ли выгружен на диск.

Дескрипторы отдельных процессов объединены в список, образующий таблицу процессов. Память для таблицы процессов отводится динамически в области ядра. На основании хранящейся в ней информации ОС осуществляет планирование и синхронизацию процессов. В дескрипторе прямо или косвенно (через указатели на структуры процесса) содержится информация 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 содержит 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,

информация о незавершенных операциях ввода-вывода и др.

Контекст процесса также 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.

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 сводится к следующему:

Сохранение контекста текущего потока,

Загрузка контекстов нового потока, выбранного в результате планирования,

Запуск нового потока на выполнение.

В системе, не содержащей потоки, все сказанное далее о планировании и диспетчеризации потоков может быть отнесено к процессам в целом.

Существует множество различных алгоритмов планирования потоков, преследующих различные цели и обеспечивающих разное качество мультипрограммирования. В большинстве ОС планирование осуществляется динамически (on-line), то есть решения принимаются во время работы системы на основе анализа текущей ситуации. Динамические планировщики могут гибко приспо-

сабливаться к изменяющейся ситуации и не используют никаких предположений относительно мультипрограммной смеси. Статическое планирование может использоваться в специализированных системах, в которых весь набор одновременно выполняемых задач определен заранее, например, в системах реального времени.

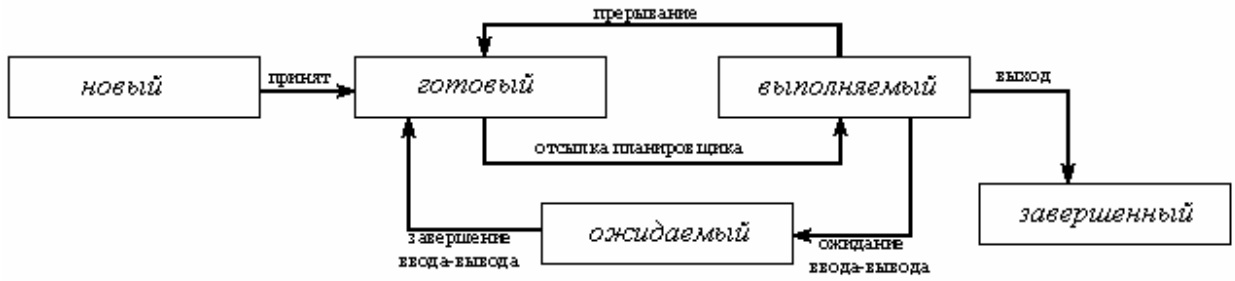
2.2. СОСТОЯНИЯ ПОТОКА

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

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

На рисунке приведена типовая диаграмма переходов для состояний процессора.



В состоянии выполнения в одно процессорном режиме может находиться не более одного потока, а в каждом из состояний ожидания и готовности – несколько. Эти потоки образуют очереди соответственно ожидающих и готовых потоков. Очереди потоков организуются путем объединения в списки описателей отдельных потоков. Каждый описатель потока содержит, по крайней мере, один указатель на другой описатель, что позволяет легко переупорядочивать потоки в очереди, включать и исключать их, переводить в другое состояние.

2.3. АЛГОРИТМЫ ПЛАНИРОВАНИЯ

С самых общих позиций все множество алгоритмов планирования можно разделить на два класса:

- *невытесняющие* алгоритмы – в которых активному потоку позволяется выполняться, пока он по собственной инициативе не отдаст

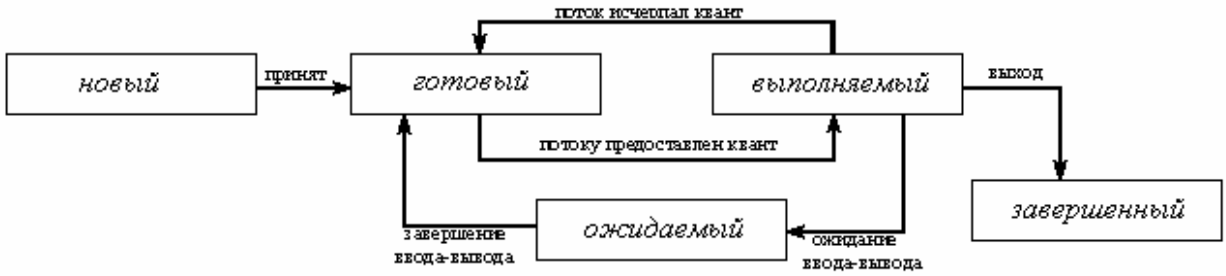
управление операционной системе для того, чтобы она выбрала другой готовый поток;

- *вытесняющие* алгоритмы – в которых решение о переключении процессора с выполнения одного потока на другой принимается ОС.

В основе многих вытесняющих алгоритмов лежит концепция *квантования*, в которой каждому потоку поочередно для выполнения предоставляется ограниченный непрерывный период процессорного времени *квант*. Смена активного потока происходит, если:

- поток завершился и покинул систему;
- произошла ошибка;
- поток перешел в состояние ожидания;
- исчерпан квант процессорного времени, отведенный данному потоку.

Поток, который исчерпал свой квант времени, переводится в состояние готовности и ожидает, когда ему будет предоставлен новый квант, а на выполнение в соответствии с определенным правилом выбирается новый поток из очереди готовых.



Кванты, выделяемые потоку, могут изменяться в разные периоды жизни потока.

Другой важной концепцией, лежащей в основе многих вытесняющих алгоритмов, является *приоритетное обслуживание. Приоритет* – это число, характеризующее степень привилегированности потока при использовании ресурсов вычислительной машины, в частности процессорного времени: чем выше приоритет, тем выше привилегии, тем меньше времени будет проводить поток в очередях.

Приоритет может выражаться целым или дробным, положительным или отрицательным значением. В некоторых ОС принято, что приоритет потока тем выше, чем больше число, обозначающее приоритет. В других системах наоборот, чем меньше число, тем выше приоритет.

Приоритеты определяются исходя из совокупности внутренних и внешних по отношению к операционной системе факторов.

Главный недостаток приоритетного планирования заключается в возможности блокирования на неопределенно долгое время низкоприоритетных процессов.

Во многих ОС предусматривается возможность изменения приоритетов в течение жизни потока. Изменение приоритета могут происходить по инициати-

ве самого потока, когда он обращается с соответствующим вызовом к операционной системе, или по инициативе пользователя – соответствующей командой. Сама ОС также может изменять приоритеты потоков в зависимости от ситуации, складывающейся в системе. Такие приоритеты называются *динамическими*, а неизменяемые – *фиксированными.*

Существует две разновидности приоритетного планирования: обслуживание с относительными приоритетами и обслуживание с абсолютными приоритетами. В обоих случаях выбор потока на выполнение из очереди готовых процессов осуществляется одинаково: по наивысшему приоритету. Однако проблема определения момента смены активного потока решается по-разному. В системах с относительными приоритетами активный поток выполняется до тех пор, пока он сам не покинет процессор, перейдя в состояние ожидания или готовности.

В ОС также широко применяются алгоритмы планирования смешанной структуры. Например, в основе планирования лежит квантование, но величина кванта и/или порядок выбора потока из очереди готовых определяется приоритетами потоков. Примером служит ОС Windows NT, в которой квантование сочетается с динамическими абсолютными приоритетами. На выполнение выбирается готовый поток с наивысшим приоритетом. Ему выделяется квант времени. Если во время выполнения в очереди готовых процессов появляется поток с более высоким приоритетом, то он вытесняет выполняемый поток. Вытесненный поток возвращается в очередь готовых, причем он становится впереди всех потоков, имеющих такой же приоритет.

3. **УПРАВЛЕНИЕ ПАМЯТЬЮ**

3.1. ФУНКЦИИ ОС ПО УПРАВЛЕНИЮ ПАМЯТЬЮ

Под памятью в этом разделе будем понимать оперативную память (ОП). В отличие от памяти жесткого диска ОП для сохранения информации требуется постоянное электропитание.

Функциями ОС по управлению памятью в мультипрограммной системе являются:

– Отслеживание свободной и занятой памяти,

– Выделение памяти процессам и ее освобождение по завершении процессов,

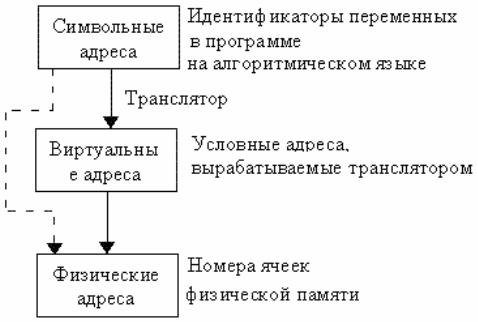
– Вытеснение кодов и данных из ОП на диск, когда размеры основной памяти не достаточны для размещения в ней всех процессов, и возвращение их в ОП, когда в них освобождается место,

– Настройка адресов программы на конкретную область физической памяти.

– Защита памяти, которая состоит в том, чтобы не позволить выполняемому процессу записывать и читать данные другого процесса.

Для идентификации переменных и программ на разных этапах жизненно-

го цикла программы используются различные имена: символьные, виртуальные и физические.



Совокупность виртуальных адресов процесса называется виртуальным адресным пространством.

Задачей ОС является отображение индивидуальных адресных пространств всех одновременно выполняемых процессов на общую физическую память. Существует два принципиально отличающихся подхода к преобразованию виртуальных адресов в физические.

1. Замена виртуальных адресов на физические выполняется один раз во время начальной загрузки программы в память.

2. Программа загружается в память в неизменном виде в виртуальных адресах, т.е. операнды инструкций и адреса переходов представлены в значениях, выработанных транслятором.

Второй способ является более гибким, т.к. программа жестко не привязана к первоначально выделенному ей участку памяти и динамическое преобразование виртуальных адресов позволяет перемещать программный код процесса в течении всего периода выполнения. Но использование перемещаемого загрузчика более экономично.

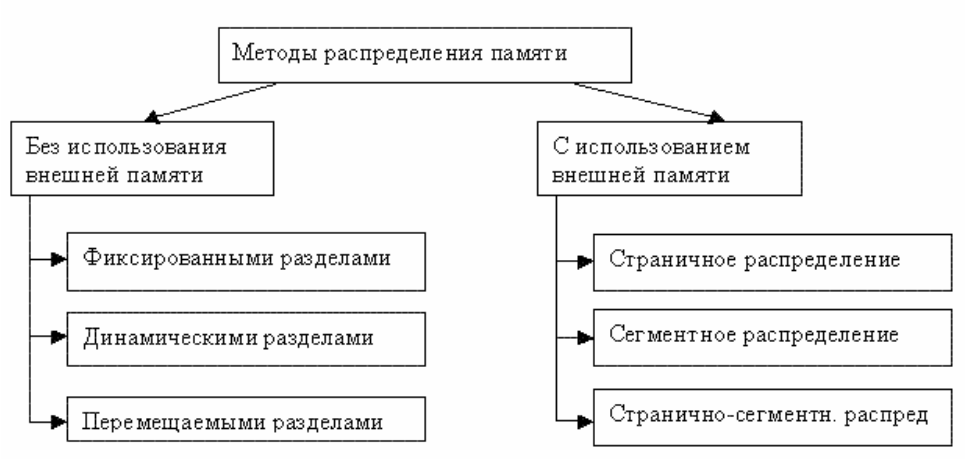
Виртуальная адресное пространство и виртуальная память – это различные механизмы, и они не обязательно реализуются в ОС одновременно.

3.2. АЛГОРИТМЫ РАСПРЕДЕЛЕНИЯ ПАМЯТИ

К базовым вопросам управления памятью относятся:

- решение о назначение каждому процессу одной непрерывной области памяти или «кусками»;
- должны ли сегменты программы, загружаемые в память, находиться на одном месте в течение всего периода выполнения процесса, или ее можно время от времени сдвигать;
- что делать, если программы не помещаются в имеющуюся память.

Разные ОС по-разному решают эти вопросы.



Некоторые из них сохранили актуальность и широко используются в современных ОС, другие представляют собой познавательный интерес, но их можно встретить до сих пор в специализированных системах.

Простейший способ управления оперативной памятью заключается в том, что память разбивается на несколько областей фиксированной величины, называемых разделами. Границы разделов не изменяются и могут быть сделаны вручную оператором во время старта.

Такой способ применялся в ранних мультипрограммных ОС. Сейчас он используется в системах реального времени, благодаря небольшим затратам на реализацию.

При распределении памяти динамическими разделами память машины заранее не делится на части. Сначала вся память, отведенная для приложений, свободна. Каждому вновь поступающему на выполнение приложению на этапе создания процесса выделяется вся необходимая ему память. Если достаточный объем памяти отсутствует, то приложение не принимается, и процесс не создается. После завершения процесса память освобождается, и на это место может быть загружен другой процесс. Таким образом, в любой момент времени ОП представляет собой случайную последовательность занятых и свободных участков произвольного размера.

Данный метод, по сравнению с предыдущим, обладает гораздо большей гибкостью. Его существенный недостаток – фрагментация памяти. Данный метод лежит в основе подсистем управления памятью многих мультипрограммных ОС 60 - 70-х годов, в частности OS/360.

Одним из способов борьбы с фрагментацией является перемещение всех занятых участков в сторону старших или младших адресов так, чтобы вся свободная память образовывала единую свободную память. В дополнение к функциям, выполняемым ОС в предыдущем методе, при использовании перемещаемых разделов она должна еще время от времени копировать содержимое разделов из одного места памяти в другое, корректируя таблицы свободных и занятых областей памяти. Эта процедура называется сжатием. Сжатие может выполняться либо при каждом завершении процесса, либо только тогда, когда вновь созданному процессу недостаточно места в свободном разделе. В первом

случае требуется меньше вычислительной работы при корректировке таблиц, а во втором – реже выполняется процедура сжатия.

Недостаток перемещаемых разделов в том, что процедура сжатия может потребовать значительного времени, что часто перевешивает преимущества данного метода.

3.2. Свопинг и виртуальная память

Необходимое условие выполнения программы – ее нахождение в ОП. Объем ОП существенно сказывается на характере протекания вычислительного процесса. Он ограничивает число одновременно выполняемых программ и размеры их виртуальных адресных пространств.

Для хорошей загрузки процессора может оказаться достаточно 3-5 вычислительных задач, но при выполнении интерактивных программ, в которых много операций ввода-вывода, разгружающих процессор, для эффективного использования процессора может потребоваться несколько десятков и даже сотен программ.

Большое количество задач, необходимое для высокой загрузки процессора, требует большого объема ОП. Для обеспечения приемлемого уровня мультипрограммирования используется метод, в котором образы некоторых процессов целиком или полностью выгружаются на диск. Как правило, это образы процессов, находящиеся в ожидании завершения операций ввода-вывода или освобождения ресурса, а также процессы в состоянии готовности, стоящие в очереди к процессору.

Такая подмена (*виртуализация*) ОП дисковой памятью позволяет повысить уровень мультипрограммирования. Она осуществляется совокупностью программных модулей ОС и аппаратных схем процессора.

Виртуализация памяти может быть осуществлена на основе двух различных подходов:

- *своппиг* - образы виртуальных процессов выгружаются на диск и в ОП целиком;
- *виртуальная память* – между ОП и диском перемещаются части (сегменты, страницы и т.п.) образов процессов.

Своппинг представляет собой частный случай виртуальной памяти. Он имеет такие недостатки как перемещение избыточной информации, замедляющее работу системы и неспособность загрузки для выполнения процесса, виртуальное адресное пространство которого превышает имеющуюся в наличие свободную память. Поэтому своппинг почти не используется в современных ОС.

Ключевой проблемой виртуальной памяти является преобразование виртуальных адресов в физические. Ее решение зависит от способа структуризации виртуального адресного пространства, который реализуется методами:

- *страничная виртуальная память* организует перемещение данных между памятью и диском страницами – частями виртуального адресного пространства, фиксированного и сравнительно небольшого размера;

- *сегментная виртуальная память* предусматривает перемещение данных сегментами – частями виртуального адресного пространства произвольного размера, полученными с учетом смыслового значения данных;
- *сегментно-страничная организация* использует двухуровневое деление: виртуальное адресное пространство делится на сегменты, а затем сегменты делятся на страницы.

Для временного хранения сегментов и страниц на диске отводится либо специальная область, либо специальный файл, который, по традиции, называют областью или файлом свопинга. Другое популярное название данной области – страничный файл. Текущий размер страничного файла является важным параметром ОС. В современных ОС он является параметром, настраиваемым администратором.

3.3.1. Страничное распределение

Виртуальное адресное пространство каждого процесса делится на части одинакового фиксированного для данной системы размера, называемые *виртуальными страницами*. В общем случае размер виртуального адресного пространства не кратен размеру страницы, поэтому последняя страница каждого процесса дополняется фиктивной областью.

Вся оперативная память машины также делится на части того же размера, называемые *физическими страницами.* Размер страницы выбирается равным степени двойки: 512, 1024, 4096 байт и т. д. При создании процесса ОС загружает в оперативную память несколько его виртуальных страниц. Копия всего виртуального адресного пространства находится на диске. Смежные виртуальные страницы необязательно находятся на смежных физических сегментах. Для каждого процесса ОС создает *таблицу страниц* – информационную структуру, содержащую записи обо всех виртуальных страницах процесса.

Сами таблицы страниц, также как и описываемые ими страницы хранятся в оперативной памяти. Адрес таблицы включается в контекст соответствующего процесса.

При каждом обращении к памяти выполняется поиск номера виртуальной страницы, содержащей требуемый адрес, затем по этому номеру определяется нужный элемент таблицы страниц, и из него извлекается описывающая страницу информация. Если же нужная виртуальная страница в данный момент выгружена на диск, то происходит так называемое *страничное прерывание*. Выполняющийся процесс переводится в состояние ожидание, и активизируется другой процесс из очереди процессов, находящихся в состоянии готовности. Параллельно программа обработки страничного прерывания находит на диске требуемую виртуальную страницу и пытается загрузить ее в ОП. Если в памяти имеется свободная физическая страница, то загрузка выполняется немедленно, если же свободных страниц нет, то на основании принятой в данной системе стратегии замещения страниц решается вопрос о том, какую страницу следует выгрузить из оперативной памяти.

Виртуальный и физический адреса представлены в виде пар – порядко-

вый номер страницы и смещение в пределах страницы.

Наиболее популярным критерием выбора страниц на выгрузку является число обращения к ней за последний период времени. Для этого ОС ведет для каждой страницы программный счетчик, обнуляемый с соответствующей периодичностью.

Интенсивность страничного обмена может быть также снижена в результате *упреждающей загрузки*, при которой при возникновении страничного прерывания в память загружается не одна страница, а сразу несколько прилегающих к ней страниц.

Выбор оптимального размера страницы является сложной оптимизационной задачей, требующей учета множества факторов. Типичный размер страницы в процессорах x86 и Pentium поддерживают страницы размером 4096 байт (4 Кбайт).

Страничное распределение памяти может быть реализовано в упрощенном варианте, без выгрузки страниц на диск. В этом случае все виртуальные страницы всех процессов постоянно находятся в ОП. В этом случае пользователю не предоставляется виртуальной памяти большого объема, но успешно происходит борьба с фрагментацией ОП.

Достоинства страничной виртуальной памяти – высокая скорость обмена, низкий уровень фрагментации, недостатки – сложность организации защиты данных, разделенных на части механически.

3.3.2. Сегментное распределение

При страничной организации виртуальное адресное пространство процесса делится на равные части механически, без учета смыслового значения данных, что не позволяет обеспечивать дифференцированный доступ к разным частям программы. Деление на «осмысленные» части делает возможным совместное использование фрагментов программ разными процессами.

Деление виртуального адресного пространства на сегменты осуществляется компилятором на основе указаний программиста или, по умолчанию, в соответствии с принятыми в системе соглашениями. Размер сегмента определяется с учетом смыслового значения содержащейся в них информации. Отдельный сегмент может представлять собой подпрограмму, массив данных и т.п. Максимальный размер сегмента определяется разрядностью виртуального адреса.

При загрузке процесса в ОП помещается только часть его сегментов, полная копия виртуального адресного пространства находится в дисковой памяти. Для каждого загружаемого сегмента ОС подыскивает непрерывный участок свободной памяти подходящего размера. Смежные в виртуальной памяти сегменты одного процесса могут занимать в ОП несмежные участки. Если во время выполнения процесса происходит обращение по виртуальному адресу, содержащемуся в сегменте, отсутствующему в памяти, происходит прерывание. ОС приостанавливает процесс, запускает на выполнение следующий процесс из очереди и параллельно организует загрузку нужного сегмента с диска. При отсутствии места ОС выбирает сегмент на выгрузку, используя критерии анало-

гичные критериям выбора страниц при страничном распределении.

Во время загрузки образа процесса в ОП система создает *таблицу сегментов* процесса.

Виртуальный адрес при сегментной организации памяти представлен парой, содержащей номер сегмента и смещение в сегменте. Физический адрес получается сложением базового адреса сегмента, который определяется по номеру сегмента из таблицы сегментов и смещения в сегменте, что замедляет процедуру преобразования адресов.

Другой недостаток данного распределения памяти – избыточность. Перемещения между памятью и диском составляет сегмент, имеющий объем больше страницы.

Главный недостаток – фрагментация, возникающая из-за непредсказуемости размеров сегмента.

3.3.3. Сегментно-страничное распределение памяти

Данный метод представляет собой комбинацию страничного и сегментного механизмов управления памятью и направлен на реализацию достоинств обоих подходов.

Виртуальное адресное пространство процесса разделено на сегменты так же как при сегментной организации памяти, что позволяет определять разные права доступа к разным частям кодов и данных программы.

Перемещение же данных между памятью и диском осуществляется не сегментами, а страницами. Для этого каждый виртуальный сегмент и физическая память делятся на страницы равного размера, что позволяет более эффективно использовать память, сократив до минимума фрагментацию.

3.3. Разделяемые сегменты памяти

Подсистема виртуальной памяти представляет удобный механизм для решения задачи совместного доступа несколько процессов к одному и тому же сегменту памяти, который в этом случае называется *разделяемой памятью*.

Для организации разделяемого сегмента при наличии подсистемы виртуальной достаточно поместить его в виртуальное адресное пространство каждого процесса, которому нужен доступ к данному сегменту, а затем настроить параметры отображения этих виртуальных сегментов так, чтобы они соответствовали одной и той же области ОП. Детали такой настройки зависят от типа используемой в ОС модели виртуальной памяти: сегментной или сегментно-страничной (чисто страничная организация делает невозможным решение рассматриваемой задачи).

При этом подразумевается, что разделяемый сегмент помещается в индивидуальную часть виртуального адресного пространства каждого процесса и описывается в каждом процессе индивидуальным дескриптором сегмента (и индивидуальными дескрипторами страниц, если используется сегментно-страничный механизм). «Попадание» же виртуальных сегментов на общую

часть ОП достигается за счет согласованной настройки операционной системой многочисленных дескрипторов для множества процессов.

Возможно и более экономичное решение данной задачи – помещение единственного разделяемого виртуального сегмента в общую часть адресного пространства процессов, которая обычно используется для модулей ОС. В этом случае настройка дескриптора сегмента (и дескрипторов страниц) выполняется только один раз, а все процессы пользуются такой настройкой и совместно используют часть ОП.

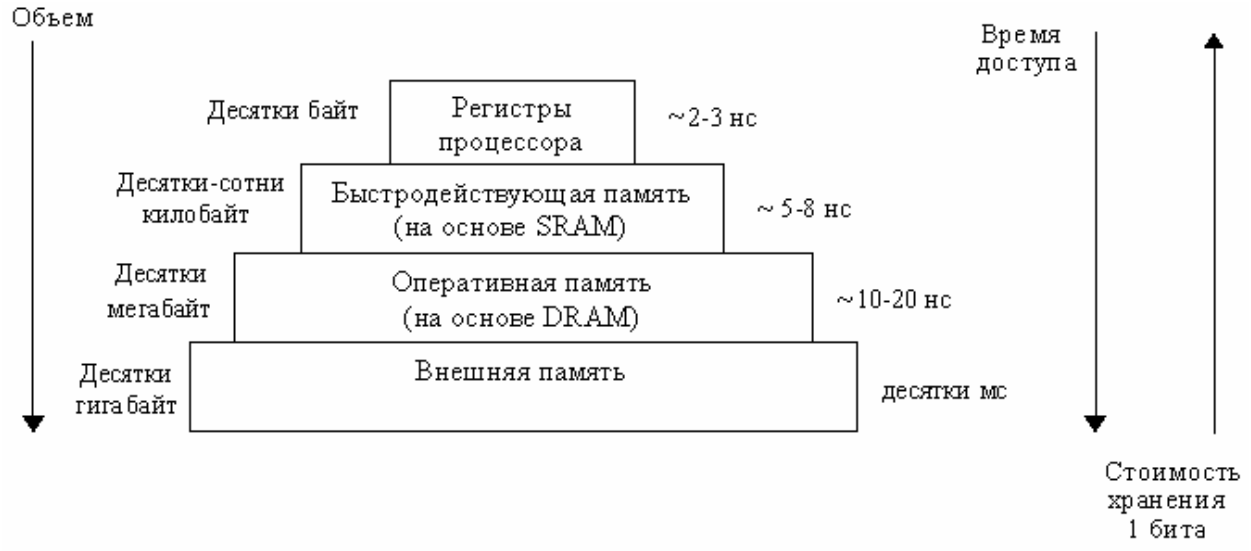
При работе с разделяемыми сегментами памяти ОС должна выполнять некоторые функции, общие для любых разделяемых между процессами ресурсов – файлов, семафоров и т.п. Эти функции состоят в поддержке схемы именования ресурсов, проверке прав доступа определенного процесса к ресурсу, а также в отслеживании количества процессов, пользующихся данным ресурсом (чтобы удалить его в случае ненадобности). Для того, чтобы отличать разделяемые сегменты памяти от индивидуальных, дескриптор сегмента должен содержать поле, имеющее два значения: shared (разделяемый) и private (индивидуальный).

ОС может создавать разделяемые сегменты, как по явному запросу, так и по умолчанию.

Разделяемые сегменты выгружаются на диск системой виртуальной памяти по тем же алгоритмам и с помощью тех же механизмов, что и индивидуальные.

3.4. Кэширование данных

Память вычислительной машины представляет собой иерархию запоминающих устройств (ЗУ), отличающихся средним временем доступа к данным, объемом и стоимостью хранения одного бита.



Из рисунка видно, что чем больше объем устройства, тем менее быстродействующим оно является. Более того, стоимость хранения данных в расчете на один бит также увеличивается с ростом быстродействия устройств. Однако

пользователю хотелось бы иметь и недорогую, и быструю память. Кэш-память предоставляет некоторое компромиссное решение данной проблемы.

Кэш-память, или просто *кэш* – способ совместного функционирования двух типов запоминающих устройств, отличающихся временем доступа и стоимостью хранения данных, который за счет динамического копирования в «быстрое» запоминающее устройство (ЗУ) наиболее часто используемой информации из «медленного» ЗУ позволяет, во-первых, уменьшить среднее время доступа к данным, во-вторых, экономить более дорогую быстродействующую память.

Неотъемлемым свойством кэш-памяти является ее прозрачность для программ и пользователей. Система не требует никакой внешней информации об интенсивности использования данных; ни пользователи, ни программы не принимают никакого участия в перемещении данных из ЗУ одного типа в ЗУ другого типа, все делается автоматически системными средствами.

Кэш-памятью часто называют не только способ организации работы двух типов ЗУ, но и одно из устройств – «быстрое» ЗУ. Оно стоит дороже и, как правило, имеет сравнительно небольшой объем. «Медленное» ЗУ будем далее называть основной памятью.

Если кэширование применяется для уменьшения среднего времени доступа к ОП, то в качестве кэша используют быстродействующую статическую память. Если кэширование используется системой ввода-вывода для ускорения доступа к данным, хранящимся на диске, роль кэш-памяти выполняют буферы в ОП, в которой оседают активно используемые данные. Виртуальную память также можно считать одним из вариантов реализации принципа кэширования данных, при котором ОП выступает в роли кэша по отношению к внешней памяти – жесткому диску. В этом случае кэширование используется не для того, чтобы уменьшить время доступа к данным, для того, чтобы заставить диск частично подменить ОП за счет перемещения временно неиспользуемых данных на диск с целью освобождения ОП для активных процессов.

Таким образом, в иерархии запоминающих устройств каждый уровень, кроме нижнего, выполняет роль кэша по отношению к нижележащему.

При каждом обращении к основной памяти по физическому адресу просматривается содержимое кэш-памяти с целью определения, не находятся ли там нужные данные. Кэш-память не является адресуемой, поэтому поиск нужных данных осуществляется по содержимому – по взятому из запроса значению поля адреса в оперативной памяти. Далее возможен один из двух вариантов:

- если данные обнаруживаются в кэш-памяти, т.е. произошло кэш-попадание, они считываются из нее, и результат передается источнику запроса;
- если нужные данные отсутствуют в кэш-памяти, т.е. произошел кэш-промах, они считываются из основной памяти, передаются источнику запроса и одновременно с этим копируются в кэш-память.

В процессе работы содержимое кэш-памяти постоянно обновляется, и данные время от времени вытесняются. Вытеснение означает либо простое объявление свободной соответствующей области кэш-памяти (сброс бита дей-

ствительности), если вытесняемые данные за время нахождения в кэше не были изменены, либо в дополнение к этому копирование данных в основную память, если они были модифицированы. Алгоритм замены данных существенно влияет на эффективность кэш-памяти. В идеале такой алгоритм должен быть максимально быстрым, чтобы не замедлять работу кэш-памяти и обеспечивать максимально возможную вероятность кэш-попаданий.

Наличие в компьютере двух копий данных – в основной памяти и в кэше – порождает проблему согласования данных. Если происходит запись в основную память по некоторому адресу, а содержимое этой ячейки находится в кэше, то в результате соответствующая запись в кэше становится недостоверной. Существует два подхода к решению данной проблемы: сквозная запись и обратная запись.

Алгоритм поиска и алгоритм замещения данных в кэше зависит от способа отображения основной памяти на кэш. Широко используются две основные схемы отображения: случайное и детерминированное.

При *случайном* – элемент оперативной памяти может быть размещен в произвольном месте кэш-памяти. Для того чтобы в дальнейшем можно было найти нужные данные в кэше, они помещаются туда вместе с адресами оперативной памяти.

Детерминированный способ предполагает, что любой элемент основной памяти всегда отображается в одно и тоже место кэш-памяти. В этом случае кэш-память разделена на строки, каждая из которых предназначается для хранения одной записи об одном элементе и имеет свой номер.

4. ФАЙЛОВАЯ СИСТЕМА

Одной из основных задач ОС является предоставление пользователю удобств пользователю 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, выводимого на экран такими утилитами, как Norton Commander или Windows Explorer, в символьных составных именах файлов, в командах работы с файлами. Базовым элементом этой модели является файл, который может характеризоваться как логической, так и физической структурой.

4.1. Понятие файловой системы

Файловая система является важной компонентой операционной системы. Файловые системы содержат, как правило, следующие **средства**:

1. **Методы доступа**. Определяют конкретную организацию доступа к данным, хранящимся в файлах.

2. **Средства управления файлами**. Обеспечивают хранение файлов, обращение к ним, коллективное использование и защиту.

3. **Средства управления внешней памятью**. Обеспечивают распределе-

ние пространства внешней памяти для размещения файлов.

4. **Средства обеспечения целостности файлов**. Призваны гарантировать сохранность информации файла. Гарантированная целостность файла означает, что в файле всегда будет храниться та информация, которая в нем должна быть, и не будет информации, которой в нем быть не должно.

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

Файловая система включает:

- совокупность всех файлов на диске;
- наборы структур данных, используемых для управления файлами, такие как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске;
- комплекс системных программных средств, реализующих различные операции над файлами, такие как создание, удаление, чтение и т.д.

4.2. Типы файлов

В разных источниках по информатике и вычислительной технике определения термина «файл» так же, как и термина «операционная система» могут варьироваться. Наиболее распространенной является формулировка, что «файл – именованный массив информации, хранящийся на носителе».

В различных файловых системах файлы могут описываться различными наборами параметров и характеристик. Основным атрибутом файла является его имя, длина которого также зависит от конкретной файловой системы. Так в файловой системе FAT (File Allocation Table), используемой в MS DOS, максимальная длина имени файла составляет 11 символов, в ОС UNIX - 14 символов, а в файловой системе NTFS (New Technology File System) для Windows NT – 255 символов. От файловой системы также зависит, какие символы могут использоваться в имени файла. Перед разработчиками файловых систем стоит проблема совместимости между ними. Но, к сожалению, пока не удалось решить проблему использования единственного имени файла в различных файловых системах. В первую очередь это связано с различием наборов разрешенных символов.

Файловой системой могут использоваться и другие атрибуты файлов, такие как:

- текущий размер файла;
- максимальный размер файла;
- длина записи;
- времена создания, последнего доступа и последнего изменения;
- владелец файла;
- создатель файла;
- информация о доступе к файлу;

- пароль для доступа к файлу;
- признак только для «чтения»;
- признак «скрытый файл»;
- признак «архивный файл»;
- признак «временный»;
- признак блокировки.

Для логической организации файлов используют каталоги. Каталог содержит файлы, объединенные по какому-либо признаку. *Каталог* – это файл, содержащий информацию о входящих в него файлах. Все каталоги, находящиеся на носителе, образуют иерархическую структуру.

В зависимости от файловой системы структура каталогов может быть древовидной, когда один файл может входить только в один каталог, и сетевой, когда один файл может входить в различные каталоги. Пример системы с древовидной структурой каталогов – система FAT. Сетевая структура, более подходящая для многопользовательских систем, реализована в UNIX.

Каталоги устанавливают соответствие между именами файлов и их характеристиками, используемыми ФС для управления файлами, например тип файла, его расположение на диске, права доступа и т.д.

Специальные файлы – фиктивные файлы, ассоциированные с устройствами ввода-вывода, которые используются для унификации механизма доступа к файлам и внешним устройствам. Они позволяют выполнять операции ввода-вывода посредством обычных команд записи в файл или вывода из файла.

4.3. Система управления файлами

Специальное системное программное обеспечение, реализующее работу с файлами по принятым спецификациям файловой системы, часто называют *системой управления файлами.* Она отвечает за создание, уничтожение, организацию, чтение, запись, модификацию и перемещение файловой информации, а также за управление доступом к файлам и за управление ресурсами, которые используются файлами.

Как правило, все современные операционные системы имеют соответствующие системы управления файлами. А некоторые - возможность работы с несколькими файловыми системами (либо с одной из нескольких, либо сразу с несколькими одновременно). В этом случае говорят о *монтируемых файловых системах* (монтируемую систему управления файлами можно установить как дополнительную), и в этом смысле они самостоятельны.

Очевидно, что система управления файлами, будучи компонентом операционной системы, не является независимой от нее, поскольку активно использует соответствующие вызовы API. С другой стороны, системы управления файлами сами дополняют API новыми вызовами. Следует заметить, что любая система управления файлами разрабатывается для работы в конкретной операционной системе. Например, файловая система FAT (File Allocation Table — таблица размещения файлов) имеет множество реализаций как система управ-

ления файлами. Система, получившая это название, была разработана для первых персональных компьютеров, называлась просто FAT. Ее разрабатывали для работы с дискетами, но некоторое время она использовалась при работе с жесткими дисками. Потом ее доработали для работы с жесткими дисками большего объема (FAT16). Это название файловой системы употребляется по отношению к подсистеме управления файлами системы MS DOS. Есть версия системы управления файлами с принципами FAT и для Windows 95/98, есть реализация для Windows NT и т. д. Таким образом, для работы с файлами, организованными в соответствии с некоторой файловой системой, для каждой операционной системы должна быть разработана соответствующая система управления файлами. И эта система управления файлами будет работать только в той операционной системе, для которой создана, но при этом обеспечит доступ к файлам, созданным с помощью системы управления файлами другой операционной системы, но работающей по тем же основным принципам файловой системы. Например, в MS DOS, OS/2, Windows 95/98/ ME, Windows NT/2000/XP, Linux, FreeBSD и других можно работать с файлами, организованными по принципам FAT.

Таким образом, термин *файловая система* определяет, прежде всего, принципы доступа к данным, организованным в файлы. Тот же термин используют и по отношению к конкретным файлам, расположенным на том или ином носителе данных. А термин *система управления файлами* следует употреблять по отношению к конкретной реализации файловой системы, то есть это — комплекс программных модулей, обеспечивающих работу с файлами в конкретной операционной системе.

4.4 Файловая система FAT

Файловая система FAT (File Allocation Table — таблица размещения файлов) получила свое название благодаря простой таблице, в которой указываются:

- непосредственно адресуемые участки логического диска, отведенные для размещения в них файлов или их фрагментов;
- свободные области дискового пространства;
- дефектные области диска (эти области содержат дефектные участки и не гарантируют чтение и запись данных без ошибок).

В файловой системе FAT дисковое пространство любого логического диска делится на две области: *системную область* и *область данных.*

Системная область логического диска создается и инициализируется при форматировании, а в последующем обновляется при работе с файловой структурой. Область данных логического диска содержит обычные файлы и файлы-каталоги; эти объекты образуют иерархию, подчиненную корневому каталогу. Элемент каталога описывает файловый объект, который может быть либо обычным файлом, либо файлом-каталогом. Область данных, в отличие от системной области, доступна через пользовательский интерфейс операционной системы. Системная область состоит из компонентов (расположенных в логическом адресном пространстве друг за другом):

- загрузочной записи (Boot Record, BR);
- зарезервированных секторов (Reserved Sectors, ResSec);
- таблицы размещения файлов (File Allocation Table, FAT);
- корневого каталога (Root Directory, RDir).

Таблица размещения файлов

Таблица размещения файлов является очень важной информационной структурой. Можно сказать, что она представляет собой адресную карту области данных, в которой описывается и состояние каждого участка области данных, и принадлежность его к тому или иному файловому объекту.

Всю область данных разбивают на *кластеры*. Кластер представляет собой один или несколько смежных секторов в логическом дисковом адресном пространстве (точнее — только в области данных). Кластер — это минимальная адресуемая единица дисковой памяти, выделяемая файлу (или некорневому каталогу). Кластеры введены для того, чтобы уменьшить количество адресуемых единиц в области данных логического диска.

Каждый файл занимает целое число кластеров. Последний кластер при этом может быть задействован не полностью, что при большом размере кластера может приводить к заметной потере дискового пространства. На дискетах кластер занимает один или два сектора, а на жестких дисках его размер зависит от объема раздела (табл.).

Логическое разбиение области данных на кластеры как совокупности секторов взамен использования одиночных секторов имеет следующий смысл:

- уменьшение размера самой таблицы FAT;
- уменьшение возможной фрагментации файлов;
- ускорение доступа к файлу, так как в несколько раз сокращается длина цепочек фрагментов дискового пространства, выделенных для него.

Поскольку файлы на диске изменяются (удаляются, перемещаются, увеличиваются или уменьшаются), то упомянутое правило выделения первого свободного кластера для новой порции данных приводит к *фрагментации* файлов, что это приводит к существенному замедлению работы с файлами.

В связи с тем, что таблица FAT используется при доступе к диску очень интенсивно, она обычно загружается в оперативную память (в буферы ввода-вывода или в кэш) и остается там настолько долго, насколько это возможно.

Корневой каталог отличается от обычного файла-каталога тем, что он помимо размещения в фиксированном месте логического диска имеет еще и фиксированное число элементов. Для каждого файла и каталога в файловой системе хранится информация.

К стандартной системе FAT (имеется в виду, прежде всего реализация FAT 16) добавились еще две разновидности, используемые в широко распространенных ОС от Microsoft (конкретно — в Windows 95 и Windows NT): VFAT (виртуальная система FAT) и система FAT32, используемая в одной из редакций ОС Windows 95 и Windows 98. Ныне файловая система FAT32 поддерживается и такими последними системами, как Windows Millennium Edition, Windows 2000 и Windows XP. Имеются реализации FAT32 и для Windows NT, и для

Linux.

Файловая система FAT32 является полностью самостоятельной 32-разрядной файловой системой и содержит многочисленные усовершенствования и дополнения по сравнению с предыдущими реализациями FAT. Самое принципиальное отличие в том, что FAT32 намного эффективнее расходует дисковое пространство. Прежде всего, кластеры в этой системе меньше, чем кластеры в предыдущих версиях. FAT32 также может перемещать корневой каталог и использовать резервную копию FAT вместо стандартной. Расширенная загрузочная запись FAT32 позволяет создавать копии критически важных структур данных; это повышает устойчивость дисков FAT32 к нарушениям структуры таблицы размещения файлов по сравнению с предыдущими версиями. Корневой каталог в FAT32 представлен в виде обычной цепочки кластеров, следовательно, он может находиться в произвольном месте диска, что снимает действовавшее ранее ограничение на размер корневого каталога (512 элементов).

Загрузочная запись для системы FAT32 несколько отличается от загрузочной записи FAT 16. Так, например, в загрузочном секторе для тома с FAT32 в блоке DPB содержатся дополнительные поля, а те поля, что находятся в привычном для системы FAT16 месте, перенесены. Поэтому операционная система, в которой есть возможность работать с файловой системой FAT16, но нет системы управления файлами, понимающей спецификации FAT32, не может читать данные с томов, отформатированных под файловую систему FAT32.

4.5. Файловая система NTFS

NTFS (New Technology File System) — файловая система новой технологии. Действительно, файловая система NTFS по сравнению FAT16 (и даже FAT32)содержит ряд значительных усовершенствований и изменений. С точки зрения пользователей файлы по-прежнему хранятся в каталогах(называемых *папками* (folders)). Однако в ней появилось много новых особенностей и возможностей.

Основные возможности файловой системы NTFS

При проектировании NTFS особое внимание было уделено надежности, механизмам ограничения доступа к файлам и каталогам, расширенной функциональности, поддержке дисков большого объема и пр. Начала разрабатываться эта система в рамках проекта OS/2 v.3, поэтому она переняла многие интересные особенности файловой системы HPFS.

Надежность

Высокопроизводительные компьютеры и системы совместного использования должны обладать повышенной надежностью, которая является ключевым элементом структуры и функционирования NTFS. Система NTFS обладает определенными средствами самовосстановления. Она поддерживает различные механизмы проверки целостности системы, включая ведение журналов транзакций, позволяющих воспроизвести файловые операции записи по специальному системному журналу.

Поскольку NTFS разрабатывалась как файловая система для серверов, для которых очень важно обеспечить бесперебойную работу без перезагрузок, в ней, как и HPFS, для повышения надежности был введен механизм аварийной замены дефектных секторов резервными.

Ограничения доступа к файлам и каталогам

Файловая система NTFS поддерживает объектную модель безопасности операционной системы Windows NT и рассматривает все тома, каталоги и файлы как самостоятельные объекты. Разрешения доступа к томам, каталогам и файла могут зависеть от учетной записи пользователя и тех групп, к которым он принадлежит. Каждый раз, при обращении к объекту файловой системы разрешения пользователя на доступ проверяются по *списку управления доступом* (ACL) для данного объекта.

Расширенная функциональность

Эта система также позволяет сжимать как отдельные файлы, так и целые каталоги. В последней, пятой, версии NTFS введена возможность шифрования хранимых файлов. В системах Windows 2000/XP в случае использования файловой системы NTFS можно включить квотирование, при котором пользователи могут хранить свои файлы только в пределах отведенной им квоты на дисковое пространство.

Поддержка дисков большого объема

Система NTFS создавалась с расчетом на работу с большими дисками. Она уже достаточно хорошо проявляет себя при работе с томами объемом 300-400 Мбайт и выше. Чем больше объем диска и чем больше на нем файлов, тем больший выигрыш мы получаем, используя NTFS вместо FAT16 или FAT32.

Структура тома с файловой системой NTFS

Одним из основных понятий, используемых при работе с NTFS, является понятие *тома* (volume). Том означает логическое дисковое пространство которое может быть воспринято как логический диск, то есть том может иметь букву (буквенный идентификатор) диска. Частным случаем тома является логический диск. Возможно также создание отказоустойчивого тома, занимающего несколько разделов, то есть поддерживается использование RAID-технологии.

Как и многие другие файловые системы, NTFS делит все полезное дисковое пространство тома на кластеры — блоки данных, адресуемые как единицы данных. Файловая система NTFS поддерживает размеры кластеров от 512 байт до 64 Кбайт; неким стандартом же считается кластер размером 2 или 4 Кбайт. К сожалению, 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). Эта зона предназначена для таблицы MFT (с учетом ее будущего роста), представляющей собой специальный файл со служебной информацией, позволяющей определять местонахождение всех остальных файлов. Запись каких-либо данных в зону MFT невозможна — она всегда остается пустой, чтобы при росте MFT по возможности не было фрагментации. Остальные 88 % тома представляют собой обычное пространство для хранения фай-

лов.

Структуру данных, называемую главной таблицей файлов, можно рассматривать как файл. В этом файле MFT хранится информация обо всех остальных файлах диска, в том числе и о самом файле MFT. Таблица MFT поделена на записи фиксированного размера в 1 Кбайт, и каждая запись соответствует какому-либо файлу (в общем смысле этого слова). Первые 16 файлов носят служебный характер и недоступны через интерфейс операционной системы — они называются *метафайлами,* причем самый первый метафайл — это сам файл MFT. Часть диска с метафайлами — единственная часть диска, имеющая строго фиксированное положение. Копия этих же 16 записей таблицы MFT (для надежности, поскольку они очень важны) хранится в середине тома. Оставшаяся часть файла MFT может располагаться, как и любой другой файл, в произвольных местах диска — восстановить его положение можно с помощью самого файла MFT. Для этого достаточно взять первую запись таблицы MFT.

Первые 16 файлов NTFS (метафайлы) являются служебными; каждый из них отвечает за какой-либо аспект работы системы. Метафайлы находятся в корневом каталоге тома NTFS. Их имена начинаются с символа «$», хотя получить какую-либо информацию о них стандартными средствами сложно.

Все файлы тома представлены в таблице MFT. За исключением собственно данных, в этой структуре хранится вся информация о файлах: имя файла, размер, положение на диске отдельных фрагментов и т. д. Если для информации не хватает одной записи MFT, то используются несколько записей, причем не обязательно последовательных. Если файл имеет не очень большой размер, тогда данные файла хранятся прямо в соответствующей записи таблицы MFT в оставшемся от служебных данных месте. Таким образом, файлы, занимающие не более сотни байтов, обычно не имеют своего «физического» воплощения в основной файловой области — все данные таких файлов хранятся прямо в таблице MFT.

Файл на томе в системе NTFS идентифицируется так называемой *файловой ссылкой* (file reference), которая представляется как 64-разрядное число.

Большинство атрибутов файла (за исключением основных) — это тоже потоки данных. Таким образом, получается, что основой файла является номер записи в таблице MFT, а все остальное, включая его потоки данных, не обязательно. Стандартные атрибуты файлов и каталогов на томе NTFS имеют фиксированные имена и коды типа.

4.6. Физическая организация файловой ситстемы

Физическая организация данных описывает правила расположения файлов на носителе. Расположение файла описывается расположением принадлежащих ему блоков. *Блоком* называется наименьшая единица данных, которой устройство ввода/вывода может обмениваться с памятью.

Простейший способ расположения файла – непрерывная последовательность блоков. Такой способ наряду со своим основным достоинством - простотой, которая позволяет адресовать файл всего лишь адресом его первого блока,

имеет ряд недостатков:

- во время создания файла системе может быть неизвестен его размер, и, следовательно, неизвестно, сколько места на носителе надо зарезервировать;
- неизбежна сильная фрагментация носителя.

Для устранения этих недостатков могут использоваться связанные блоки. В таком случае блок помимо данных содержит ссылку на следующий блок и т. д. Основным недостатком такого способа является то, что программа не может непосредственно обратиться к произвольному участку файла, и чтобы прочитать, например, последний блок, необходимо последовательно обратиться ко всем блокам файла. Кроме того, информация, хранящаяся в блоке данных, теряет свою однородность, так как содержит не только данные файла, но и служебную информацию.

Решить эти проблемы может использование связанного списка индексов, что делается, например, в MS DOS. При такой организации данных для чтения последнего блока файла достаточно просмотреть таблицу индексов. При этом сохраняется однородность данных, поскольку служебная информация сохраняется в отдельной области, располагающейся в случае жесткого диска, на внутренних дорожках, что обеспечивает быстрый доступ к ней.

Файловая система также может предоставлять некоторые оптимизирующие функции, основными из которых являются: кэширование диска и отображение файла в память.

Кэширование диска служит для ускорения доступа к наиболее часто используемым данным. При запросе данных система сначала ищет их в буфере, располагающемся на носителе с более высокой скоростью доступа. Если требуемые данные не найдены в буфере, они читаются с диска и заносятся в буфер. По мере наполнения буфера из него удаляются наиболее редко используемые данные.

Отображения файла в память заключается в создании образа файла, организованного как память, т.е. представляющего собой непрерывную последовательность байтов. Это упрощает доступ к произвольному участку файла и в целом повышает скорость выполнения основных операций с данными.

Поскольку главной функцией файловой системы является распределение пространства внешней памяти и управление ее работой, и в частности работой дисковой памяти.

5. Обзор современных ОС. Интерфейсы и основные стандарты в области системного программного обеспечения

5.1. Обзор современных ОС

Наиболее популярными являются ОС семейства Windows компании Microsoft. Это и Windows 95/98/ME, и Windows NT/2000, и новое поколение Windows XP/2003.

Следующая группа – UNIX-подобные операционные системы Linux и FreeBSD, а также QNX, OS/2.

ОС Windows 9x создавались для работы только на IBM-совместимых компьютерах и главным образом предназначены для домашнего, а не корпоративного применения. Они все поддерживают механизм автоматического подключения устройств Plug and Play, являются 32-разрядными и многозадачными. Ядро построено по макроядерной архитектуре и состоит из трех компонент: Kernel, User, GDI. Для работы с периферийными устройствами используется архитектура «универсальный драйвер – мини-драйвер», а также драйверы виртуальных устройств.

Windows NT обладает микроядерной архитектурой, аппаратной независимостью, мультипроцессорной обработкой и масштабируемостью, возможностью выполнения приложений, созданных для других ОС, высокопроизводительной файловой системой, встроенными сетевыми функциями и поддержкой распределенных вычислений, защитой информации от несанкционированного доступа.

Windows 95/98/ME, Windows NT/2000/XP поддерживают потоковые вычисления, при этом диспетчер задач работает с несколькими очередями. Однако в системах последнего поколения многозадачность реализована лучше.

Семейство UNIX-совместимых – удачная реализация простой мультипрограммной и многопользовательской ОС.

Основные преимущества:

- простота – все сложности в пользовательских программах, файловая система проста и не зависит от устройств;
- общность – одни и те же методы и механизмы используются во многих случаях: в обращение к файлам и устройствам ввода-вывода, в отношении программных и аппаратных прерываний и пр.;
- решение сложных задач комбинацией простых;
- мультитерминальность.

Центральной частью UNIX-систем является ядро, состоящее из большого количества модулей, и с точки зрения архитектуры считается монолитным. В ядре выделяют три подсистемы: управления процессами, управление файлами, управление операциями ввода-вывода между центральной частью и периферийными устройствами.

Ядро может быть перекомпилировано с учетом конкретного состава устройств компьютера и решаемых задач. Не все драйверы и системные функции могут быть включены в состав ядра. Однако основные системные функции строго стандартизированы, за счет чего достигается переносимость между разными версиями UNIX и абсолютно разным программным обеспечением.

Текущее состояние виртуального компьютера называется образом. Образ процесса состоит из образа памяти, значений регистров процессора, состояния открытых файлов, текущего состояния каталога файлов и др.

Традиционный способ взаимодействия пользователя с системой UNIX основывается на командных языках.

Linux – современная многопользовательская многозадачная свободно-распространяемая UNIX-подобная ОС. Все компоненты распространяются с лицензией на свободное копирование и установку для неограниченного числа пользователей. Согласно принятому соглашению в систему Linux каждый может внести свои изменения, но при этом должен сделать свой код открытым.

Ядро Linux создано с учетом возможностей защищенного режима 32-разрядных процессоров 80386 и 80486 фирмы Intel.

Linux имеет макроядро, содержащие три подсистемы, но в отличии от старых версий использует страничный механизм организации виртуальной памяти. Оно также использует универсальный пул памяти для пользовательских программ и дискового кэша с механизмом агрессивного кэширования. Исполняемые программы задействуют динамически связываемые библиотеки DLL.

FreeBSD также является свободно-распространяемой UNIX-подобной ОС. В отличии от Linux FreeBSD имеет своего координатора – университет Беркли, Калифорния. Внесение изменений в ОС должно согласовываться с координатором.

Сетевая операционная система реального времени QNX разработана для процессоров с архитектурой ia32. Встроенные средства позволяют обеспечивать поддержку многозадачного режима и взаимодействие параллельно выполняемых задач на разных компьютерах сети. Соответствует стандарту POSIX. Микроядро ОС QNX имеет объем в несколько десятков килобайт, содержащее механизм передачи сообщений между процессами, редиктор прерываний, блок планирования выполнения задач, сетевой интерфейс для перенаправления сообщений.

Строение и функционирование ОС OS/2 можно считать идеальным с точки зрения теории и неплохим в реализации. OS/2 практически неизменна с 1992 года, что свидетельствует о продуманности архитектуры. В ней используется плоская модель памяти, тщательное кодирование требует небольших вычислительных ресурсов, удачно реализована диспетчеризация задач, статическая форма информационных структур обеспечивает высокое быстродействие. Разнообразные сервисные функции API обеспечиваются системными библиотеками DLL. Ядро предоставляет многие базовые сервисные функции API, обеспечивает поддержку файловой системы, управление памятью, имеет диспетчер аппаратных прерываний. Драйверы виртуальных устройств обеспечивают уровень аппаратной абстракции. Драйверы физических устройств напрямую взаимодействуют с аппаратурой.

5.2. Интерфейсы и основные стандарты в области системного программного обеспечения

Под интерфейсами ОС понимают специальные интерфейсы системного и прикладного программирования (API).

В последние годы большую популярность получили графические интерфейсы GUI, которые являются частным случаем задачи управления вводом-

выводом и не относятся к функциям ядра ОС, хотя в ряде случаев разработчики ОС относят GUI к основному системному интерфейсу API.

Термин API делят на следующие направления:

- API как интерфейс высокого уровня, принадлежащий к библиотекам RTL;
- API прикладных и системных программ, входящий в состав в ОС;
- прочие интерфейсы API.

Программный интерфейс API включает не только сами функции, но и соглашения об их использовании, которые регламентируются самой ОС, архитектурой целевой вычислительной системы и системой программирования.

Существует несколько вариантов реализации функций API:

- на уровне модулей ОС;
- на уровне систем программирования;
- на уровне внешней библиотеки процедур и функций.

Возможности API оценивают с позиций эффективности выполнения функций API, широты предоставляемых возможностей, зависимости прикладной программы от архитектуры целевой вычислительной системы.

Как правило, функции API не стандартизированы. В каждом конкретном случае набор вызовов API определяется архитектурой ОС. Принимаются попытки стандартизировать некоторый набор функций.

Частным случаем попытки стандартизировать API является внутренний корпоративный стандарт компании Microsoft – WinAPI. Он включает реализации Win16, Win32s, Win32, WinCE.

Примером стандартизации API служит один из самых распространенных стандартов – POSIX.

POSIX – независимый от платформы системный интерфейс для компьютерного окружения – является стандартом IEEE. Он описывает системные интерфейсы для открытых ОС, в том числе оболочки, утилиты, инструментарии, задачи обеспечения безопасности, сетевые функции и обработку транзакций. Стандарт базируется на UNIX-системах, но допускает реализацию и в других ОС.

Этот стандарт подробно описывает систему виртуальной памяти, многозадачность и технологию переноса ОС. Он представляет собой множество стандартов POSIX.1 – POSIX.12.

VI. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ

В качестве самостоятельной работы рекомендуется знакомство с публикациями в периодических изданиях, поиск информации в Internet (например, по ссылке www:/amursu.ru/citforum) по темам предлагаемым в качестве самостоятельного изучения и подготовки лабораторных работ.

Темы для самостоятельного изучения:

Физическая организация файловых систем FAT16, FAT32, NTFS
Динамические, последовательные и параллельные структуры программ
Сохранность и защита программных средств

К видам самостоятельной работы также относится подготовка к лабораторным работам, составление отчетов, разбор материалов лекций во время семестра.

Самостоятельная проверка полученных знаний может осуществляться на основе тестов, предлагаемых в рабочей программе.

Контроль за выполнением самостоятельной работы осуществляется на экзамене.

VII. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ МЕЖСЕССИОННОГО КОНТРОЛЯ ЗНАНИЙ

Межсессионный контроль осуществляется на основе выполнения домашних заданий, лабораторных работ, промежуточных тестов.

По итогам выполнения перечисленных видов работ в сроки, установленные деканатом (как правило, на 6-ой и 12-ой неделе семестра) преподавателем выставляется аттестационная оценка по пятибалльной системе.

VIII. КОМПЛЕКТЫ ЭКЗАМЕНАЦИОННЫХ БИЛЕТОВ

Билет № 1

1. Составные части операционной системы
2. Сегментное распределение памяти
3. Составить командный файл, продемонстрировав его пошаговое выполнение и выполнив в нем вызов другого командного файла, задание списка каталогов, в котором производится поиск запускаемого файла.

Билет № 2

1. Цели и задачи файловой системы
2. Мультипрограммирование с перемещаемыми разделами
3. Составить командный файл, продемонстрировав в нем копирование нескольких файлов в диалоговом режиме и использование комментариев.

Билет № 3

1.Ядро и вспомогательные модули ОС
2. Развитие операционных систем в 80-е годы XX века
3. Составить диалоговый командный файл, продемонстрировав выполнение команд для набора файлов и вывод сообщений при его выполнении

Билет № 4

1. Сохранность и защита программных средств
2. Свопинг и виртуальная память
3. Составить собственный командный файл, продемонстрировав его пошаговое выполнение и приостановку его выполнения

Билет № 5

1. Задачи операционной системы по управлению файлами и устройствами
2. Классификация операционных систем
3. В командном файле выполнить 3 произвольные команды MS-DOS, осуществляя их выбор в меню.

Билет № 6

1. Понятие файловой системы
2. Распределение памяти динамическими разделами

3. В файле autoexec.bat выполнить загрузку программы кэширования диска и указать каталог, в котором будут создаваться временные файлы, продемонстрируйте использование диалогового режима.

Билет № 7

1. Файловые системы FAT16, FAT32
2. Алгоритмы планирования, основанные на квантовании
3. Создать командный файл, работающий в диалоговом режиме и выполняющий ряд команд ОС MS-DOS

Билет № 8

1. Первые операционные системы, их развитие в 1965-75 гг
2. Странично-сегментное распределение памяти
3. Создать командный файл, работающий в диалоговом режиме и выполняющий ряд команд ОС MS-DOS

Билет № 9

1. Дескриптор файла
2. Алгоритмы планирование, основанные на приоритетах
3. Создайте файлы конфигураций, продемонстрировав в них работу на основе меню. Используйте комментарии и вывод сообщений на экран.

Билет № 10

1. Понятия «процесс» и «поток». Очередь потоков
2. Иерархия запоминающих устройств
3. Предусмотреть автоматический запуск программной оболочки WinCom при загрузки компьютера, и задание размеров внутренних структур DOS, таких как число буферов для операций ввода-вывода с дисков и числа одновременно открываемых файлов.

Билет № 11

1. Последовательные и параллельные структуры программ
2. Разделяемые сегменты памяти
3. Изменить файл config.sys таким образом, чтобы по выбору проходила автоматическая загрузка MS-DOS или Windows по выбору пользователя. Используйте автоматическую загрузку MS-DOS по-умолчанию через 5-ти секундный интервал

Билет № 12

1. Понятия «прерывание», «вектор прерываний», «обработчик прерываний»
2. Динамические структуры программ
3. Предусмотреть автоматический запуск программной оболочки WinCom при загрузки компьютера, и задание размеров внутренних структур DOS, таких как число буферов для операций ввода-вывода с дисков и числа одновременно открываемых файлов.

Билет № 13

1. Механизм обработки прерывания
2. Страничное распределение памяти
3. Создайте командный файл с пошаговым выполнением команд MS-DOS

Билет № 14

1. Кэш-память, принцип действия
2. Многослойная архитектура ОС
3. Изменить файл config.sys таким образом, чтобы по выбору проходила автоматическая загрузка MS-DOS или Windows по выбору пользователя. Используйте автоматическую загрузку MS-DOS по-умолчанию через 10-ти секундный интервал

Билет № 15

1. Понятие «свопинг»
2. Страничное распределение памяти
3. Дополните, имеющиеся на компьютере файлы конфигураций автоматической загрузкой MS-DOS, задав число одновременно открываемых файлов равное 30 и последней буквы дисковода – C.

Билет № 16

1. Классификация методов распределения памяти
2. Стандартные сервисные программы
3. Создайте командный файл с пошаговым выполнением команд MS-DOS

Билет № 17

1. Идентификация переменных и программ, типы адресов
2. Файловая система NTFS
3. Продемонстрировать в командном файле autoexec.bat запуск программы кэширования диска, установку списка каталогов, в которых производится поиск программ и дублирование выполнения команд на экране

Билет № 19

1. Операционные системы и компьютерные сети
2. Микроядерная архитектура файловой системы
3. Создать командный файл, выполняющий команды MS-DOS в диалоговом режиме

Билет № 20

1. Классификация вирусов
2. Основные стандарты в области системного программного обеспечения
3. Создайте файлы конфигураций, продемонстрировав в них работу на основе меню. Используйте комментарии и вывод сообщений на экран.

Билет № 21

1. Назначение и функции ОС
2. Операционные системы семейства Windows
3. Продемонстрировать все способы обновления антивирусных баз антивирусной программы

Билет № 22

1. Требование, предъявляемые к современным ОС
2. Состояния потока
3. Создайте командный файл с пошаговым выполнением команд MS-DOS

Билет № 23

1. Распределение памяти фиксированными разделами
2. Семейство UNIX-подобных операционных систем
3. Предусмотреть автоматический запуск программной оболочки WinCom при загрузки компьютера, и задание размеров внутренних структур DOS, таких как число буферов для операций ввода-вывода с дисков и числа одновременно открываемых файлов.

Билет №24

1. Интерфейсы операционных систем
2. Вытесняющие и невытесняющие алгоритмы распределения памяти
3. Создать командный файл, работающий в диалоговом режиме и выполняющий ряд команд ОС MS-DOS

IX. КАРТА ОБЕСПЕЧЕННОСТИ ДИСЦИПЛИНЫ КАДРАМИ ПРОФЕССОРСКО-ПРЕПОДАВАТЕЛЬСКОГО СОСТАВА

Все виды занятий по дисциплине ведет кандидат технических наук, доцент Галаган Т.А.

СОДЕРЖАНИЕ

Татьяна Алексеевна Галаган,
доцент кафедры ИиУС АмГУ

Учебно-методический комплекс по дисциплине «Операционные системы» для спец. 230201 – Информационные системы и технологии

Изд-во АмГУ. Подписано к печати Формат 60x84/16. Усл. печ. л.
Тираж Заказ