

Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

А. В. Воруев, П. Л. Чечет

**ОПЕРАЦИОННЫЕ СИСТЕМЫ И СЕТИ:
СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

Рекомендовано УМО по образованию
в области информатики и радиоэлектроники
для специальности 1-53 01 02
«Автоматизированные системы обработки информации»
в качестве учебно-методического пособия

Гомель
ГГУ им. Ф. Скорины
2016

УДК 004.45(075.8)
ББК 32.972.11я73
В75

Рецензенты:

кандидат технических наук В. В. Старченко;
кафедра программного обеспечения информационных технологий
Белорусского государственного университета
информатики и радиоэлектроники

Рекомендовано к изданию научно-методическим советом
учреждения образования
«Гомельский государственный университет
имени Франциска Скорины»

Воруев, А. В.

В75 Операционные системы и сети : системное программное
обеспечение : учебно-методическое пособие / А. В. Воруев,
П. Л. Чечет ; М-во образования Республики Беларусь,
Гомельский гос. ун-т им. Ф. Скорины. – Гомель: ГГУ
им. Ф. Скорины, 2016. – 131 с.
ISBN 978-985-577-238-6

В учебно-методическом пособии, актуальность которого обусловлена широким внедрением вычислительной техники во все сферы производственной и общественной деятельности, даются определения и классификация вычислительных систем и программного обеспечения; рассматриваются интерфейсы операционных систем, их основные задачи, виртуализация и конфигурирование; излагаются вопросы функционирования и объединения компьютерных сетей и систем.

Издание адресовано студентам учреждений высшего образования специальности 1-53 01 02 .

УДК 004.45(075.8)
ББК 32.972.11я73

ISBN 978-985-577-238-6

© Воруев А. В., Чечет П. Л., 2016
© Учреждение образования «Гомельский
государственный университет
имени Франциска Скорины, 2016

Оглавление

ВВЕДЕНИЕ	6
1 СОСТАВ И КЛАССИФИКАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ.....	7
1.1 Понятие компьютерной системы, масштабы компьютерных систем.....	7
1.2 Назначение компьютерных систем и профилирование по классам задач.....	8
1.3 Шинная архитектура микропроцессорных систем.....	9
1.4 Режимы работы микропроцессорной системы.....	11
1.5 Понятие прерывания и их обработка.....	14
1.6 Управление вводом – выводом.....	16
1.7 Драйверы устройств.....	19
1.8 Стационарные вычислительные системы.....	20
1.9 Мобильные и портативные вычислительные системы.....	21
1.10 Бездисковые станции и рекламные планшеты.....	23
1.11 Банкоматы, инфокиоски и фотокиоски.....	25
1.12 Использование гибридных телефонно-вычислительных устройств.....	27
2 ОПРЕДЕЛЕНИЕ И КЛАССИФИКАЦИЯ СИСТЕМНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	31
2.1 Различные определения ОС.....	31
2.2 Функции ОС.....	33
2.3 Категории пользователей ОС.....	35
2.4 Логические уровни общения человек – ЭВМ.....	37
2.5 Мультипрограммный режим работы ОС.....	38
2.6 Описание различных режимов работы ОС.....	40
2.7 Однозадачные и многозадачные операционные системы.....	41
2.8 Однопользовательские и многопользовательские ОС.....	43
2.9 Основные характеристики и особенности работы ОС MS-DOS и управление памятью в MS-DOS.....	44
2.10 Описание особенностей работы ОС Windows.....	46
2.11 ОС семейства Unix и их особенности работы.....	47
3 ИНТЕРФЕЙСЫ ОПЕРАЦИОННЫХ СИСТЕМ.....	50
3.1 Понятие интерфейса и его основные виды.....	50
3.2 Пакетный режим работы ОС.....	51
3.3 Описание простого графического интерфейса.....	52

3.4	Описание WIMP интерфейса.....	53
3.5	Описание SILK интерфейса и необходимое оборудование для его реализации	55
3.6	Описание тактильного (touch) интерфейса и необходимое оборудование для его реализации	57
3.7	Описание мимического и семантического интерфейсов и необходимое оборудование для их реализации	60
3.8	Виды мобильных устройств и типы их интерфейсов.....	61
3.9	Командный режим ОС Linux.....	63
4	ОСНОВНЫЕ ЗАДАЧИ ОПЕРАЦИОННЫХ СИСТЕМ.....	66
4.1	Понятие процесса и операции над процессами.	66
4.2	Алгоритмы планирования процессов.....	67
4.3	Вытесняющие и невытесняющие алгоритмы планирования.....	69
4.4	Методы решения проблемы синхронизации	70
4.5	Понятие нити и основные отличия от процесса.....	72
4.6	Контекст и дескриптор процесса.....	73
4.7	Классификация методов распределения данных.....	75
4.8	Распределение памяти фиксированными разделами	77
4.9	Распределение памяти разделами переменной величины (динамически).....	78
4.10	Распределение памяти перемещаемыми разделами.....	80
4.11	Понятие виртуальной памяти	81
4.12	Страничное распределение.....	82
4.13	Сегментное распределение памяти	86
4.14	Сегментно-страничное распределение памяти.....	88
4.15	Борьба с фрагментацией памяти	89
4.16	Управление программными ресурсами	92
5	ВИРТУАЛИЗАЦИЯ ОПЕРАЦИОННЫХ СИСТЕМ.....	94
5.1	Концепция виртуальной машины.....	94
5.2	Концепция виртуального ресурса	96
5.3	Классификация систем управления образами виртуальных машин.....	97
6	КОМПЬЮТЕРНЫЕ СЕТИ	99
6.1	Способы и цели объединения компьютерных сетей и систем.....	99
6.2	Понятие информационного потока в компьютерной сети.....	101
6.3	Выбор эффективной версии межсетевого экрана для реализации информационной защиты	103
6.4	Обмен данными мобильных устройств с ПК.....	105
6.5	Защита межсетевого трафика, межсетевые экраны	107

6.6	Применение фильтров доступа к внешним сетевым ресурсам	109
6.7	Использование Proxy-серверов.....	110
7	ФАЙЛОВЫЕ СИСТЕМЫ.....	112
7.1	Файловая система и примеры распространенных файловых систем.....	112
7.2	Имена файлов в различных файловых системах	114
7.3	Атрибуты файлов и права доступа к файлу.....	115
7.4	Файловые системы ОС семейства Unix	117
7.5	Особенности работы файловой системы NTFS	119
8	КОНФИГУРИРОВАНИЕ ОПЕРАЦИОННЫХ СИСТЕМ.....	121
8.1	Настройка файла autoexec.bat и config.sys в ОС MS-DOS	121
8.2	Автоматическое обслуживание настроек ОС	122
8.3	Антивирусная профилактика средств мобильной связи	124
8.4	Понятие групповой политики в сетях Microsoft.....	128
	ЛИТЕРАТУРА	130

Введение

Изучение дисциплины «Системное программное обеспечение» предусмотрено учебным планом подготовки специалистов специальности 1-53 01 02 – «Автоматизированные системы обработки информации» (ОСРБ 1-53 01 02-2007).

Актуальность изучения дисциплины обусловлена широким внедрением вычислительной техники во все сферы производственной и общественной деятельности.

В настоящее время в связи с интеграцией корпоративных сетей передачи данных все более остро встает проблема управления распределенными гетерогенными сетями, состоящими из множества локальных сетей, функционирующих на основе различных стандартов и протоколов. Цель дисциплины заключается в изучении программного обеспечения, предназначенного для интегрированного и сетевого управления компьютерными системами, что требует решения целого ряда задач. В их число входят:

- традиционные задачи сетевого управления (управление конфигурацией, управление производительностью, управление сбоями, управление безопасностью, учет использования ресурсов);

- управление распределенными приложениями в гетерогенных сетях;

- мониторинг текущего состояния системно-технического обеспечения организации (ведение визуализированной базы данных, содержащей полную информацию как о технических, так и об учетных параметрах всего технического и программного обеспечения, имеющегося в той или иной организации);

- поддержка принятия решений по модернизации технического и программного обеспечения с учетом текущего состояния технического прогресса, информации о производителях и поставщиках технических и программных средств и о сравнительных характеристиках этих продуктов;

- управление модернизацией (контроль и управление установкой нового технического и программного обеспечения, включая оптимизацию этого процесса);

- моделирование работы существующих сетей (включая анализ нагрузок на отдельные их участки и поддержку принятия решений по перепланированию).

1 СОСТАВ И КЛАССИФИКАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

1.1 Понятие компьютерной системы, масштабы компьютерных систем

Под термином «*компьютерные системы*» подразумевают непосредственно компьютер с установленным на него системным и прикладным программным обеспечением, а также электронные носители данных, локальные и глобальные компьютерные сети [17].

Выделяют следующие четыре базовые характеристики компьютерных систем:

- отношение стоимость/производительность;
- надежность и отказоустойчивость;
- масштабируемость;
- совместимость и мобильность программного обеспечения.

Составляющие компьютерной системы могут выполнять 5 основных функций (одну или несколько сразу):

- получение информации из внешних источников;
- выдачу информации;
- хранение информации;
- передачу информации;
- обработку информации.

Вычислительные системы классифицируются:

- по назначению (универсальные, специализированные);
- по типу вычислителя (многопроцессорные, многомашинные);
- по характеру устройств (однородные (гомогенные), неоднородные (гетерогенные));
- по территории (совмещенные, разобщенные);
- по управлению (централизованные, децентрализованные, смешанные);
- по режиму работы (оперативные (реальный масштаб времени), неоперативные);
- по соединению элементов (шинная схема, канальная схема, коммутационная схема).

Современная компьютерная система содержит центральный процессор, первичное и вторичное устройства хранения данных (память), устройства ввода и вывода, а также коммуникационные устройства (рисунок 1).

Центральный процессор обрабатывает данные и управляет другими устройствами компьютера. Первичная память хранит выполняющиеся в данный момент программы и обрабатываемые данные, а вторичная память – программы и данные для дальнейшего использования. Устройства ввода преобразуют данные и инструкции в форму, удобную для обработки в компьютере, а устройства вывода представляют информацию, обработанную компьютером, в виде, удобном для человеческого восприятия. Коммуникационные устройства управляют приемом и передачей данных в локальных и глобальных сетях.



Рисунок 1 – Компоненты аппаратного обеспечения компьютера

1.2 Назначение компьютерных систем и профилирование по классам задач

«Компьютерная система» – это непосредственно компьютер, с установленным на него системным и прикладным программным обеспечением, а также электронные носители данных, локальные и глобальные компьютерные сети [8].

Компьютером называется техническая система, предназначенная для автоматизации процесса обработки информации и вычислений на основе принципа программного управления.

По принципу действия вычислительные машины делятся на три больших класса: аналоговые (АВМ), цифровые (ЦВМ) и гибридные (ГВМ). Критерием деления вычислительных машин на эти три класса является форма представления информации, с которой они работают.

Цифровые вычислительные машины (ЦВМ) – вычислительные машины дискретного действия, работают с информацией, представленной в дискретной, а точнее, в цифровой форме.

Аналоговые вычислительные машины (АВМ) – вычислительные машины непрерывного действия; работают с информацией, представленной в непрерывной (аналоговой) форме, то есть в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения). На АВМ наиболее эффективно решать математические задачи, содержащие дифференциальные уравнения, не требующие сложной логики.

Гибридные вычислительные машины (ГВМ) – вычислительные машины комбинированного действия, работают с информацией, представленной и в цифровой, и в аналоговой форме; они совмещают в себе достоинства АВМ и ЦВМ. ГВМ целесообразно использовать для решения задач управления сложными быстродействующими техническими комплексами.

Наиболее широкое применение получили ЦВМ с электрическим представлением дискретной информации – электронные цифровые вычислительные машины, обычно называемые просто электронными вычислительными машинами (ЭВМ) или компьютерами, без упоминания об их цифровом характере.

1.3 Шинная архитектура микропроцессорных систем

Процессором называется устройство переработки информации.

Микропроцессор – это процессор, выполненный на одной или нескольких больших интегральных схемах.

Микропроцессорной системой называется система, центральным элементом которой является микропроцессор.

Существует два типа архитектуры микропроцессорных систем [7]:

1 Архитектура с общей, единой шиной для данных и команд (одношинная, или принстонская, фон-неймановая). В составе системы присутствует одна общая память, как для данных, так и для команд (рисунок 2).

2 Архитектура с отдельными шинами данных и команд (двухшинная, или гарвардская). Она предполагает наличие в системе отдельной памяти для данных и отдельной памяти для команд (рисунок 3).

Архитектура с общей шиной распространена гораздо больше, она применяется, например, в персональных компьютерах и в сложных

микрокомпьютерах. Архитектура с отдельными шинами применяется в основном в однокристальных микроконтроллерах.



Рисунок 2 – Архитектура с общей шиной данных и команд

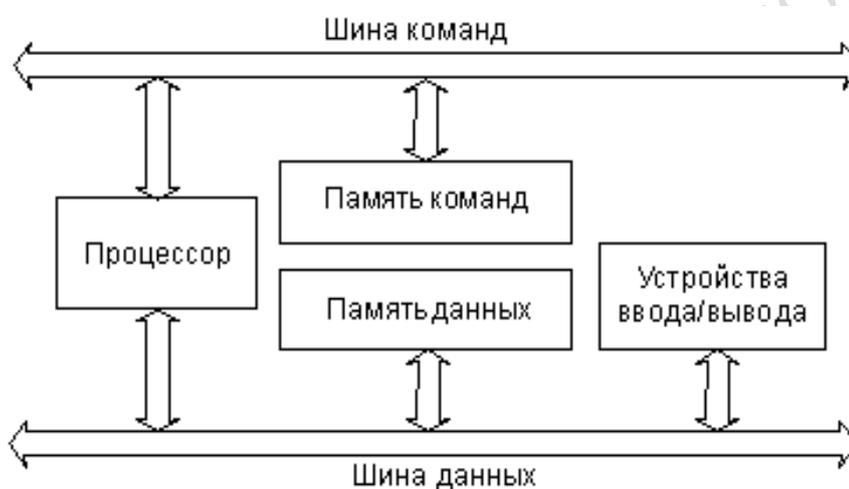


Рисунок 3 – Архитектура с отдельными шинами данных и команд

Архитектура с общей шиной проще, она не требует от процессора одновременного обслуживания двух шин, контроля обмена по двум шинам сразу. Наличие единой памяти данных и команд позволяет гибко распределять ее объем между кодами данных и команд.

Архитектура с отдельными шинами данных и команд сложнее, она заставляет процессор работать одновременно с двумя потоками кодов, обслуживать обмен по двум шинам одновременно. Программа может размещаться только в памяти команд, данные – только в памяти данных. Такая узкая специализация ограничивает круг задач, решаемых системой, так как не дает возможности гибкого перераспределения памяти. Память данных и память команд в этом случае имеют не слишком большой объем, поэтому применение систем с данной архитектурой ограничиваются обычно не слишком сложными задачами.

1.4 Режимы работы микропроцессорной системы

Микропроцессорной системой называется система, центральным элементом которой является микропроцессор [7].

Практически любая развитая микропроцессорная система (в том числе и компьютер) поддерживает три основных режима обмена по магистрали:

- программный обмен информацией;
- обмен с использованием прерываний (Interrupts);
- обмен с использованием прямого доступа к памяти (ПДП, DMA Direct Memory Access).

Программный обмен информацией является основным. Он предусмотрен всегда, без него невозможны другие режимы обмена. В этом режиме процессор является единоличным хозяином системной магистрали. Все операции (циклы) обмена информацией в данном случае инициируются только процессором, все они выполняются строго в порядке, предписанном исполняемой программой.

Процессор читает (выбирает) из памяти коды команд и исполняет их, читая данные из памяти или из устройства ввода/вывода, обрабатывая их, записывая данные в память или передавая их в устройство ввода/вывода. Путь процессора по программе может быть линейным, циклическим, может содержать переходы (прыжки), но он всегда непрерывен и полностью находится под контролем процессора. Ни на какие внешние события, не связанные с программой, процессор не реагирует (рисунок 4). Все сигналы на магистрали в данном случае контролируются процессором.



Рисунок 4 – Программный обмен информацией

Обмен по прерываниям (рисунок 5) используется тогда, когда необходима реакция микропроцессорной системы на какое-то внешнее событие, на приход внешнего сигнала. В случае компьютера внешним событием может быть, например, нажатие на клавишу клавиатуры или приход по локальной сети пакета данных. Компьютер должен реагировать на это, соответственно, выводом символа на экран или же чтением и обработкой принятого по сети пакета.

В общем случае организовать реакцию на внешнее событие можно тремя различными путями:

- с помощью постоянного программного контроля факта наступления события (так называемый метод опроса флага или polling);
- с помощью прерывания, то есть насильственного перевода процессора с выполнения текущей программы на выполнение экстренно необходимой программы;
- с помощью прямого доступа к памяти, то есть без участия процессора при его отключении от системной магистрали.

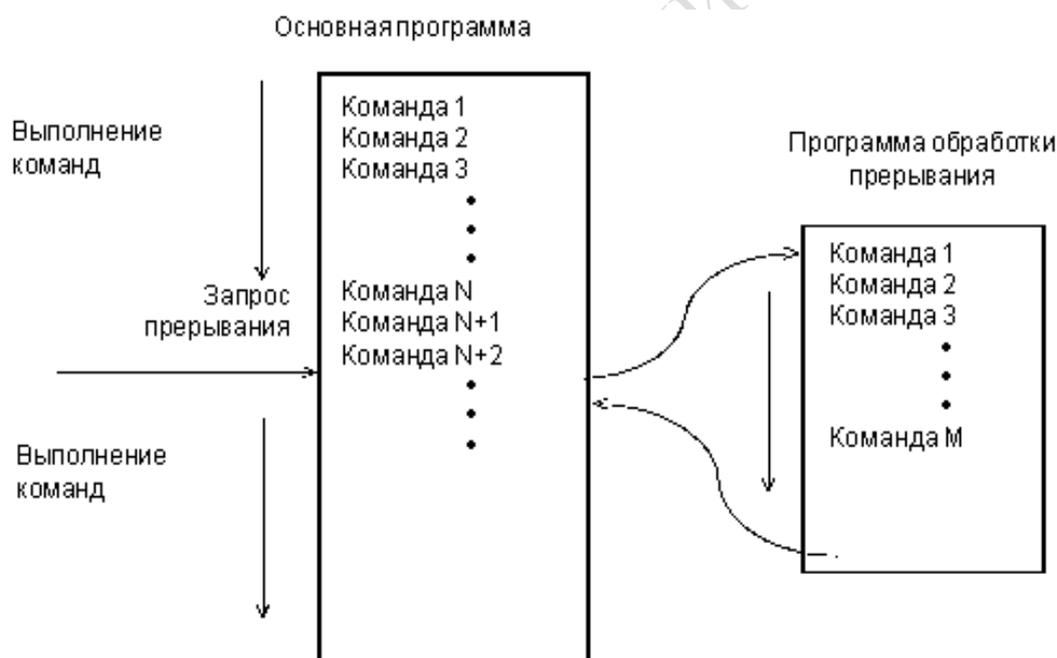


Рисунок 5 – Обслуживание прерывания

Прямой доступ к памяти (ПДП, DMA) – обмен по системной шине идет без участия процессора. Внешнее устройство, требующее обслуживания, сигнализирует процессору, что режим ПДП необходим, в ответ на это процессор заканчивает выполнение текущей команды и отключается от всех машин, сигнализируя запрошившему устройству, что обмен в режиме ПДП можно начинать.

Операция ПДП сводится к пересылке информации из устройства ввода/вывода в память или же из памяти в устройство ввода/вывода. Когда пересылка информации будет закончена, процессор вновь возвращается к прерванной программе, продолжая ее с той точки, где его прервали (рисунок 6). Это похоже на режим обслуживания прерываний, но в данном случае процессор не участвует в обмене. Как и в случае прерываний, реакция на внешнее событие при ПДП существенно медленнее, чем при программном режиме.

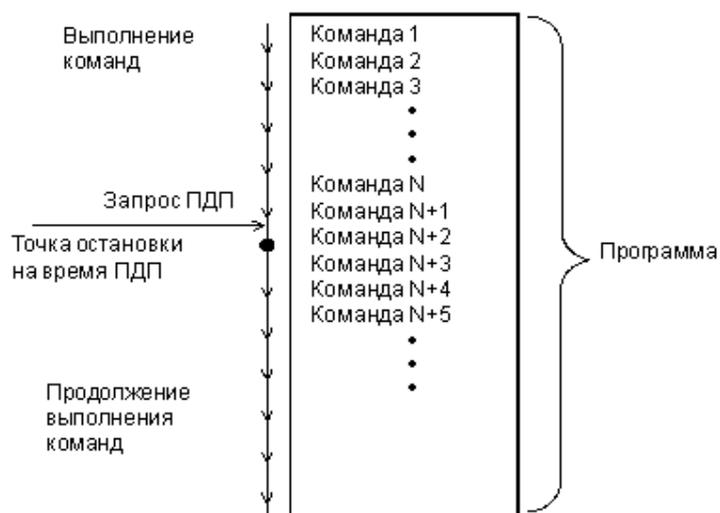


Рисунок 6 – Обслуживание ПДП

В этом случае требуется введение в систему дополнительного устройства (контроллера ПДП), которое будет осуществлять полноценный обмен по системной магистрали без всякого участия процессора. Причем процессор предварительно должен сообщить этому контроллеру ПДП, откуда ему следует брать информацию и/или куда ее помещать. Контроллер ПДП может считаться специализированным процессором, который отличается тем, что сам не участвует в обмене, не принимает в себя информацию и не выдает ее (рисунок 7).

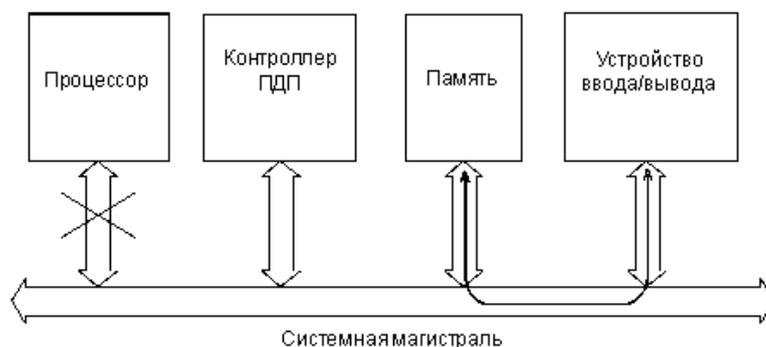


Рисунок 7 – Информационные потоки в режиме ПДП

1.5 Понятие прерывания и их обработка

Прерывания являются основной движущей силой любой ОС[11]. Периодические прерывания от таймера вызывают смену процессов в мультипрограммной ОС, а прерывания от устройств ввода-вывода управляют потоками данных, которыми ВС обменивается с внешним миром.

Система прерываний переводит процессор на выполнение потока команд, отличного от того, который выполнялся до сих пор, с последующим возвратом к исходному коду.

В зависимости от источника прерывания делятся на три больших класса:

- внешние;
- внутренние;
- программные.

Внешние (аппаратные) прерывания могут возникать в результате действий пользователя или оператора за терминалом или же в результате поступления сигналов от аппаратных устройств – сигналов завершения операций ввода-вывода, вырабатываемых контроллерами внешних устройств компьютера, такими как принтер или накопитель на жестких дисках, или же сигналов от датчиков управляемых компьютером технических объектов.

Внутренние прерывания, называемые также исключениями, происходят синхронно выполнению программы при появлении аварийной ситуации в ходе исполнения некоторой инструкции программы. Примерами исключений являются деление на нуль, ошибки защиты памяти и т. д.

Программные прерывания, по сути, не являются «истинными» прерываниями. Программное прерывание возникает при выполнении особой команды процессора, выполнение которой имитирует прерывание.

Прерываниям приписывается приоритет, с помощью которого они ранжируются по степени важности и срочности.

Прерывания обычно обрабатываются модулями ОС, так как действия, выполняемые по прерыванию, относятся к управлению разделяемыми ресурсами вычислительной системы. Процедуры, вызываемые по прерываниям, обычно называют обработчиками прерываний, или процедурами обслуживания прерываний. Аппаратные прерывания обрабатываются драйверами соответствующих внешних устройств, исключения – специальными модулями ядра,

а программные прерывания – процедурами ОС, обслуживающими системные вызовы. Кроме этих модулей, в ОС может находиться так называемый диспетчер прерываний, который координирует работу отдельных обработчиков прерываний.

В зависимости от источника возникновения сигнала прерывания выделяют следующие виды прерываний:

1) асинхронные или внешние (аппаратные) – события, которые исходят от внешних источников и могут произойти в любой произвольный момент. Факт возникновения в системе такого прерывания трактуется как запрос на прерывание;

2) синхронные или внутренние – события в самом процессоре как результат нарушения каких-то условий при исполнении машинного кода;

3) программные – инициируются исполнением специальной инструкции в коде программы. Используются для обращения к функциям встроенного ПО, драйверов и ОС.

В зависимости от возможности запрета внешние прерывания делятся на маскируемые (прерывания, которые можно запрещать установкой соответствующих битов в регистре маскирования прерываний; немаскируемые (обрабатываются всегда, независимо от запретов на другие прерывания).

Существуют два основных способа выполнения прерывания:

– векторный (vectored), когда в процессор передается номер вызываемой процедуры обработки прерывания;

– опрашиваемый (polled), когда процессор вынужден последовательно опрашивать потенциальные источники запроса прерывания.

Устройствам, которые используют векторные прерывания, назначается вектор прерываний. Он представляет собой электрический сигнал и несет в себе информацию о закреплённом за данным устройством номере, который идентифицирует соответствующий обработчик прерываний. Процессор использует этот вектор для нахождения обработчика данного прерывания.

При использовании опрашиваемых прерываний процессор получает от устройства информацию об уровне приоритета прерывания. Процессор должен определить, какое устройство из тех, которые связаны с данным уровнем прерываний, действительно запросило прерывание. Для этого вызываются все обработчики прерываний, для данного уровня приоритета, пока один из уровней обработчика не подтвердит, что прерывание пришло от устройства, обслуживаемого им.

Для некоторых аппаратных платформ могут сочетаться оба типа прерываний (рисунок 8).

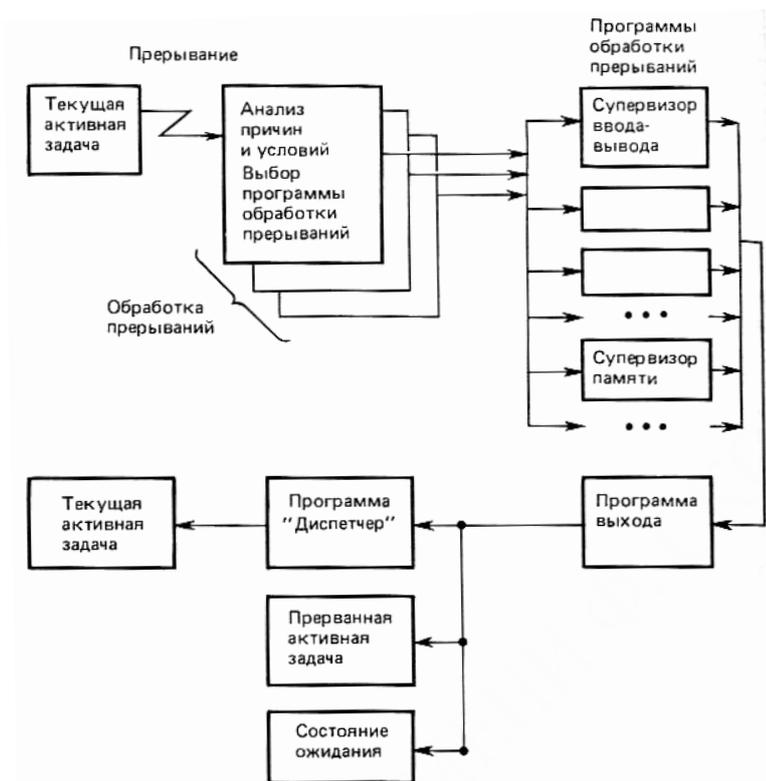


Рисунок 8 – Общая схема обработки прерываний

1.6 Управление вводом – выводом

Одной из главных задач ОС является обеспечение обмена данными между приложениями и периферийными устройствами компьютера [11]. В современной ОС функции обмена данными с периферийными устройствами выполняет подсистема ввода-вывода. Основными компонентами подсистемы ввода-вывода являются драйверы, управляющие внешними устройствами, и файловая система.

Физическая организация устройств ввода-вывода. Устройства ввода-вывода делятся на два типа: блок-ориентированные устройства и байт-ориентированные устройства. Блок-ориентированные устройства хранят информацию в блоках фиксированного размера, каждый из которых имеет свой собственный адрес. Байт-ориентированные устройства не адресуемы и не позволяют производить операцию поиска, они генерируют или потребляют последовательность байтов. Примерами являются терминалы, строчные

принтеры, сетевые адаптеры. Однако некоторые внешние устройства не относятся ни к одному классу.

Внешнее устройство обычно состоит из механического и электронного компонента. Электронный компонент называется контроллером устройства или адаптером. Механический компонент представляет собственно устройство. Некоторые контроллеры могут управлять несколькими устройствами. Если интерфейс между контроллером и устройством стандартизован, то независимые производители могут выпускать совместимые как контроллеры, так и устройства.

ОС выполняет ввод-вывод, записывая команды в регистры контроллера. Когда команда принята, процессор оставляет контроллер и занимается другой работой. При завершении команды контроллер организует прерывание, для того чтобы передать управление процессором ОС, которая должна проверить результаты операции. Процессор получает результаты и статус устройства, читая информацию из регистров контроллера.

Организация ПО ввода-вывода. Основная идея организации ПО ввода-вывода состоит в разбиении его на несколько уровней, причем нижние уровни обеспечивают экранирование особенностей аппаратуры от верхних, а те – удобный интерфейс для пользователей.

Ключевым принципом является независимость от устройств. Вид программы не должен зависеть от того, читает ли она данные с гибкого диска или с жесткого диска.

Другим важным вопросом для программного обеспечения ввода-вывода является обработка ошибок. Ошибки следует обрабатывать как можно ближе к аппаратуре. Если контроллер обнаруживает ошибку чтения, то он должен попытаться ее скорректировать. Если же это ему не удастся, то исправлением ошибок должен заняться драйвер устройства. Многие ошибки могут исчезать при повторных попытках выполнения операций ввода-вывода, например, ошибки, вызванные наличием пылинок на головках чтения или на диске. И только если нижний уровень не может справиться с ошибкой, он сообщает об ошибке верхнему уровню.

Большинство операций физического ввода-вывода выполняется асинхронно – процессор начинает передачу и переходит на другую работу, пока не наступает прерывание. Пользовательские программы намного легче писать, если операции ввода-вывода блокирующие – после команды READ программа автоматически

приостанавливается до тех пор, пока данные не попадут в буфер программы. ОС выполняет операции ввода-вывода асинхронно, но представляет их для пользовательских программ в синхронной форме.

Одни устройства являются разделяемыми, а другие – выделенными. Диски – это разделяемые устройства, так как одновременный доступ нескольких пользователей к диску не представляет собой проблему. Принтеры – это выделенные устройства, потому что нельзя смешивать строчки, печатаемые различными пользователями. Наличие выделенных устройств создает для операционной системы некоторые проблемы.

Исходя из вышесказанного, ПО ввода-вывода можно разделить на четыре слоя (рисунок 9):

- 1) обработка прерываний;
- 2) драйверы устройств;
- 3) независимый от устройств слой операционной системы;
- 4) пользовательский слой программного обеспечения.



Рисунок 9 – Многоуровневая организация подсистемы ввода-вывода

1.7 Драйверы устройств

Драйвер – компьютерная программа, с помощью которой другие программы получают доступ к аппаратному обеспечению некоторого устройства [11]. Обычно с ОС поставляются драйверы для ключевых компонентов аппаратного обеспечения, без которых система не может работать. Однако для некоторых устройств могут потребоваться специальные драйверы, как правило, представляемые производителем устройств. В общем случае, драйвер не обязан взаимодействовать с аппаратными устройствами, он может их только имитировать, представлять программные сервисы, не связанные с управлением устройствами, либо не делать ничего.

ОС управляет некоторым «виртуальным устройством», которое понимает стандартный набор команд. Драйвер переводит эти команды в команды, которые понимает непосредственно устройство. Эта идеология называется «абстрагированием от аппаратного обеспечения».

Драйвер состоит из нескольких функций, которые обрабатывают определенные события ОС. Обычно это семь основных событий: загрузка драйвера; выгрузка; открытие драйвера; чтение; запись; закрытие; управление вводом-выводом.

По мере развития систем, сочетающих в себе на одной плате не только центральные элементы компьютера, но и большинство устройств компьютера в целом, возник вопрос удобства поддержки таких систем, получивших название «аппаратная платформа», или просто «платформа». Сначала производители платформ поставляли набор отдельных драйверов для ОС, собранный на один носитель. Затем появились установочные пакеты, называвшиеся 4-in-1 и One touch и позволявшие упростить установку драйверов в систему. При этом можно выбрать либо полностью автоматическую установку всех драйверов, либо ручную нужных. Однако единого, устоявшегося термина долго не было. Современный термин, описывающий такие наборы драйверов устройств – Board Support Package, или «пакет поддержки платформы». Помимо собственно драйверов, он может, как и прочие установочные пакеты, содержать модули ОС и программы (рисунок 10).

Автоматизация установки драйверов (драйверпаки, использование внешних источников).

Такой механизм обеспечения необходимыми драйверами используется как в операционных системах семейства Microsoft, так и в Unix-подобных операционных системах.



Рисунок 10 – Размещение ОС

1.8 Стационарные вычислительные системы

Вычислительная система (ВС) – это взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для обработки информации [14].

Иногда под ВС понимают совокупность технических средств ЭВМ, в которую входит не менее двух процессоров, связанных общностью управления и использования общесистемных ресурсов (память, периферийные устройства, программное обеспечение и т. п.) (рисунок 11).

К ресурсам вычислительной системы относят такие средства вычислительной системы, которые могут быть выделены процессу обработки данных на определенный квант времени. Основными ресурсами ВС являются процессоры, области оперативной памяти, наборы данных, периферийные устройства, программы.

В зависимости от ряда признаков различают следующие вычислительные системы:

- однопрограммные и многопрограммные (в зависимости от количества программ, одновременно находящихся в оперативной памяти);
- индивидуального и коллективного пользования (в зависимости от числа пользователей, которые одновременно могут использовать ресурсы ВС);
- с пакетной обработкой и разделением времени (в зависимости от организации и обработки заданий);

- однопроцессорные, многопроцессорные и многомашинные (в зависимости от числа процессоров);
- сосредоточенные, распределенные (вычислительные сети) и ВС с теледоступом (в зависимости от территориального расположения и взаимодействия технических средств);
- работающие или не работающие в режиме реального времени (в зависимости от соотношения скоростей поступления задач в ВС и их решения);
- универсальные, специализированные и проблемно-ориентированные (в зависимости от назначения).



Рисунок 11 – Вычислительная система

1.9 Мобильные и портативные вычислительные системы

Мобильные вычислительные устройства завоевали массовую популярность во всем мире [3]. Не так давно можно было только мечтать о возможности пользоваться всеми функциями персонального компьютера в дороге. Отличным решением этой проблемы стали мобильные ПК. Сегодня, однако, возникает множество других мобильных потребностей.

В ответ на запросы рынка появилось множество мобильных устройств – от ноутбуков и нетбуков до MID (мобильных интернет-устройств), отвечающих разнообразным потребностям пользователей. Основные характеристики мобильных устройств приведены в таблице 1.

Таблица 1 – Основные характеристики мобильных устройств

Виды устройств	Краткое описание устройств	Модификации
Ноутбуки	Полнофункциональный мобильный ПК, обеспечивающий высокую производительность	
Нетбуки	Легкие и компактные компьютеры с поддержкой базовых вычислительных возможностей, предназначенные для использования в качестве второго компьютера или компьютера начального уровня	
Мобильные интернет – устройства (MID)	Сверхпортативные компьютеры карманного формата. Предназначены для полнофункционального подключения к Интернету, коммуникаций, мультимедиа	Планшет Смартфон
Гаджеты	Портативные специализированные устройства, не участвующие напрямую в реализации вычислительного процесса	Smartwatch GoogleGlass Mp3/mp4-плееры

Мобильные вычислительные устройства – это электронные устройства, типа ноутбуков, блокнотных компьютеров, планшетных компьютеров, портативных миникомпьютеров, личных цифровых помощников (персональный цифровой секретарь) и других устройств. Эти устройства имеют встроенные радио, сигнализирующие устройства, которые применяются для получения и передачи электронных данных в любом месте, которое покрывается радиотрансляционной сетью, используя при этом Wi-Fi или WiMax технологию.

Различные аксессуары предназначены для предложения различных услуг. Портативные вычислительные устройства, типа ноутбуков и записных книжек, могут использоваться для получения доступа к Интернету или локальной сети организации или глобальной сети сервера. Портативные устройства, типа мобильных телефонов, могут также применяться для получения доступа к Интернету с целью загрузки электронных данных.

Мобильные вычислительные устройства используются в различных отраслях – образовании и корпоративном обучении, розничной торговле, транспортировке и логистике, складировании

и распределении, в офисе и местах общего пользования, производстве, здравоохранении и в гостиничном бизнесе. Они дают возможность передавать большие объемы информации и другие электронные данные, получать доступ к всемирной паутине, поддерживать звук/видео во время конференц-связи, отправлять и получать сообщения электронной почты. Также помогают в своевременном управлении инфраструктурным инвентарем, информационном управлении, обслуживании клиента и доступе к необходимым ресурсам сети. Они позволяют организациям повысить производительность, сократить затраты и увеличить доходность.

Эти устройства применяются совместно с соответствующим программным обеспечением, которое адаптирует сетевые настройки устройства для приспособления к различным сетевым технологиям. Программное обеспечение использует пароли для предотвращения неправомерного доступа. Личная информация, хранящаяся в этих устройствах, может быть зашифрована для того, чтобы избежать кражи данных.

На рисунке 12 представлены мобильные вычислительные устройства.



Рисунок 12 – Мобильные вычислительные устройства

1.10 Бездисковые станции и рекламные планшеты

Бездисковая станция – это компьютер без жесткого диска. Такой компьютер является частью распределенной вычислительной системы (компьютерной сети). Файловая система этого компьютера располагается на других узлах компьютерной сети.

Бездисковая станция может быть самостоятельным вычислительным узлом сети, простым терминалом или тем и другим одновременно. Терминал отличается от полнофункционального компьютера тем, что он не выполняет пользовательских программ, а лишь отображает результаты работы этих программ, запущенных на других узлах сети.

В простейшем случае система состоит из сервера загрузки, одной или нескольких бездисковых станций и коммутатора (рисунок 13).



Рисунок 13 – Схема простой системы

Инициализация бездисковых станций происходит следующим образом:

1 При включении питания бездисковая станция выполняет свой код BIOS и код, записанный в ПЗУ своей сетевой карты (BOOTROM).

2 Бездисковая станция по сети обращается к серверу загрузки посредством широковещательного запроса.

3 Сервер загрузки дает бездисковой станции информацию о ее сетевых настройках и указывает место в сети, где расположены файлы операционной системы для неё.

4 Бездисковая станция выполняет загрузку операционной системы согласно соответствующим файлам конфигурации.

К представленной системе легко можно подключать дополнительные функциональные возможности. Это могут быть самые разные модули (рисунок 14).

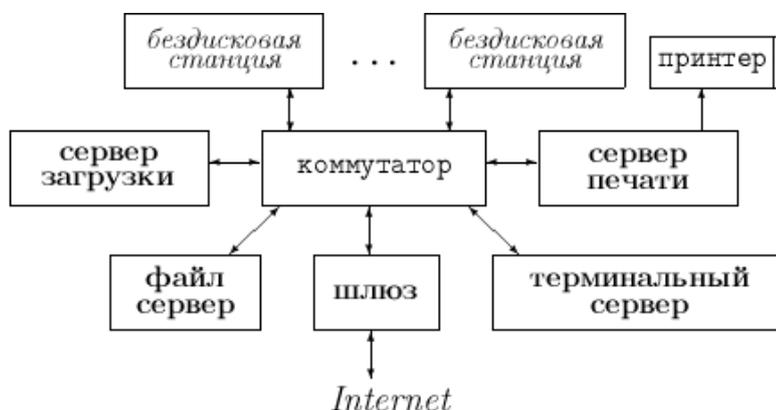


Рисунок 14 – Схема сложной системы

По мере роста конкуренции и развития корпоративной культуры, все большее число коммерческих компаний начинают использовать такие эффективные визуальные средства рекламы, как рекламные планшеты. Они предназначены для продвижения товаров услуг.

Для многих административных, государственных учреждений, компаний и банков рекламный планшет стал неотъемлемым атрибутом офиса – подобная «наглядная агитация» удобна, действует очень эффективно.

Различные рекламные планшеты стали настолько популярны, что ныне они востребованы не только внутри помещений, но и для наружного применения: на вокзалах, в аэропортах, на улицах города, возле больниц, учебных заведений.

1.11 Банкоматы, инфокиоски и фотокиоски

Банкомат – программно-технический комплекс, предназначенный для автоматизированной выдачи и приёма наличных денежных средств как с использованием платёжных карт, так и без них, а также для выполнения других операций, в том числе оплаты товаров и услуг, составления документов, подтверждающих соответствующие операции (рисунок 15) [9].



Рисунок 15 – Вид банкомата

После загрузки карты в карт-ридер банкомата пользователю предлагается ввести секретный код (ПИН-код) для авторизации картодержателя. Далее предлагается выбор доступных операций. После выбора операции банкомат шифрует полученную информацию и передает данные в процессинговый центр банка-эквайера (банка, обслуживающего банкомат).

Банк-эквайер отправляет в платежную систему запрос на проведение операции. Платежная система маршрутизирует запрос в банк-эмитент и, получив согласие либо отказ (код авторизации), передает банкомату команды на выполнение или отклонение запроса. При этом все действия по отправке запроса, обработке ответа на запрос, выдаче/приему денег из кассет фиксируются, что позволяет провести расследование в случае, если операция оспорена.

Так как ПИН-код известен только держателю карты, операции, подтвержденные ПИН-кодом, считаются выполненными непосредственно держателем карты.

Информационный киоск – автоматизированный программно-аппаратный комплекс, предназначенный для предоставления справочной информации.

Информационные киоски собирают на базе персонального компьютера, оснащенного сенсорным монитором и установленного в эргономичный вандалостойкий, как правило, металлический корпус. В отличие от обычного справочного киоска, электронный информационный киоск работает автономно.

Дополнительно на инфокиоск может устанавливаться купюроприемник, разъемы USB2.0, фискальный регистратор, аудиосистема, термопринтер, дополнительный рекламный монитор, сканер штрих кодов, RFID приемник, NFC и прочее оборудование.

Информационные киоски показаны на рисунке 16.



Рисунок 16 – Информационные киоски

Фотокиоск – специальный терминал, посредством которого осуществляется просмотр, распечатка и сохранение фотографий на различные носители.

Перед распечаткой фотографии многие фотокиоски позволяют их редактирование. Таким образом, фотографии можно вставить

в различные рамки, изменить их размер, наложить на них определенный фотофильтр.

Большинство фотокиосков, являясь платежными терминалами, позволяют оказывать и другие услуги, например, оплату услуг сотовой связи. Дополнительные сопутствующие услуги, внедренные в фотокиоск, увеличивают совокупный доход их владельца.

Фотокиоск (рисунок 17), как и другие платежные терминалы, может управляться удаленно. Для этого фотокиоск подключается к сети Интернет через EDGE/GPRS канал связи, предоставляемый сотовыми операторами. Таким образом, вы можете контролировать текущее состояние фотокиоска, как, например, расход чернил, длину чековой ленты, количество денежных средств в кассете, а также получать совокупную статистику по произведенным операциям.



Рисунок 17 – Фотокиоски

1.12 Использование гибридных телефонно-вычислительных устройств

Выделяют шесть основных поколений мобильной связи: поколение 1G, поколение 2G, поколение 2,5G, поколение 3G, поколение 3,5G, поколение 4G [25]. Рассмотрим поколения 2G, 3G, 4G.

Для начала, «G» означает «поколение», поэтому, когда вы слышите, что что-то относят к «сети 4G», это означает, что говорится о беспроводной сети, построенной на основе технологии четвертого поколения.

Самым известным стандартом 2G является GSM (Global System for Mobile Communications – Глобальная система мобильной связи). Широкое распространение 2G позволяет использовать международный роуминг между операторами сотовой связи, что дает возможность абоненту пользоваться своим телефоном практически

в любом уголке Земли. Причем именно возможность роуминга (в том числе и международного) является главной отличительной чертой стандарта GSM от стандартов первого поколения. Диапазон частот 1800 МГц.

Сети стандарта GSM позволяют предоставлять широкий перечень услуг: голосовые соединения; передачу данных (до 384 кбит/сек благодаря технологии EDGE); передачу коротких текстовых сообщений (SMS); передачу факсов; голосовую почту; конференц-связь и многое другое.

3G – технологии мобильной связи 3-го поколения – набор услуг, объединяющий как высокоскоростной мобильный доступ с услугами Интернет, так и технологию радиосвязи, которая создает канал передачи данных.

Устройства поколения 3G работают в диапазоне около 2 ГГц, передавая данные со скоростью до 3,6 Мбит/с. Они позволяют организовывать видеотелефонную связь, смотреть на мобильном телефоне фильмы и телепрограммы; с их применением улучшилось качество передаваемого звука, сеть стала более емкой, увеличилась скорость передачи данных и многое другое.

Наибольшее распространение получили 2 стандарта связи: **UMTS** (Universal Mobile Telecommunications System – универсальная система мобильной электросвязи) и **CDMA** (Code Division Multiple Access – множественный доступ с кодовым разделением каналов).

4G – поколение мобильной связи с повышенными требованиями. К 4G принято относить перспективные технологии, позволяющие осуществлять передачу данных со скоростью, превышающей 100 Мбит/с подвижным и 1 Гбит/с – стационарным абонентам. Примерами технологии являются **LTE** (Long Term Evolution – долговременное развитие) и **WiMAX** (Worldwide Interoperability for Microwave Access – Международное взаимодействие для микроволнового доступа).

Технология 4G основана на протоколах пакетной передачи данных. Для пересылки данных используется протокол IPv6, для передачи данных – частоты 40 и 60 GHz. Для четкого приёма и передачи планируют применять адаптивные антенны, которые смогут подстраиваться под конкретную базовую станцию.

Примером устройства 4G является первый специализированный и специально адаптированный для автомобилистов планшет **Asteroid** от французской компании Parrot, который поступил в продажу на рынок России. Главной его особенностью является то, что

он полностью без каких-либо проблем взаимодействует с электроникой автомобиля, что делает его очень удобным и функциональным для автолюбителей. Самые важные кнопки продублированы на выносном пульте дистанционного управления, который закреплен на самом руле, что позволяет всегда с легкостью управлять планшетом. Все остальные кнопки размещены по левому краю планшета, которыми также довольно удобно управлять (рисунок 18).



Рисунок 18 – Автомобильный планшет Parrot Asteroid с беспроводным сенсорным ПДУ

Каждый элемент интерфейса имеет большие размеры, для того чтобы было проще попасть пальцем по любому ярлыку на экране, что позволяет не подносить планшет близко и не будет лишней раз отвлекать водителя от дороги. Модель планшета оснащена 1 Гб встроенной памяти, а объем оперативной памяти равен 512 Мб. Диагональ экрана составляет 5 Мп с разрешением экрана 800 x 480 пикселей и поддержкой одновременного распознавания нескольких прикосновений.

Parrot Asteroid – это планшет с большим набором полезных функций для предоставления информационных услуг, беспроводной сети, навигационных и развлекательных услуг. На Parrot Asteroid установлена ОС Android v.2.3. Также на планшете наряду с беспроводными интерфейсами, такими как Bluetooth и Wi-Fi, установлено множество разъемов, через которые можно подключать магнитолу (АК ISO), 3G/4G модемы, внешний микрофон, USB-флэшнакопитель и карту памяти SD, а также другие планшеты, смартфоны и плееры. Также, как и для Google Play, пользователям предоставлен полный доступ ко всем приложениям Parrot ASTEROID Market.

Российская марка **TeXet** предлагает отечественным экстремалам и путешественникам смартфон TeXet X-Driver (TM 4104R) [16], полностью отвечающий их потребностям (рисунок 19). Теперь наслаждаться высококачественным изображением и связью можно даже в самом трудном походе. Аппарат оснащен всем необходимым для этого и способен выдержать непростые внешние условия.



Рисунок 19 – Смартфон X-Driver (TM 4104R)

Данный смартфон поддерживает следующие стандарты связи: GSM (2G), EDGE (2.9G), UMTS (3G), HSPA (3.5G).

Дизайн аппарата продуман до мелочей с учетом специфики модели. Аппарат выполнен в классическом форм-факторе типа моноблок. Степень защищенности корпуса TeXet TM-4104R отвечает установленному международному стандарту IP68. Цифра 6 подразумевает полную защиту от попадания пыли, а цифра 8 означает возможность функционирования устройства даже под водой. Он изготовлен из прочного пластика и дополнительно усилен специальными резиновыми пластинами. Все разъемы аппарата тщательно скрыты под плотноприлегающими заглушками, поставляемыми в комплекте. Подобному корпусу без труда удастся «пережить» погружение в толщу воды на глубину до 1 метра в течение длительного времени, падения на пол и прочие жесткие поверхности. Низкие температуры также не станут помехой нормальной работе устройства. Ударопрочный и герметичный корпус имеет в смартфоне сверхпрочное японское стекло Asashi Glass, что обеспечивает надежную защиту экрана от царапин и прочих повреждений.

Оригинал обзора: <http://megaobzor.com/obzor-mobilnogo-telefona-TeXet-TM-4104R.html>

2 ОПРЕДЕЛЕНИЕ И КЛАССИФИКАЦИЯ СИСТЕМНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Различные определения ОС

Операционная система (ОС) – комплекс системных программ по управлению ресурсами вычислительной системы [11]. Схема, иллюстрирующая место операционной системы в многоуровневой структуре компьютера, показана на рисунке 20.

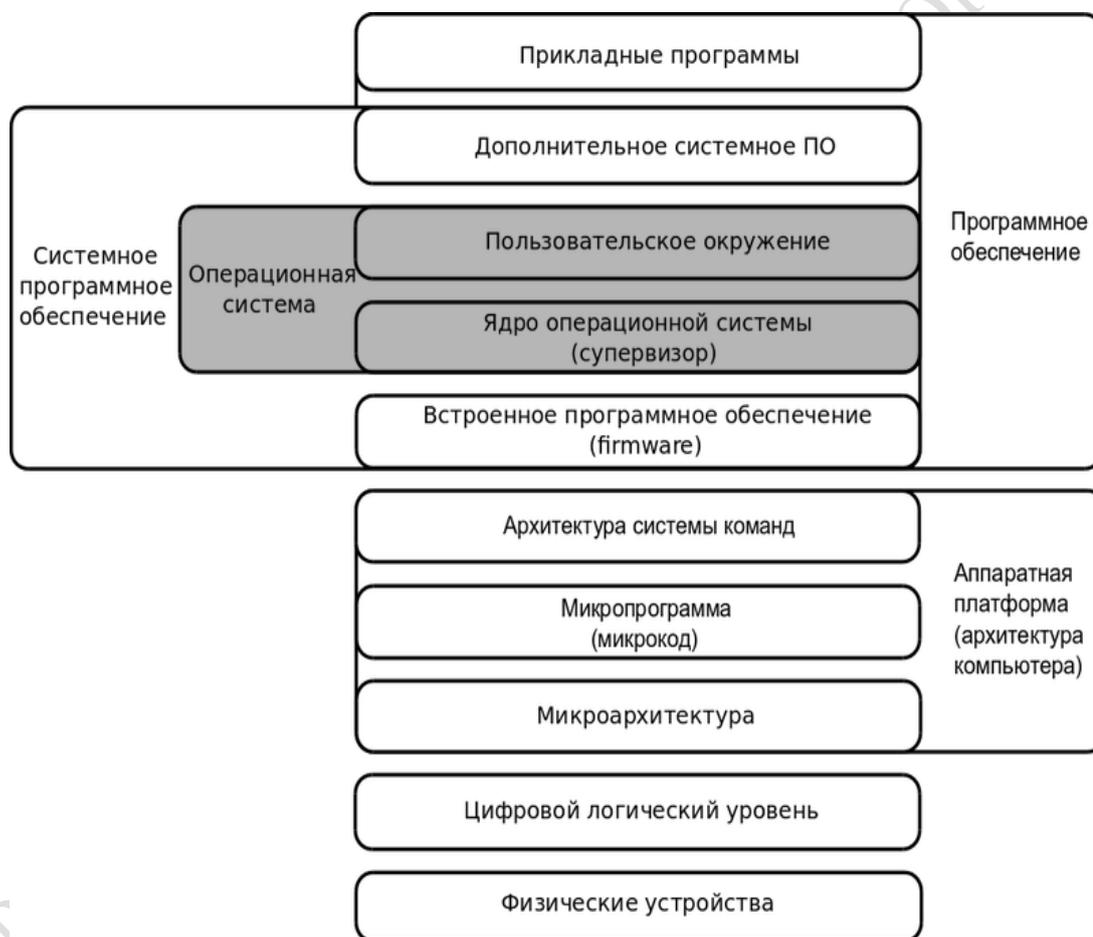


Рисунок 20 – Схема, иллюстрирующая место операционной системы в многоуровневой структуре компьютера

В логической структуре вычислительной системы операционная система занимает положение между устройствами с их микроархитектурой, машинным языком и, возможно, собственными (встроенными) микропрограммами – с одной стороны – и прикладными

программами, с другой. В большинстве вычислительных систем операционная система является основной, наиболее важной частью системного программного обеспечения.

Существуют две группы определений ОС:

- совокупность программ, управляющих оборудованием;
- совокупность программ, управляющих другими программами.

ОС – это программа, которая обеспечивает возможность рационального использования оборудования компьютера удобным для пользователя образом.

ОС – базовый комплекс компьютерных программ, обеспечивающий управление аппаратными средствами компьютера, работу с файлами, ввод и вывод данных, а также выполнение прикладных программ и утилит.

Кроме различных определений ОС, пользователи выделяют ряд различных «точек зрения» на ОС:

1 ОС как виртуальная машина.

Использование архитектуры персонального компьютера на уровне машинных команд является крайне неудобным для использования прикладных программ. Так, работа с диском предполагает знание внутреннего устройства его электронного компонента – контроллера, для ввода команд вращения диска, поиска и форматирования дорожек, чтения и записи секторов и т. д. Организация прерываний, работы таймера, управления памятью и так далее также может требовать при программировании знания и учета большого количества деталей.

В связи с этим необходимо обеспечить интерфейс между пользователем и компьютером, скрывая лишние подробности за счет использования относительно простых и высокоуровневых абстракций. Например, представлять информационное пространство диска как набор файлов, которые можно открывать для чтения или записи, использовать для получения или сброса информации, а затем закрывать; создавать иллюзию неограниченного размера операционной памяти, числа процессоров и прочее. Обеспечением такого высокоуровневого абстрагирования занимается ОС, что позволяет представлять ее пользователю в виде виртуальной машины, с которой проще иметь дело, чем непосредственно с оборудованием компьютера.

2 ОС как система управления ресурсами.

В случае, если несколько программ, работающих на одном компьютере, будут пытаться одновременно осуществлять вывод

на принтер, то можно получить «мешанину» строчек и страниц. ОС должна предотвращать такого рода хаос за счет буферизации подобной информации и организации очереди на печать. Не менее актуальная проблема – управление ресурсами для многопользовательских компьютеров.

Таким образом, ОС как менеджер ресурсов осуществляет упорядоченное и контролируемое распределение процессоров, памяти и других ресурсов между различными программами.

3 ОС как защитник пользователей и программ.

Если в вычислительной системе требуется обеспечение совместной работы нескольких пользователей, то возникает проблема организации их безопасной деятельности. Так, необходимо обеспечить:

- сохранность информации на диске, защиту от повреждения или удаления файлов;
- разрешение программам одних пользователей произвольно вмешиваться в работу программ других пользователей;
- пресечение попыток несанкционированного использования вычислительной системы.

Эти задачи, как правило, возложены на ОС как организатора безопасной работы пользователей и их программ.

4 ОС как постоянно функционирующее ядро.

Можно говорить об ОС, как о программе (программах), постоянно работающей на компьютере и взаимодействующей с множеством прикладных программ. Очевидно, что такое определение верно лишь отчасти, так как во многих современных ОС постоянно работает на компьютере лишь часть ОС, которую принято называть ее ядром.

2.2 Функции ОС

Операционная система – комплекс системных и служебных программных средств [22]. С одной стороны, она опирается на базовое программное обеспечение, входящее в BIOS, с другой стороны, сама является опорой для программного обеспечения более высокого уровня – прикладного и служебных приложений.

Операционные системы, под руководством которых работают прикладные программные средства, реализуют следующий ряд функций (рисунок 21).



Рисунок 21 – Функции ОС

1 Обеспечение интерфейса пользователя. По реализации интерфейса пользователя различают неграфические и графические операционные системы.

2 Обеспечение автоматического запуска. Все операционные системы обеспечивают свой автоматический запуск.

3 Организация файловой системы. Все современные дисковые ОС обеспечивают создание файловой системы, предназначенной для хранения данных на дисках и обеспечения доступа к ним.

4 Обслуживание файловой системы:

- создание и именованье файлов;
- создание каталогов (папок);
- копирование и перемещение файлов;
- удаление файлов и каталогов;
- навигация по файловой структуре;
- управление атрибутами файлов.

5 Управление установкой, исполнением и удалением приложений:

- установка приложений;
- удаление приложений.

6 Обеспечение взаимодействия с аппаратным обеспечением.

7 Обслуживание компьютера:

- средства проверки дисков;
- средства «сжатия» дисков;
- средства управления виртуальной памятью;
- средства кэширования дисков;
- средства резервного копирования данных.

Кроме основных (базовых) функций, ОС могут предоставлять различные дополнительные функции:

- возможность поддержки функционирования локальной компьютерной сети без специального программного обеспечения;

- обеспечение доступа к основным службам Интернета средствами, интегрированными в состав операционной системы;
- возможность создания системными средствами сервера Интернета, его обслуживание и управление посредством удаленного соединения;
- наличие средств защиты данных от несанкционированного доступа, просмотра и внесения изменений;
- возможность оформления рабочей среды операционной системы, в том числе и средствами, относящимися к категории мультимедиа;
- возможность обеспечения комфортной поочередной работы различных пользователей на одном персональном компьютере с сохранением персональных настроек рабочей среды каждого из них;
- возможность автоматического исполнения операций обслуживания компьютера по заданному расписанию или под управлением удаленного сервера;
- возможность работы с компьютером для лиц, имеющих физические недостатки, связанные с органами зрения, слуха и другими.

2.3 Категории пользователей ОС

Полезная особенность ОС Windows заключается в том, что в ней ведется строгий учет, кто именно сидит за компьютером и что ему разрешено делать [24].

ОС Windows может эффективно работать на семейных или служебных компьютерах, доступных пользователям с разной квалификацией и разными правами. Никак не ограничивать их действия – это то же самое, что заранее планировать сбои, уничтожение важных файлов, порчу критических настроек и другие неприятности.

Из этих соображений в системе Windows все пользователи классифицированы примерно по десяти разным категориям, из которых важнейшими являются три (рисунок 22):

- администраторы;
- обычные пользователи;
- посетители (гости);
- программисты.



Рисунок 22 – Основные категории пользователей ОС

Администратор (Administrator) имеет право устанавливать и удалять программы, изменять служебные настройки, определять права других пользователей. В общем, если операция может при ошибке вывести компьютер из строя, то выполнять ее разрешено только администратору. Система Windows устроена так, что хотя бы один пользователь с правами администратора зарегистрирован на компьютере всегда.

Обычный пользователь (User) имеет ограниченные права – это человек, использующий компьютер для работы. Он имеет право запускать программы, хотя и не все, а также создавать и редактировать документы. Пользователю выделяются личные папки на жестком диске, которые он может сделать недоступными для других пользователей.

Посетитель (Guest) – это временный пользователь, не имеющий личных файлов на данном компьютере. Он тоже может запускать программы, например, для просмотра веб-страниц Интернета или электронной почты. Если посетители допускаются на компьютер, пароль им не нужен.

Прикладные программисты занимаются разработкой программ для решения прикладных задач, реализации запросов к базе данных.

Системные программисты осуществляют поддержку информационной системы и обеспечивают ее работоспособность, занимаются разработкой и сопровождением базового программного обеспечения компьютеров (операционных систем, систем управления базами данных, трансляторов, сервисных программ общего назначения).

Все пользователи, которым разрешен доступ к компьютеру, должны быть на нем зарегистрированы. Сведения о пользователях хранятся в виде специальных записей в Реестре Windows.

Эти записи исполняют роль регистрационных записей и называются учетными записями.

Software Asset Management (SAM) – это методология, направленная на оптимизацию процессов управления активами ПО в организации и их защиту: учет ПО, его использования, лицензий (прав на использование), документов, подтверждающих наличие прав на использование (сертификаты, лицензионные свидетельства, лицензионные соглашения, договора, бухгалтерские документы), разработка и использование регламентов и политик закупки ПО, ввода его в эксплуатацию, эксплуатация, вывод из эксплуатации и др.

2.4 Логические уровни общения человек – ЭВМ

ОС является посредником (связующим звеном, интерфейсом) между ЭВМ, с одной стороны, и человеком (пользователем, программистом, оператором, инженером и т. д.) – с другой.

Другими словами, ОС – логическое расширение функций аппаратуры в сторону человека. Она позволяет от «физического» уровня аппаратуры перейти к более высокому «логическому» уровню, который становится уровнем вычислительной системы и который более удобен для человека.

Можно выделить три уровня общения человека с ЭВМ:

- логический;
- реляционный;
- уровень представления знаний.

На логическом уровне любое высказывание оценивается как истинное или ложное.

Обычно значения истинности представляются элементами одного из следующих множеств:

{истина, ложь}, {true, false}, {1, 0}.

В принципе, для представления истинности значений высказываний может использоваться любое двухэлементное множество. Например, истинность высказывания «Предприятие заплатило налоги» может принять значение «да» (1) или «нет» (0).

Структура логического уровня общения человек – ЭВМ представлена на рисунке 23.

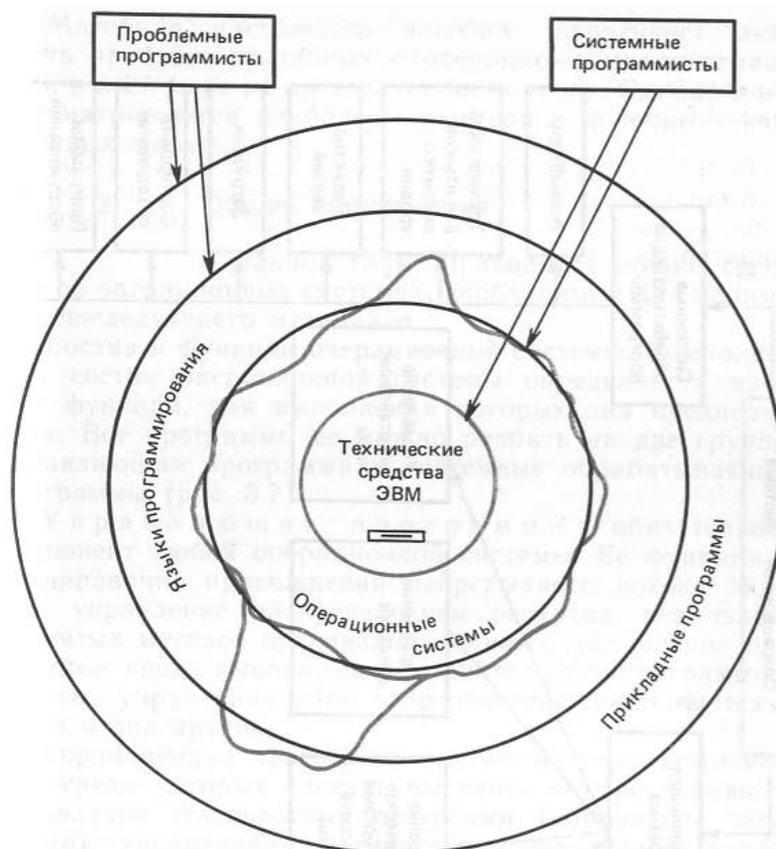


Рисунок 23 – Логический уровень общения человек – ЭВМ

2.5 Мультипрограммный режим работы ОС

Мультипрограммирование, или многозадачность, – это способ организации вычислительного процесса, при котором на одном процессоре попеременно выполняются сразу несколько программ [24] (рисунок 24). Эти программы совместно используют не только процессор, но и другие ресурсы компьютера: оперативную и внешнюю память, устройства ввода-вывода, данные. Мультипрограммирование призвано повысить эффективность использования вычислительной системы, однако эффективность может пониматься по-разному. Наиболее характерными критериями эффективности вычислительных систем являются:

- пропускная способность – количество задач, выполняемых вычислительной системой в единице времени;
- удобство работы пользователей;
- реактивность системы – способность её выдерживать заранее заданные (возможно, очень короткие) интервалы времени между запуском программы и получением результата.

Мультипрограммный режим		
Пакетная обработка	система разделения времени (СРВ)	система реального времени
	пакетный режим	системы управления процессами:
	режим запрос – ответ	автоматические
	диалоговый	коммерческие
	универсальный	диалоговые запросные системы

Рисунок 24 – Мультипрограммный режим работы

В зависимости от выбранного критерия эффективности ОС делятся на системы пакетной обработки, системы разделения времени и системы реального времени. Каждый тип ОС имеет специфические внутренние механизмы и особые области применения. Некоторые операционные системы могут поддерживать одновременно несколько режимов, например, часть задач может выполняться в режиме пакетной обработки, а часть – в режиме реального времени или в режиме разделения времени.

Пакетная обработка применяется для достижения максимальной эффективности использования ресурсов вычислительной машины при выполнении вычислительных задач путём сбалансированной загрузки её компонентов. Задачи, планируемые к выполнению, называются пакетом. Переключение между задачами в пакетном режиме инициируется выполняющейся в данный момент задачей, поэтому промежутки времени выполнения той или иной задачи не определены.

Системы разделения времени используются для «одновременного» выполнения нескольких программ в интерактивном режиме. В отличие от пакетного режима, все программы получают определённые временные промежутки для выполнения, затем система инициирует переключение. Выделяемые временные интервалы могут быть равными для всех задач, а могут определяться их приоритетами.

Системы реального времени предназначены для управления от компьютера различными техническими объектами или технологическими процессами. Во всех этих случаях существует предельно допустимое время, в течение которого должна быть выполнена та или иная управляющая объектом программа.

2.6 Описание различных режимов работы ОС

Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности (рисунок 25):

- системы пакетной обработки (например, ОС ЕС);
- системы разделения времени (UNIX, VMS VMS);
- системы реального времени (QNX, RT/11).

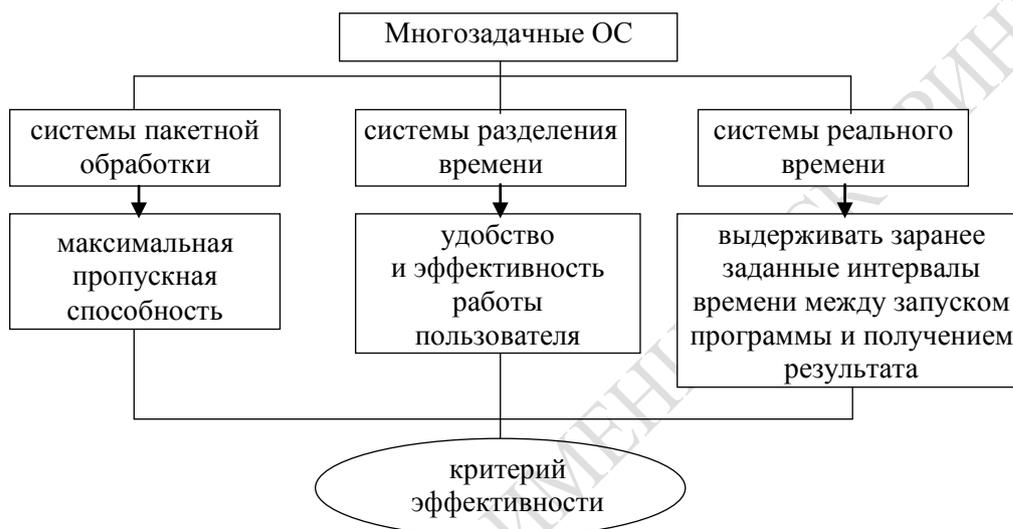


Рисунок 25 – Критерии эффективности многозадачных ОС

Системы пакетной обработки предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов. Главной целью и критерием эффективности систем пакетной обработки является максимальная пропускная способность, то есть решение максимального числа задач в единицу времени.

В системах пакетной обработки переключение процессора с выполнения одной задачи на выполнение другой происходит только в случае, если активная задача сама отказывается от процессора, например, из-за необходимости выполнить операцию ввода-вывода. Поэтому одна задача может надолго занять процессор, что делает невозможным выполнение интерактивных задач. Таким образом, взаимодействие пользователя с вычислительной машиной, на которой установлена система пакетной обработки, сводится к тому, что он приносит задание, отдает его диспетчеру-оператору, а в конце дня после выполнения всего пакета заданий получает результат. Очевидно, что такой порядок снижает эффективность работы пользователя.

Системы разделения времени призваны исправить основной недостаток систем пакетной обработки – изоляцию пользователя-программиста от процесса выполнения его задач. Каждому пользователю системы разделения времени предоставляется терминал, с которого он может вести диалог со своей программой.

Системы разделения времени обладают меньшей пропускной способностью, чем системы пакетной обработки, так как на выполнение принимается каждая запущенная пользователем задача, а не та, которая «выгодна» системе, и, кроме того, имеются накладные расходы вычислительной мощности на более частое переключение процессора с задачи на задачу. Критерием эффективности систем разделения времени является не максимальная пропускная способность, а удобство и эффективность работы пользователя.

Системы реального времени применяются для управления различными техническими объектами, такими, например, как станок, спутник, научная экспериментальная установка, или технологическими процессами, такими как гальваническая линия, доменный процесс и т. п. Во всех этих случаях существует предельно допустимое время, в течение которого должна быть выполнена та или иная программа, управляющая объектом. Критерием эффективности для систем реального времени является их способность выдерживать заранее заданные интервалы времени между запуском программы и получением результата (управляющего воздействия). Это время называется временем реакции системы, а соответствующее свойство системы – реактивностью. Для этих систем мультипрограммная смесь представляет собой фиксированный набор заранее разработанных программ, а выбор программы на выполнение осуществляется исходя из текущего состояния объекта или в соответствии с расписанием плановых работ.

2.7 Однозадачные и многозадачные операционные системы

По числу одновременно выполняемых задач операционные системы делятся на два класса:

- однозадачные (MS-DOS, DR-DOS, Palm OS и т. д.);
- многозадачные (OS/2, Unix, Windows, MS Pocket PC, Symbian, Mac OS).

Однозадачные ОС в основном выполняют функцию предоставления пользователю виртуальной машины, делая более простым

и удобным процесс взаимодействия пользователя с компьютером. В однозадачных системах используются средства управления периферийными устройствами, средства управления файлами, средства общения с пользователями. Многозадачные ОС используют все средства, которые характерны для однозадачных, и, кроме того, управляют разделением совместно используемых ресурсов: процессор, ОЗУ, файлы и внешние устройства.

В зависимости от областей использования многозадачные ОС подразделяются на три типа:

- системы пакетной обработки (ОС ЕС);
- системы с разделением времени (Windows 98/NT/2000/XP, Linux);
- системы реального времени (QNX, RT/11).

Системы пакетной обработки предназначены для решения задач, которые не требуют быстрого получения результатов. Главной целью ОС пакетной обработки является максимальная пропускная способность или решение максимального числа задач в единицу времени. Эти системы обеспечивают высокую производительность при обработке больших объемов информации, но снижают эффективность работы пользователя в интерактивном режиме.

В системах с разделением времени для выполнения каждой задачи выделяется небольшой промежуток времени, и ни одна задача не занимает процессор надолго. Если этот промежуток времени выбран минимальным, то создается видимость одновременного выполнения нескольких задач. Эти системы обладают меньшей пропускной способностью, но обеспечивают высокую эффективность работы пользователя в интерактивном режиме.

Системы реального времени применяются для управления технологическим процессом или техническим объектом.

В таблице 1 представлены виды однозадачных и многозадачных ОС для смартфонов, компьютеров и PDA.

Таблица 1 – Однозадачные и многозадачные ОС

Смартфоны		PDA		Компьютеры	
одно-задачные	много-задачные	одно-задачные	много-задачные	одно-задачные	много-задачные
Windows Phone, Bada	BlackBerry OS, Windows Mobile, Apple iOS, Symbian OS, Android	Palm OS, Windows Phone 7	Windows Mobile	MS-DOS, MSX	ОС ЕС, OS/2, Unix, Windows NT

2.8 Однопользовательские и многопользовательские ОС

По числу одновременно работающих пользователей ОС делятся на [22] (рисунок 26):

– однопользовательские (MS-DOS, Windows 3.x, ранние версии OS/2);

– многопользовательские (UNIX, Windows NT).

Главным отличием многопользовательских систем от однопользовательских является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей. Следует заметить, что не всякая многозадачная система является многопользовательской, и не всякая однопользовательская ОС является однозадачной.

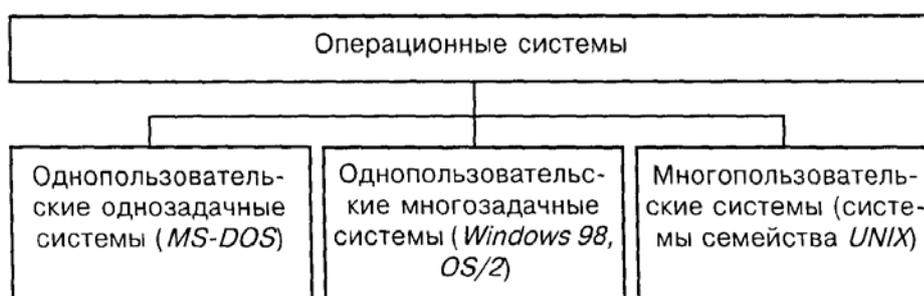


Рисунок 26 – Классификация ОС по числу одновременно работающих пользователей

Однопользовательские ОС бывают двух видов:

1 Однопользовательские однозадачные системы предназначены для управления компьютером таким образом, чтобы в любой заданный момент времени один пользователь мог эффективно выполнять одну задачу либо действие. Хорошим примером однопользовательской однозадачной ОС является Palm OS для карманных компьютеров Palm.

2 Однопользовательские многозадачные – такие ОС большинство пользователей в настоящее время применяют в своих настольных компьютерах и ноутбуках. Windows от Microsoft и MacOS от Apple – примеры операционных систем, позволяющих одному пользователю одновременно выполнять несколько программ.

Многопользовательская система позволяет многим разным людям одновременно пользоваться ресурсами одного компьютера. Операционная система должна сбалансировать требования различных

пользователей, а также обеспечить использование каждой задействованной ими программой достаточных и разделенных ресурсов, чтобы проблема, возникшая у одного пользователя, не распространилась на все сообщество пользователей.

2.9 Основные характеристики и особенности работы ОС MS-DOS и управление памятью в MS-DOS

Операционная система MS-DOS – это однопользовательская, однозадачная, не сетевая 16-разрядная операционная система, ориентированная на использование на ПЭВМ с микропроцессором Intel 8088(80286) [18].

Основными характеристиками данной ОС являются (рисунок 27):

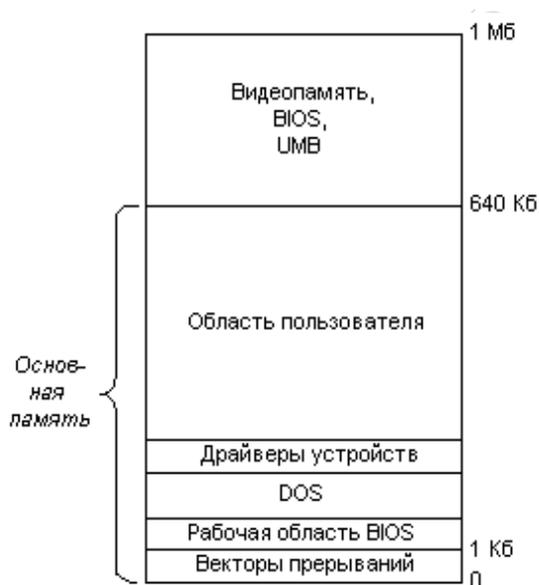


Рисунок 27 – Распределение памяти в ОС MS-DOS

– максимальный объем адресуемой физической памяти – 640 Кбайт;

– максимальный объем памяти, доступный из прикладных программ – 640 Кбайт. Последние версии MS-DOS (начиная с 5.0) могут использовать адресное пространство между 640 Кбайт и 1 Мбайт для размещения своих составных частей и некоторых драйверов, освобождая тем самым память в адресном пространстве 0–640 Кбайт для прикладных программ;

– представление всех ресурсов персонального компьютера для одной, активной в настоящий момент, программы;

- развитая файловая система и процессор командного языка;
- слабая поддержка интерактивных средств взаимодействия с пользователем;
- занимаемый объем на диске, в зависимости от версии, от 1 Мбайта до 6 Мбайт (минимум, при котором можно работать – 100 Кбайт).

Под управлением памятью понимают процессы обеспечения доступа к ней, распределения ее между программами и контроля работы ОЗУ компьютера. Функции управления работой памяти в MS-DOS полностью возложены на саму операционную систему. Другими средствами управления памятью в DOS являются различные загружаемые драйверы и специальные команды.

Нижнюю часть памяти занимают модули ОС. Некоторые системные программы могут быть ради экономии загружены в верхний блок памяти (выше 640 Кб). Все, что остается в середине, может быть предоставлено процессу пользователя [26].

Для экономии памяти некоторые нерезидентные модули DOS могут занимать верхнюю часть области пользователя, но только до тех пор, пока не будут затерты пользовательской программой, которой потребуется вся имеющаяся память.

Часть системной памяти и вся область пользователя разбита на прилегающие друг к другу блоки, размер которых кратен параграфу. Перед началом каждого блока памяти размещается блок управления памятью, который занимает один параграф и содержит следующие данные:

- признак, определяющий, последний ли это блок памяти или за ним будут еще блоки;
- адрес PSP программы, владеющей этим блоком;
- размер блока в параграфах;
- имя программы-владельца (до 8 символов).

Когда система должна выделить блок памяти для собственных нужд или по запросу программы пользователя, она просматривает список блоков от начала, перемещаясь от одного МСВ к следующему. Найдя свободный блок достаточного размера, система отмечает его как занятый соответствующим владельцем. Если выделяется не весь свободный блок, то после выделенного блока система записывает еще один МСВ, описывающий свободный остаток блока.

При освобождении блока система записывает 0 в поле владельца МСВ. Если с одной или с двух сторон от освобождаемого блока лежат свободные блоки, то два или три свободных блока сливаются в один.

2.10 Описание особенностей работы ОС Windows

Windows – это объектно-ориентированная среда. Все элементы интерфейса (окна, кнопки, пиктограммы и др.) – это объекты, каждый со своими свойствами и способами поведения [5].

Для работы с объектами предусмотрена система контекстных меню, которые отражают их свойства и поведение, позволяют изменять свойства (настраивать) и выполнять допустимые над данным объектом действия. По существу, все действия, выполняемые в ОС Windows, – это обмен сообщениями между объектами.

К особенностям ОС Windows относятся:

1) вытесняющая многозадачность (процессор переключается между программами);

2) единый программный интерфейс – это возможность создавать данные в одних программах и переносить их в другие;

3) единый интерфейс пользователя. Это означает, что интерфейс приложений (программ, написанных для работы под управлением ОС) стандартизован, а значит, изучив одну программу, легче освоить другие;

4) единый аппаратно-программный интерфейс. ОС Windows сама обеспечивает совместимость разнообразного оборудования и программ. Изготовители оборудования добиваются только работы с Windows, а дальше ОС берет на себя заботы по обеспечению работы устройств. Точно так же изготовители программ могут не беспокоиться о работе с неизвестным им оборудованием. Их задача сводится к тому, чтобы обеспечить взаимодействие с Windows. ОС Windows ввела новый стандарт самоустанавливающихся устройств (plug and play). Подключение таких устройств происходит автоматически. ОС сама «узнает», что установлено в компьютере, и настраивается на работу с новым оборудованием.

К преимуществам Windows можно отнести следующие:

1) принцип точного отображения WYSIWYG (What You See Is What You Get). При работе с документом пользователь видит на экране изображение, практически полностью соответствующее тому, что будет получено после распечатки на принтере. Это происходит благодаря использованию масштабируемых шрифтов TrueType;

2) удобные принципы работы с мышью (в частности, принцип Drag and Drop);

3) встроенные сетевые возможности и поддержка работы в Internet;

- 4) широкая поддержка средств мультимедиа.
- 5) особые возможности для пользователей с отклонениями в состоянии здоровья.

Архитектура ОС Microsoft Windows 2000 представлена на рисунке 28.

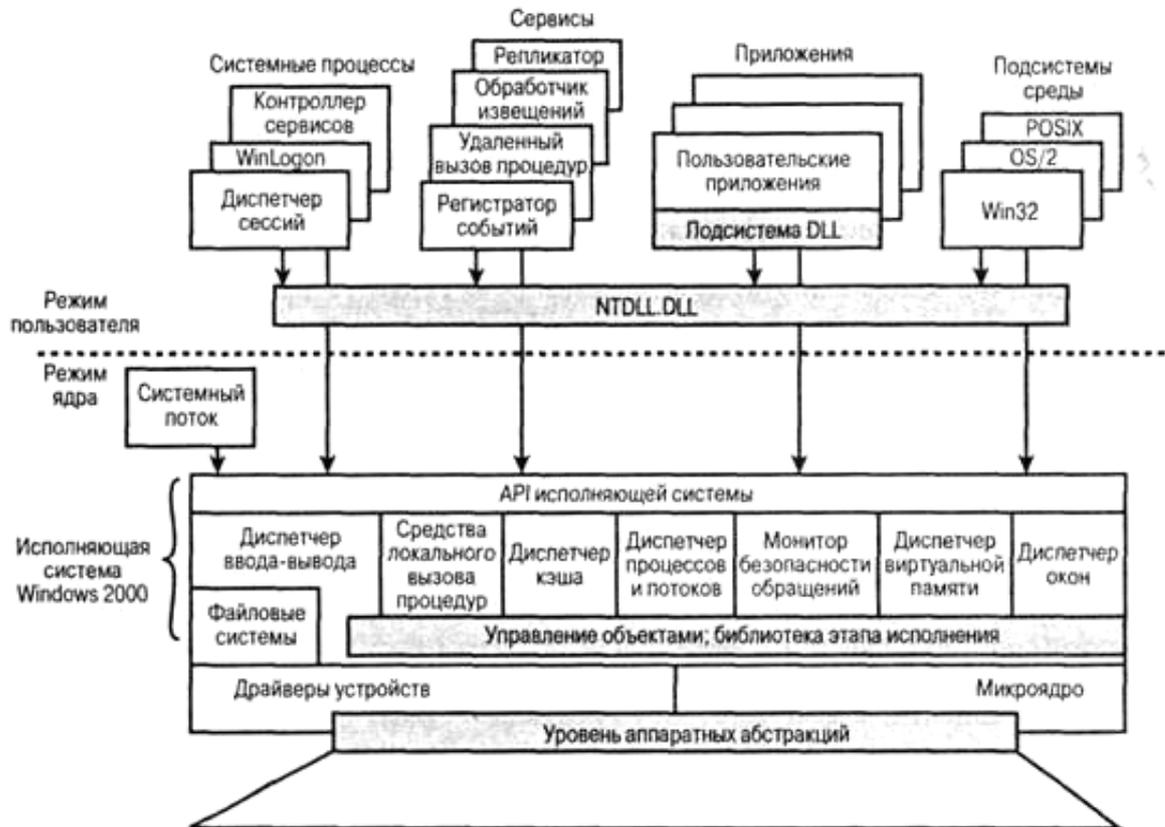


Рисунок 28 – Архитектура ОС Windows 2000

2.11 ОС семейства Unix и их особенности работы

UNIX – семейство переносимых, многозадачных и многопользовательских операционных систем [4].

Общими чертами для любой ОС UNIX считаются:

- многопользовательский режим со способом защиты данных от несанкционированного доступа;
- реализация мультипрограммной обработки в режиме разделения времени, которая основана на применении алгоритмов вытесняющей многозадачности; повышение уровня мультипрограммирования;
- унификация операций ввода-вывода на базе расширенного использования понятия «файл»;

- иерархическая файловая система, которая образует единое дерево каталогов независимо от числа физических устройств, используемых для размещения файлов;
- переносимость системы, которая осуществляется за счет написания ее основной части на языке С;
- разнообразные средства взаимодействия процессов, например, через сеть;
- кэширование диска с целью уменьшения среднего времени доступа к файлам.

Архитектура Unix (на рисунке 29) – двухуровневая модель системы, состоящая из пользовательской и системной части (ядра). Ядро непосредственно взаимодействует с аппаратной частью компьютера, изолируя прикладные программы от особенностей ее архитектуры. Ядро имеет набор услуг, предоставляемых прикладным программам посредством системных вызовов. Таким образом, в системе можно выделить два уровня привилегий: уровень системы (привилегии специального пользователя root) и уровень пользователя (привилегии всех остальных пользователей).



Операционную систему UNIX можно представить в виде трёх уровней:

- монолитное ядро;
- системные утилиты и демоны;
- пользовательские программы.

Существует два уровня доступа:

- системный (ядро и службы);
- пользовательский.

Рисунок 29 – Архитектура Unix

Важную часть системных программ составляют демоны. Демон – это процесс, выполняющий определенную функцию в системе, который запускается при старте системы и не связан ни с одним пользовательским терминалом.

ОС Unix обладает классическим монолитным ядром (рисунок 30), в котором можно выделить следующие основные части: файловая подсистема, управление процессами и драйверы устройств.

ОС UNIX обеспечивает ряд стандартных системных программ для решения задач администрирования, переконфигурирования и поддержки файловой системы, в частности:

- для настройки параметров конфигурации системы;
- для переконфигурирования ядра (если она необходима) и добавления новых драйверов устройств;
- для создания и удаления учетных записей пользователей;
- для создания и подключения физических файловых систем;
- для установки параметров контроля доступа к файлам.



Рисунок 30 – Ядро Unix

В целях решения этих задач системное ПО (работающее в пользовательском режиме) часто использует системные вызовы.

Одними из самых известных примеров UNIX-подобных ОС являются OS X, Linux, Solaris, BSD и NeXTSTEP (рисунок 31).



Рисунок 31 – Примеры Unix-подобных ОС

3 ИНТЕРФЕЙСЫ ОПЕРАЦИОННЫХ СИСТЕМ

3.1 Понятие интерфейса и его основные виды

Интерфейс – это совокупность логических и физических принципов взаимодействия компонентов технических средств вычислительной системы (ВС), то есть совокупность правил алгоритмов и временных соглашений по обмену данными между компонентами ВС (логический интерфейс), а также совокупность физических, механических и функциональных характеристик средств подключения, реализующих такое взаимодействие (физический интерфейс) [24].

Интерфейсом нередко называют также технические и программные средства, реализующие сопряжение между устройствами и узлами ВС. Интерфейс распространяется на все логические и физические средства взаимодействия вычислительной системы с внешней средой, например с операционной системой, с оператором и т. п.

Интерфейсы различают по таким характеристикам, как структура связей, способ подключения и передачи данных, принципы управления и синхронизации.

1 Внутримашинный интерфейс – система связи и средств сопряжения узлов и блоков ЭВМ между собой. Внутримашинный интерфейс представляет собой совокупность электрических линий связи (проводов), схем сопряжения с компонентами компьютера, протоколов (алгоритмов) передачи и преобразования сигналов. Различают два варианта организации интерфейса:

- многосвязный интерфейс, при котором каждый блок ПК связан с другими блоками своими локальными проводами;
- односвязный интерфейс, в результате которого все блоки ПК связаны друг с другом через общую или системную шину.

2 Внешний интерфейс – система связи системного блока с периферийными устройствами ЭВМ или с другими ЭВМ:

- интерфейс периферийных устройств, подключаемых с помощью шин ввода-вывода (ISA, EISA, VLB, PCI, AGP, USB IEEE 1384 SCSI и др.);
- сетевой интерфейс, типа одноранговой сети или сети клиент – сервер с топологиями типа звезда, кольцевая или шинная.

3 Интерфейс «человек – машина» или интерфейс «человек – компьютер» или пользовательский интерфейс – это способ, которым выполняется какая-либо задача с помощью каких-либо

средств, а именно совершаемые действия и то, что получают в ответ. Интерфейс является ориентированным на человека, если он отвечает нуждам человека и учитывает его слабости.

Различают три вида интерфейсов пользователя: командный, WIMP- и SILK-интерфейсы.

Взаимодействие перечисленных интерфейсов с операционными системами и технологиями показано на рисунке 32.

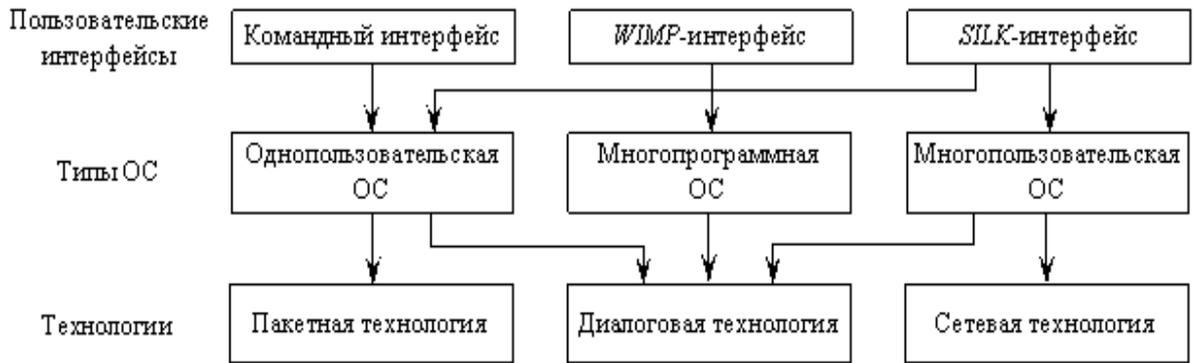


Рисунок 32 – Пример взаимодействия интерфейсов с операционными системами и технологиями

Схема взаимодействия пользователя с компьютером представлена на рисунке 33.

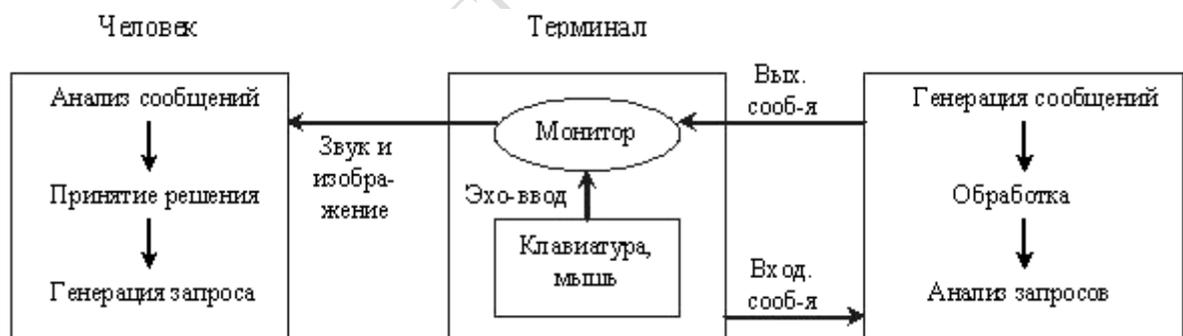


Рисунок 33 – Взаимодействие пользователя с компьютером

3.2 Пакетный режим работы ОС

Исторически этот вид технологии появился первым [11]. Она существовала уже на релейных машинах Зюса и Цюзе (Германия, 1937 год). Идея ее проста: на вход компьютера подается последовательность символов, в которых по определенным правилам указывается последовательность запущенных на выполнение программ.

После выполнения очередной программы запускается следующая и т. д. Машина по определенным правилам находит для себя команды и данные. В качестве этой последовательности может выступать, например, перфолента, стопка перфокарт, последовательность нажатия клавиш электрической пишущей машинки (типа CONSUL). Машина также выдает свои сообщения на перфоратор, алфавитно-цифровое печатающее устройство (АЦПУ), ленту пишущей машинки.

Такая машина представляет собой «черный ящик», в который постоянно подается информация, в том числе фиксирующая данные о своём текущем состоянии (рисунок 34). Человек здесь имеет малое влияние на работу машины – он может лишь приостановить работу машины, сменить программу и вновь запустить ЭВМ. Впоследствии, когда машины стали помощнее и могли обслуживать сразу нескольких пользователей, вечное ожидание пользователей типа: «Я послал данные машине. Жду, что она ответит. И ответит ли вообще?» – стало, мягко говоря, надоедать. К тому же вычислительные центры стали вторым крупным «производителем» макулатуры. Поэтому с появлением алфавитно-цифровых дисплеев началась эра по-настоящему пользовательской технологии – командной строки.

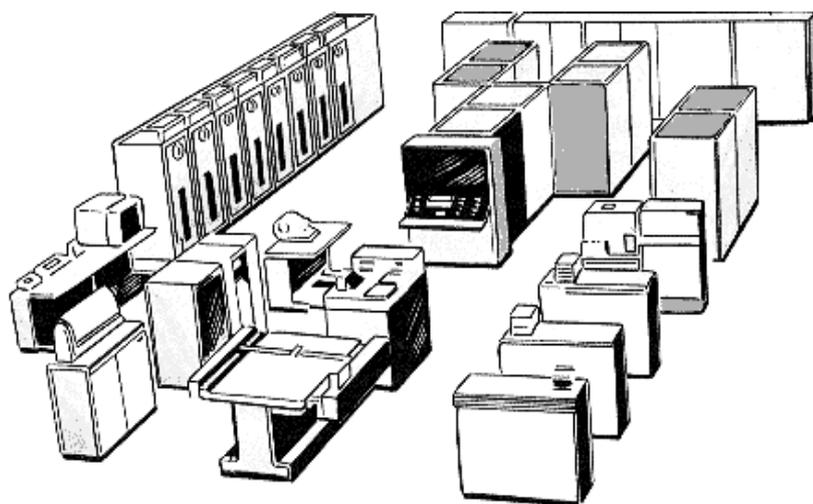


Рисунок 34 – Вид большой ЭВМ серии ЕС ЭВМ

3.3 Описание простого графического интерфейса

На первом этапе графический интерфейс очень походил на технологию командной строки [15]. Отличия от технологии командной строки заключались в следующем:

1 При отображении символов допускалось выделение части символов цветом, инверсным изображением, подчеркиванием и мерцанием. Благодаря этому повысилась выразительность изображения.

2 В зависимости от конкретной реализации графического интерфейса курсор может представляться не только мерцающим прямоугольником, но и некоторой областью, охватывающей несколько символов и даже часть экрана. Эта выделенная область отличается от других, невыделенных частей (обычно цветом).

3 Нажатие клавиши **Enter** не всегда приводит к выполнению команды и переходу к следующей строке. Реакция на нажатие любой клавиши во многом зависит от того, в какой части экрана находился курсор.

4 Кроме клавиши **Enter**, на клавиатуре все чаще стали использоваться «серые» клавиши управления курсором.

5 Уже в этой редакции графического интерфейса стали использоваться манипуляторы (рисунок 35). Они позволяли быстро выделять нужную часть экрана и перемещать курсор.

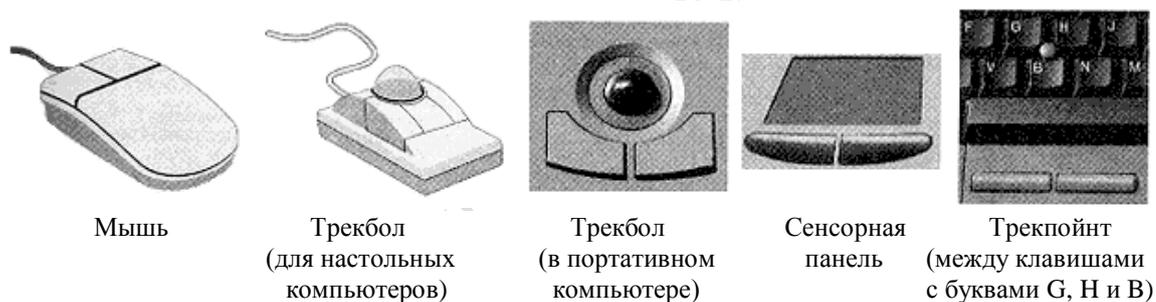


Рисунок 35 – Манипуляторы

Типичным примером использования этого вида интерфейса является файловая оболочка Norton Commander (о файловых оболочках смотри ниже) и текстовый редактор Multi-Edit. А текстовые редакторы Лексикон, ChiWriter и текстовый процессор Microsoft Word for Dos являются примером, как этот интерфейс превзошел сам себя.

3.4 Описание WIMP интерфейса

Вторым этапом в развитии графического интерфейса стал «чистый» интерфейс WIMP [15].

Многооконный WIMP-интерфейс расшифровывается следующим образом:

W – информация представляется пользователю на экране дисплея в виде нескольких окон (Windows);

I – объекты, с которыми информационная система имеет дело, представляются в виде пиктограмм (Icons);

M – выборка производится с помощью манипулятора типа «мышь» (Mouse);

P – означает меню, которые автоматически всплывают (pop-up) на экране или которые пользователь может «вытянуть» (pull down) из строки меню.

Характерной особенностью этого вида интерфейса является то, что диалог с пользователем ведется не с помощью команд, а с помощью графических образов – меню, окон, других элементов (рисунок 36). Хотя и в этом интерфейсе подаются команды машине, но это делается «опосредственно», через графические образы.



Рисунок 36 – Пример WIMP-интерфейса

Этот подвид интерфейса характеризуется следующими особенностями:

1) вся работа с программами, файлами и документами происходит в окнах – определенных очерченных рамкой частях экрана;

2) все программы, файлы, документы, устройства и другие объекты представляются в виде значков – иконок. При открытии иконки превращаются в окна;

3) все действия с объектами осуществляются с помощью меню. Хотя меню появилось на первом этапе становления графического интерфейса, оно не имело в нем главенствующего значения, а служило лишь дополнением к командной строке. В чистом WIMP-интерфейсе меню становится основным элементом управления;

4) широкое использование манипуляторов для указания на объекты. Манипулятор перестает быть просто игрушкой – дополнением к клавиатуре, а становится основным элементом управления. С помощью манипулятора **УКАЗЫВАЮТ** на любую область экрана, окна или иконки, **ВЫДЕЛЯЮТ** ее, а уже потом через меню или с использованием других технологий осуществляют управление ими.

Следует отметить, что WIMP требует для своей реализации цветной растровый дисплей с высоким разрешением и манипулятор. Также программы, ориентированные на этот вид интерфейса, предъявляют повышенные требования к производительности компьютера, объему его памяти, пропускной способности шины и т. п. Однако этот вид интерфейса наиболее прост в усвоении и интуитивно понятен. Поэтому сейчас WIMP-интерфейс стал стандартом де-факто.

Ярким примером программ с графическим интерфейсом является операционная система Microsoft Windows [5].

3.5 Описание SILK-интерфейса и необходимое оборудование для его реализации

SILK-интерфейс расшифровывается как Speech (речь), Image (образ), Language (язык), Knowledge (знание) [15].

Этот интерфейс наиболее приближен к обычной человеческой форме общения. В рамках этого интерфейса идет обычный разговор человека и компьютера. При этом компьютер находит для себя команды, анализируя человеческую речь и находя в ней ключевые фразы. Результаты выполнения команд он также преобразует в понятную человеку форму.

SILK-интерфейс для общения человека с машиной использует:

- речевую технологию;
- биометрическую технологию (мимический интерфейс);
- семантический (общественный) интерфейс.

Речевая технология (рисунок 37) возникла в середине 90-х годов после появления недорогих звуковых карт и широкого распространения технологий распознавания речи. При этой технологии команды подаются голосом путем произнесения специальных зарезервированных слов – команд. Основными такими командами являются:

- «Проснись» – включение голосового интерфейса;
- «Отдыхай» – выключение речевого интерфейса;

- «Открыть» – переход в режим вызова той или иной программы. Имя программы называется в следующем слове;
- «Буду диктовать» – переход из режима команд в режим набора текста голосом;
- «Режим команд» – возврат в режим подачи команд голосом и т. д.

Слова должны выговариваться четко, в одном темпе. Между словами обязательна пауза. Учитывая, что алгоритмы распознавания речи недостаточно развиты, требуется индивидуальная предварительная настройка компьютерной системы на конкретного пользователя. Это простейшая реализация SILK-интерфейса.



Рисунок 37 – Модуль речевого интерфейса для автомобиля

Биометрические технологии (мимический интерфейс) основаны на биометрии, измерении уникальных характеристик отдельно взятого человека (рисунок 38). Это могут быть как уникальные признаки, полученные им с рождения (ДНК, отпечатки пальцев, радужная оболочка глаза), так и характеристики, приобретенные со временем, или же способные меняться с возрастом или внешним воздействием (почерк, голос или походка).



Рисунок 38 – Биометрические технологии

Обычно при классификации биометрических технологий выделяют две группы систем по типу используемых биометрических параметров.

Первая группа систем использует статические биометрические параметры: отпечатки пальцев, геометрию руки, сетчатку глаза и т. п.

Вторая группа систем использует для идентификации динамические параметры: динамику воспроизведения подписи или рукописного ключевого слова, голос и т. п.

Семантический (общественный) интерфейс возник еще в конце 70-х годов XX века, с развитием искусственного интеллекта. Его трудно назвать самостоятельным видом интерфейса, так как он включает в себя и интерфейс командной строки, и графический, и речевой, и мимический интерфейсы. Основной его особенностью является отсутствие команд при общении с компьютером. Запрос формируется на естественном языке в виде связанного текста и образов. По сути – это моделирование общения человека с компьютером.

С середины 90-х годов XX века публикации, относящихся к семантическому интерфейсу, уже не встречались. Похоже, что в связи с важным военным значением этих разработок (например, для автономного ведения современного боя машинами – роботами, для «семантической» криптографии) эти направления были засекречены. Информация, что эти исследования продолжаются, иногда появляется в периодической печати (обычно в разделах компьютерных новостей).

3.6 Описание тактильного (touch) интерфейса и необходимое оборудование для его реализации

Привычные способы общения человека с машиной – клавиатуры, мыши, кнопки – умирают на глазах. Последние пару лет мир захлестнула настоящая революция интерфейсов.

В тактильных интерфейсах средства отображения и позиционирования функционально объединены и пространственно совмещены.

Тактильный интерфейс описывает управление посредством касания сенсорного экрана с изображением стилусом или пальцами (рисунок 39) или иными частями тела (рисунок 40).

Tactus Technology представила на выставке SID 2012 в Бостоне новую технологию обратной связи, которая делает сенсорные дисплеи «тактильными» – Tactile Layer создает рельефные узоры

поверх обычных тачскринов, делает экранные кнопки, стрелки, полосы прокрутки и прочие формы объемными (рисунок 41).

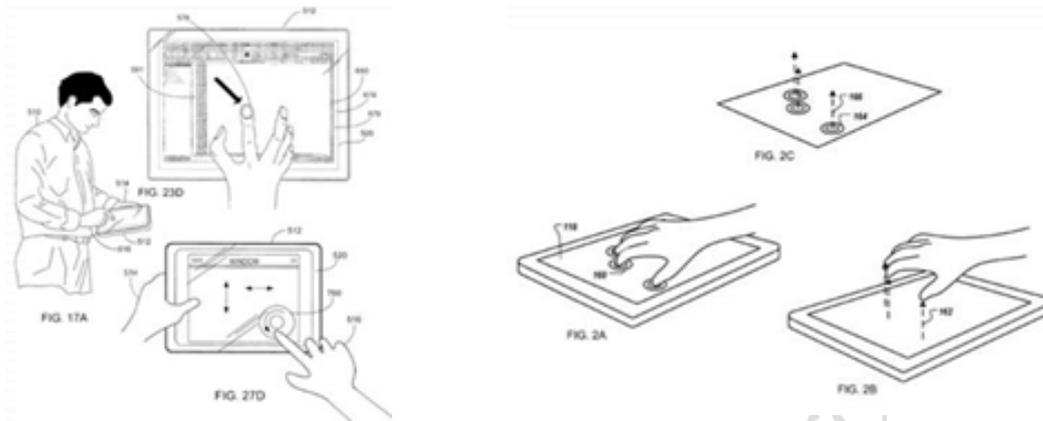


Рисунок 39 – Касание экрана с изображением пальцами

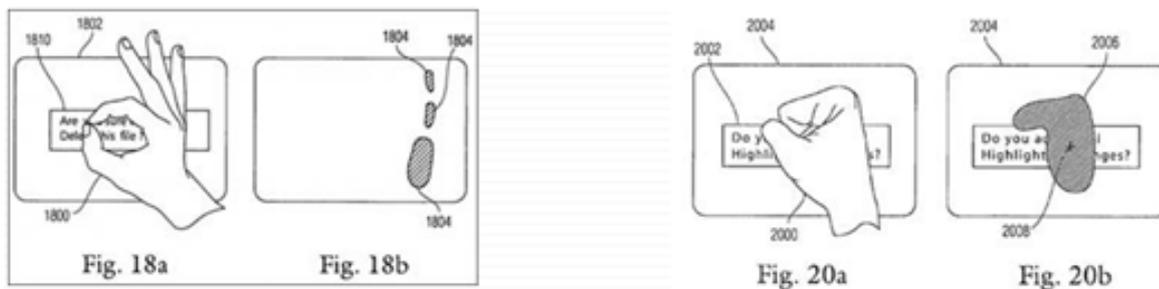


Рисунок 40 – Касание экрана с изображением различными частями тела



Рисунок 41 – Tactile Layer

Возможно, различные клавиатуры и являются наиболее распространенным и, быть может, наиболее удобным из существующих на сегодняшний день средств ввода текстовой информации, но

использовать их можно далеко не всегда. В частности, в условиях плохого освещения или при печати на ходу эффективность клавиатур представляется довольно сомнительной.

К тому же люди, страдающие от различных нарушений зрения, не могут пользоваться обычными клавиатурами, а специализированные аналоги зачастую оказываются довольно неудобными, особенно когда речь идет о портативных устройствах (рисунок 42).



Рисунок 42 – Тактильный интерфейс для незрячих

Специалисты компании Disney разработали новую электро-вибрационную технологию, которая может дать тачскринам то, чего им так не достает – тактильность. Компания Disney представила несколько демонстрационных панелей со своей новой технологией. На самом деле, покрытие сенсорного экрана не меняет форму, поскольку для этого бы понадобилось множество движущихся частей, что исказило бы само изображение. Специалисты Disney пошли другим путем. Меняя электрическое напряжение на поверхности экрана, им удалось изменять ощущения от трения пальцем по экрану, что, по существу, и является тактильной обратной связью. Таким образом, скользя пальцем по экрану можно ощутить трехмерные ребра и грани (рисунок 43).



Рисунок 43 – Разработка компании Disney

3.7 Описание мимического и семантического интерфейсов и необходимое оборудование для их реализации

Все биометрические системы работают практически по одинаковой схеме. Вначале система запоминает образец биометрической характеристики (это и называется процессом записи). Во время записи некоторые биометрические системы могут попросить сделать несколько образцов для того, чтобы составить наиболее точное изображение биометрической характеристики. Затем полученная информация обрабатывается и преобразовывается в математический код. Кроме того, система может попросить произвести ещё некоторые действия для того, чтобы «приписать» биометрический образец к определённому человеку. Например, персональный идентификационный номер (PIN) прикрепляется к определённому образцу, либо смарт-карта, содержащая образец, вставляется в считывающее устройство. В таком случае снова делается образец биометрической характеристики и сравнивается с представленным образцом. Идентификация по любой биометрической системе проходит четыре стадии:

1 Запись – физический или поведенческий образец запоминается системой.

2 Выделение – уникальная информация выносится из образца и составляется биометрический образец.

3 Сравнение – сохраненный образец сравнивается с представленным.

4 Совпадение/несовпадение – система решает, совпадают ли биометрические образцы, и выносит решение.

подавляющее большинство людей считают, что в памяти компьютера хранится образец отпечатка пальца, голоса человека или картинка радужной оболочки его глаза. Но на самом деле в большинстве современных систем это не так. В специальной базе данных хранится цифровой код длиной до 1000 бит, который ассоциируется с конкретным человеком, имеющим право доступа. Сканер или любое другое устройство, используемое в системе, считывает определённый биологический параметр человека. Далее он обрабатывает полученное изображение или звук, преобразовывая их в цифровой код. Именно этот ключ и сравнивается с содержимым специальной базы данных для идентификации личности.

3.8 Виды мобильных устройств и типы их интерфейсов

Сегодня на рынке имеется несколько основных разновидностей мобильных устройств: смартфоны, коммуникаторы и карманные компьютеры (КПК) [20].

Смартфон – это устройство, которое совмещает в себе функции КПК и телефона (рисунок 44а). Смартфоны еще называют коммуникаторами. В смартфоне установлена операционная система (Apple iOS, Windows Mobile, Windows Phone 7(Mango), Bada, Android и Symbian). Еще одной отличительной чертой смартфона является многозадачность, это означает, что параллельно могут выполняться несколько команд или быть запущены несколько приложений. В смартфонах обычно множество дополнительных функций для связи (WiFi, 3G), программы системы позиционирования (GPS, A-GPS). Существует возможность работы с документами, с планировщиком заданий, с полнофункциональным Интернетом. На операционные системы Android и Apple iOS созданы приложения, практически ничем не отличающиеся от компьютерных.



Рисунок 44 – а) смартфон; б) коммуникатор

Коммуникатор – это такое КПК (причем чаще на базе Windows Mobile) со встроенным модулем GSM (рисунок 44б). То есть он, по сути, представляет собой миниатюрный компьютер, к многочисленным возможностям которого прибавляется способность в нужный

момент превращаться в телефон. Функционируют под управлением операционных систем, открытых для сторонних пользователей. В первую очередь он предназначен для работы с данными и доступа в Интернет, функции связи для него в общем вторичны. Коммуникаторы оснащаются крупными экранами (обычно сенсорными, но бывают и исключения), емкой памятью, клавиатурой – настоящей или виртуальной, нарисованной на сенсорном экране. В общем, это настоящий мобильный офис. Устройство сочетает в себе компактный размер и многофункциональность. Подлинная разница между смартфоном и коммуникатором кроется в производительности устройств и совместимости с программным обеспечением.

Карманный персональный компьютер – это малогабаритное вычислительное устройство размером с записную книжку (рисунок 45а). На сегодняшний день в КПК «стоит» система Windows Mobile 6.0. КПК позволяет работать с офисными приложениями (просмотр документов форматов WORD, EXCEL, презентаций в программе POWER POINT, создание и редактирование текстовых файлов TXT и заметок PVI); выходить в Интернет, посылать письма по электронной почте, просматривать цветные фотографии и клипы, прослушивать MP3-файлы, смотреть видеофильмы и т. п. Теперь КПК всё чаще используют и как электронную книгу, которую можно просматривать в программах Naali Reader и AlReader2. В последние годы в разных КПК стали применять технологии беспроводного обмена данными и выхода в Интернет (ИК-порт, Wi-Fi, Bluetooth), GPS-приёмники, VGA-камеры, FM-радио и TV-тюнер. Элементом комплектования практически любого КПК является диск с программой синхронизации данных через USB- и ИК-порты. Современные КПК оснащены также слотом расширения постоянной памяти – специальными картами (SD, miniSD, microSD), с помощью которых можно переносить файлы с КПК на другие устройства.

Еще одна интересная разновидность мобильных устройств – **электронные книги** (e-book). Это фактически узкоспециализированные КПК, рассчитанные прежде всего на чтение электронных книг (рисунок 45б). В отличие от КПК они обычно имеют более крупные габариты, что связано с необходимостью иметь крупный экран со специальной матрицей.

К мобильным устройствам относятся устройства различных типов – телефоны, смартфоны, коммуникаторы, планшеты, навигаторы и так далее, то есть все устройства, имеющие установленную операционную систему для мобильных устройств.

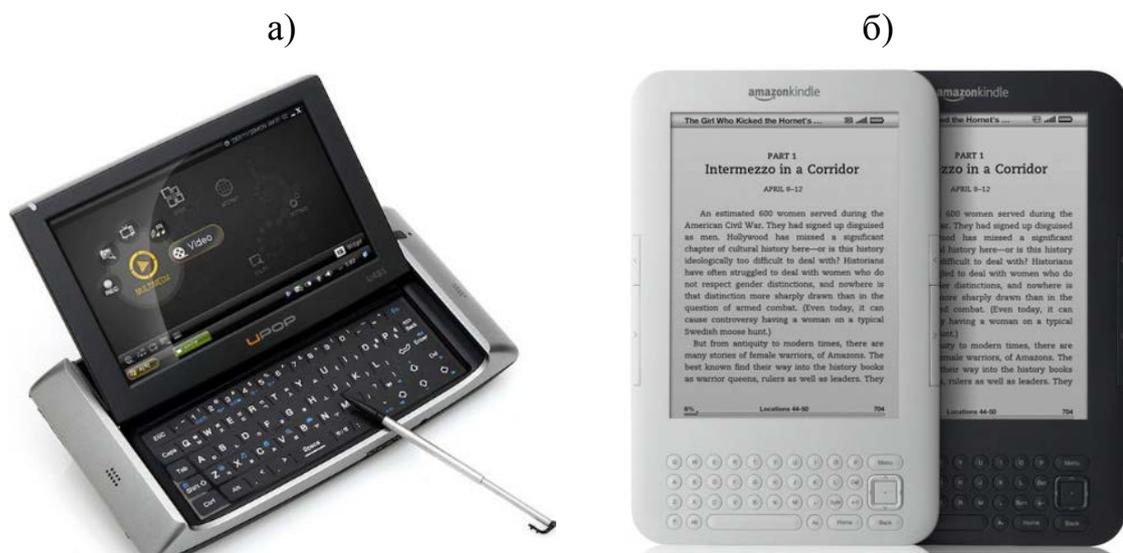


Рисунок 45 – а) карманный персональный компьютер; б) электронные книги

3.9 Командный режим ОС Linux

В Ubuntu существует два вида интерфейса: графический интерфейс пользователя и интерфейс командной строки [4].

Графический интерфейс пользователя – управление с помощью графических кнопок, меню, панелей, окон. Множество действий можно выполнять с помощью мыши.

Интерфейс командной строки – управление с помощью команд. Команды состоят из букв, цифр, символов, набираются построчно, выполняются после нажатия клавиши **Enter**. Основным инструментом здесь является клавиатура. Данный интерфейс встроен в ядро системы, он будет доступен, даже если графический интерфейс не запустится.

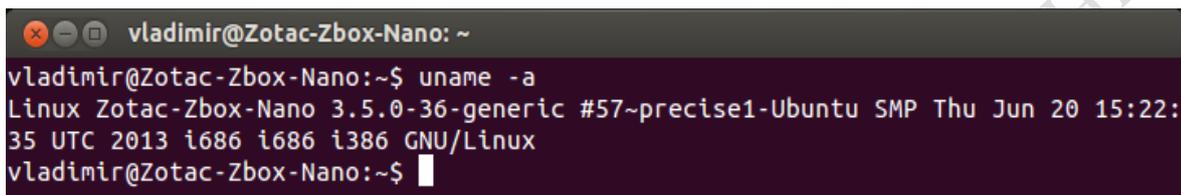
Добраться до командной строки можно двумя способами: через консоль и терминал.

Консоль. Во время загрузки Ubuntu запускаются семь полноэкранных консолей, у каждой свой независимый сеанс, с первой по шестую с интерфейсом командной строки, в седьмой запускается графический режим. Пользователь во время загрузки видит только графический режим. На рисунке 46 показана консоль в Ubuntu.

```
Ubuntu 12.04.1 LTS bismark-desktop tty1
bismark-desktop login: _
```

Рисунок 46 – Консоль в ОС Linux

Терминал – графическая программа эмулирующая консоль. Такие программы позволяют, не выходя из графического режима, выполнять команды. Терминал по сравнению с консолью имеет дополнительный функционал (различные настройки, вкладки, можно запускать много окон, управление мышью в некоторых программах, контекстное меню, главное меню, полоса прокрутки). Для того чтобы запустить эмулятор терминала, необходимо выбрать в меню **Приложения – Стандартные – Терминал**. После чего появляется строка с приглашением к вводу команд (рисунок 47).



```
vladimir@Zotac-Zbox-Nano: ~
vladimir@Zotac-Zbox-Nano:~$ uname -a
Linux Zotac-Zbox-Nano 3.5.0-36-generic #57~precise1-Ubuntu SMP Thu Jun 20 15:22:
35 UTC 2013 i686 i686 i386 GNU/Linux
vladimir@Zotac-Zbox-Nano:~$
```

Рисунок 47 – Терминал в ОС Linux

Приглашение к вводу команды в эмуляторе терминала имеет следующий формат:

`<имя пользователя>@<хост>:<текущая директория>$`

Здесь *<имя пользователя>* – это имя, которое вы использовали для входа в систему и под которым вы сейчас работаете. Затем идет значок @, за ним имя локальной системы, далее двоеточие и имя текущего каталога, в котором вы в данный момент находитесь.

В эмуляторе *gnome-terminal* при запуске терминала в качестве каталога используется значок тильда ~. В Linux символ ~ используется для сокращенного обозначения домашней директории пользователя, а именно директории */home/<имя пользователя>*.

В конце приглашения выводится знак доллара \$.

Ниже представлены основные команды Linux, которые часто используются.

Утилита *ls* предназначена, для вывода списка файлов и каталогов в текущем каталоге. Свойства команды:

- A – включить в список «скрытые» файлы;
- S – отсортировать файлы по размеру;
- t – отсортировать по дате изменения.

Команда *mkdir* предназначена для создания новой директории. Чтобы создать новую директорию, выполните *mkdir <имя директории>*.

Утилита *cp* используется для создания копии файла. Для обычного копирования используется следующий синтаксис *cp* <имя файла источника> <имя файла назначения>.

Утилита *mv* предназначена для перемещения, переименования файла. Для переименования файла используется синтаксис *mv* <имя файла> <новое имя файла>. Для перемещения файла в другую директорию используется синтаксис *mv* <имя файла> <директория>. С *mv* можно также использовать опцию *-i*, чтобы случайно не перезаписать существующий файл.

Команда *rm* предназначена для удаления файлов. Основные опции команды: *-i* – выводить запрос перед удалением каждого файла; *-r* (или *-R*) – удалить директории и их содержимое.

Команда *cd* предназначена для смены каталога, в котором вы сейчас находитесь. Например, вы сейчас находитесь в своем домашнем каталоге *~* и хотите перейти в каталог *~/information*, тогда нужно просто выполнить команду *cd information*.

Утилита *cat* выводит содержимое текстового файла. Простейший синтаксис команды *cat* <имя файла>. Утилита *cat* может использоваться для соединения конкатенации двух текстовых файлов, собственно от этого и происходит название утилиты (*catenation*, *concatenation*).

Команда *pwd* используется, чтобы вывести полный путь текущего каталога.

Системная утилита *man* выводит страницы системной документации. С ее помощью можно получить информацию по любым командам Linux. Для этого используется синтаксис *man* <имя утилиты>.

4 ОСНОВНЫЕ ЗАДАЧИ ОПЕРАЦИОННЫХ СИСТЕМ

4.1 Понятие процесса и операции над процессами

Процесс (или по-другому, задача) – абстракция, описывающая выполняющуюся программу. Для операционной системы процесс представляет собой единицу работы, заявку на потребление системных ресурсов [24].

В многозадачной (многопроцессной) системе процесс может находиться в одном из трех основных состояний: выполнение, ожидание и готовность.

В ходе жизненного цикла каждый процесс переходит из одного состояния в другое в соответствии с алгоритмом планирования процессов, реализуемым в данной операционной системе. Типичный граф состояний процесса показан на рисунке 48.

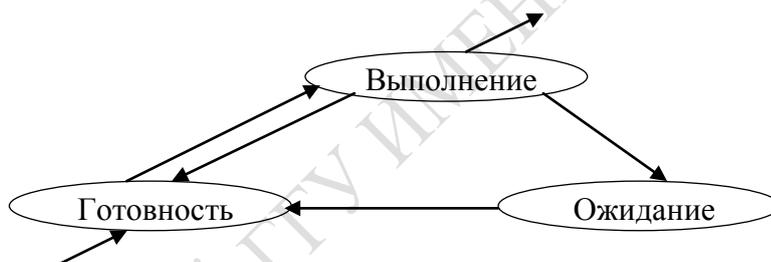


Рисунок 48 – Граф состояний процесса в многозадачной среде

В состоянии «Выполнение» в однопроцессорной системе может находиться только один процесс, а в каждом из состояний «Ожидание» и «Готовность» – несколько процессов, эти процессы образуют очереди соответственно ожидающих и готовых процессов. Жизненный цикл процесса начинается с состояния «Готовность», когда процесс готов к выполнению и ждет своей очереди. При активизации процесс переходит в состояние «Выполнение» и находится в нем до тех пор, пока либо он сам освободит процессор, перейдя в состояние «Ожидания» какого-нибудь события, либо будет насильно «вытеснен» из процессора, например, вследствие исчерпания отведенного данному процессу кванта процессорного времени. В последнем случае процесс возвращается в состояние «Готовность». В это же состояние процесс переходит из состояния «Ожидание», после того как ожидаемое событие произойдет.

4.2 Алгоритмы планирования процессов

Существует достаточно большой набор разнообразных алгоритмов планирования, которые предназначены для достижения различных целей и эффективны для разных классов задач. Многие из них могут использоваться на нескольких уровнях планирования.

Простейшим алгоритмом планирования является алгоритм **FCFS (First-Come, First-Served)**. Процессы, находящиеся в состоянии «Готовность», выстроены в очередь. Когда процесс переходит в состояние «Готовность», ссылка на его PCB помещается в конец этой очереди. Выбор нового процесса для исполнения осуществляется из начала очереди с удалением оттуда ссылки на его PCB. Очередь подобного типа называется FIFO (First In, First Out). Алгоритм осуществляется невытесняющим планированием.

Модификацией алгоритма FCFS является алгоритм, получивший название **Round Robin** или сокращенно **RR**. По сути дела, это тот же самый алгоритм, только реализованный в режиме вытесняющего планирования. Реализуется такой алгоритм так же, как и предыдущий, с помощью организации процессов, находящихся в состоянии готовности, в очередь FIFO.

Для FCFS и RR существенным является порядок расположения процессов в очереди процессов, готовых к исполнению. Если короткие задачи расположены в очереди ближе к ее началу, то общая производительность этих алгоритмов значительно возрастает. Если же таких процессов два или больше, то для выбора одного из них можно использовать алгоритм FCFS. Квантование времени при этом не применяется. Описанный алгоритм получил название «кратчайшая работа первой» или **Shortest Job First (SJF)**. SJF – алгоритм краткосрочного планирования может быть как вытесняющим, так и невытесняющим.

Гарантированное планирование. При интерактивной работе N пользователей в вычислительной системе можно применить алгоритм планирования, который гарантирует, что каждый из пользователей будет иметь в своем распоряжении $\sim 1/N$ часть процессорного времени. Пронумеруем всех пользователей от 1 до N . Для каждого пользователя с номером i введем две величины: T_i – время нахождения пользователя в системе или, другими словами, длительность сеанса его общения с машиной и τ_i – суммарное процессорное время, уже выделенное всем его процессам в течение сеанса. Справедливым для пользователя было бы получение T_i/N процессорного

времени. Если $\tau_i \ll T_i / N$, то i -й пользователь несправедливо обделен процессорным временем. Если же $\tau_i \gg T_i / N$, то система явно благоволит к пользователю с номером i . Вычислим для процессов каждого пользователя значение коэффициента справедливости $\tau_i N / T_i$ и будем предоставлять очередной квант времени готовому процессу с наименьшей величиной этого отношения. Предложенный алгоритм называют алгоритмом гарантированного планирования. К недостаткам этого алгоритма можно отнести невозможность предугадать поведение пользователей.

При **приоритетном планировании** каждому процессу присваивается определенное числовое значение – приоритет, в соответствии с которым ему выделяется процессор. Планирование с использованием приоритетов может быть как вытесняющим, так и невытесняющим.

Подход, получивший название **многоуровневых очередей**, повышает гибкость планирования: для процессов с различными характеристиками применяется наиболее подходящий им алгоритм.

Дальнейшим развитием алгоритма многоуровневых очередей является добавление к нему механизма обратной связи (рисунок 49). Здесь процесс не постоянно приписан к определенной очереди, а может мигрировать из одной очереди в другую в зависимости от своего поведения.

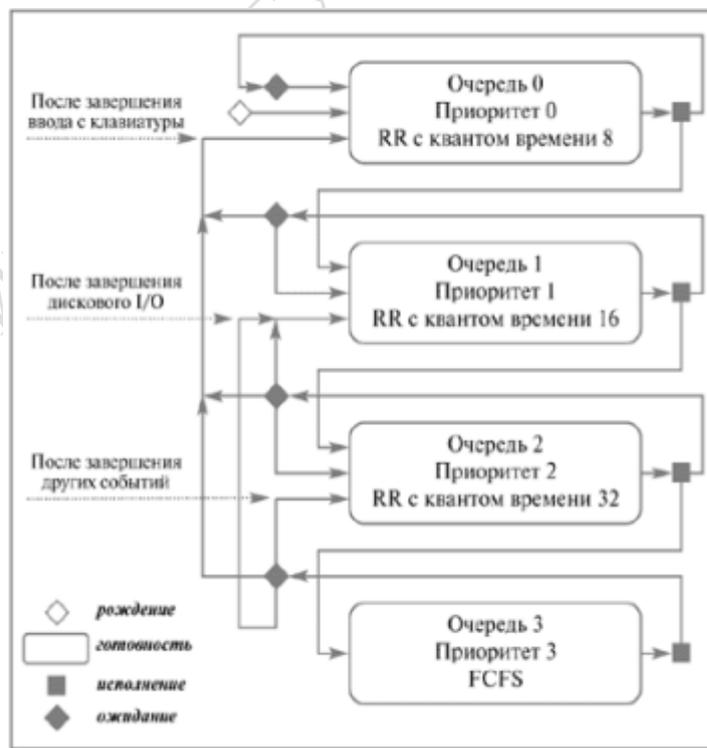


Рисунок 49 – Схема миграции процессов в многоуровневых очередях планирования с обратной связью

4.3 Вытесняющие и невытесняющие алгоритмы планирования

Алгоритмы планирования можно разделить на два класса: вытесняющие и невытесняющие алгоритмы [11].

Невытесняющие алгоритмы основаны на том, что активному потоку позволяется выполняться, пока он сам, по собственной инициативе, не отдаст управление ОС, для того чтобы та выбрала из очереди другой готовый к выполнению поток.

Вытесняющие алгоритмы – это такие способы планирования потоков, в которых решение о переключении процессора с выполнения одного потока на выполнение другого принимается ОС, а не активной задачей.

Основным различием между вытесняющими и невытесняющими алгоритмами является степень централизации механизма планирования потоков. При вытесняющем мультипрограммировании функции планирования потоков целиком сосредоточены в ОС, и программист пишет свое приложение, не заботясь о том, как оно будет выполняться параллельно с другими задачами.

При этом ОС выполняет следующие функции:

- определяет момент снятия с выполнения активного потока;
- запоминает его контекст;
- выбирает из очереди готовых к выполнению потоков следующий;
- запускает новый поток на выполнение, загружая его контекст.

При невытесняющем мультипрограммировании механизм планирования распределен между ОС и прикладными программами. Прикладная программа, получив управление от ОС, сама определяет момент завершения очередного цикла своего выполнения и только затем передает управление ОС. ОС формирует очереди потоков и выбирает в соответствии с некоторым правилом следующий поток на выполнение. Такой механизм создает проблемы как для пользователей, так и для разработчиков приложений. Для пользователей это означает, что управление системой теряется на произвольный период времени, который определяется приложением. Если приложение тратит слишком много времени на выполнение какой-либо работы, пользователь не может переключиться на другую задачу, например, на текстовый редактор, в то время как форматирование продолжалось бы в фоновом режиме.

Примером эффективного использования невытесняющей многозадачности является файл-сервер NetWare, в котором достигнута

высокая скорость выполнения файловых операций (рисунок 50). В соответствии с концепцией невытесняющего планирования, чтобы не занимать процессор слишком долго, поток NetWare сам отдает управление планировщику ОС, используя следующие вызовы:

- ThreadSwitch – поток, вызвавший эту функцию, считает себя готовым к немедленному выполнению, но отдает управление для того, чтобы могли выполняться и другие потоки;

- ThreadSwitchWithDelay – функция аналогична предыдущей, но поток считает, что будет готов к выполнению только через определенное количество переключений с потока на поток;

- Delay – функция аналогична предыдущей, но задержка дается в миллисекундах;

- ThreadSwitchLowPriority – функция отдачи управления, отличается от ThreadSwitch тем, что поток просит поместить его в очередь готовых к выполнению, но низкоприоритетных потоков.

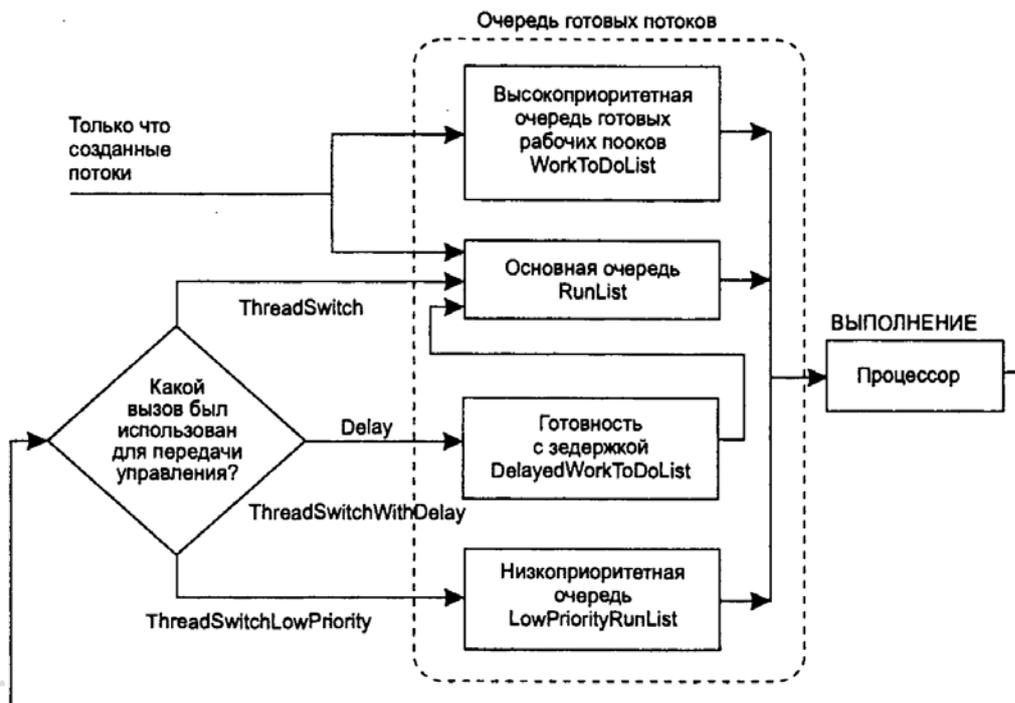


Рисунок 50 – Схема планирования потоков NetWare

4.4 Методы решения проблемы синхронизации

Процессам часто нужно взаимодействовать друг с другом, например, один процесс может передавать данные другому процессу, или несколько процессов могут обрабатывать данные из общего файла.

Во всех этих случаях возникает проблема синхронизации процессов, которая может решаться приостановкой и активизацией процессов, организацией очередей, блокированием и освобождением ресурсов.

Пренебрежение вопросами синхронизации в режиме мультипрограммирования может привести к неправильному решению задачи или даже к краху системы.

На рисунке 51 показан пример необходимости синхронизации.

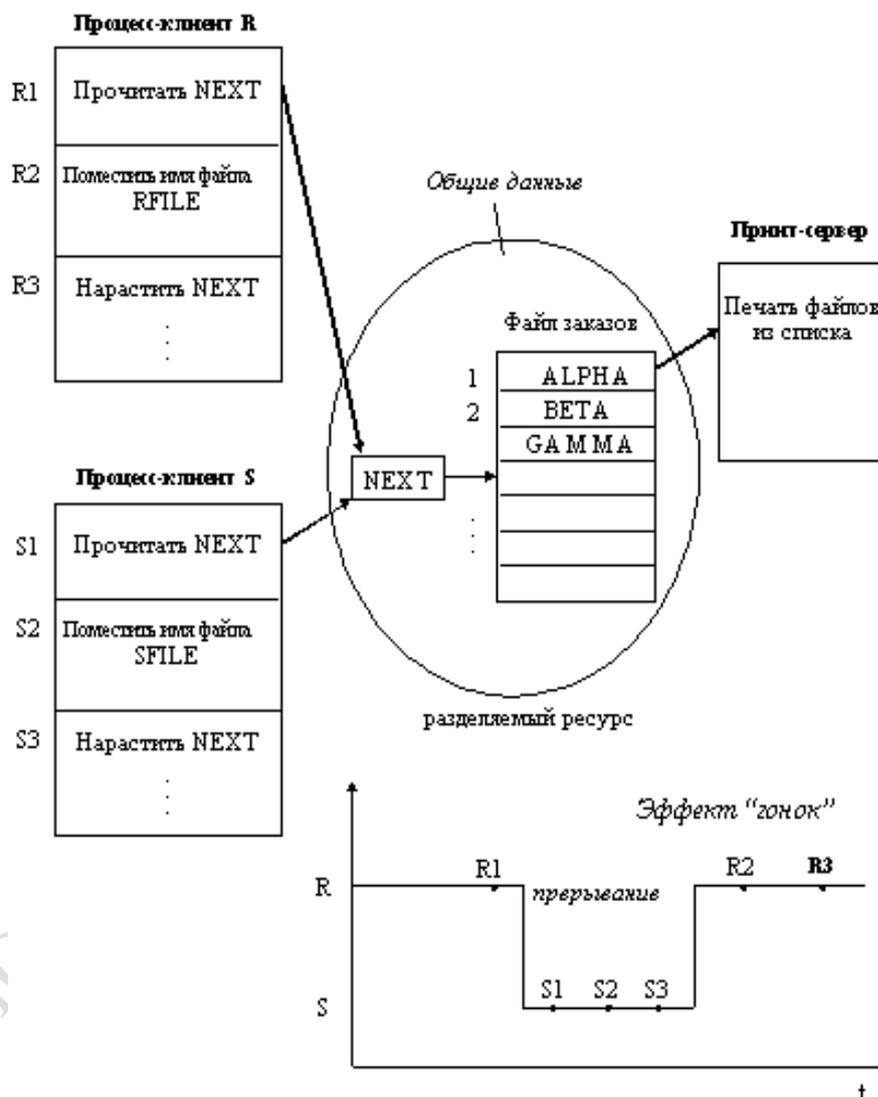


Рисунок 51 – Пример необходимости синхронизации

Голодание – задержка времени от пробуждения потока до его вызова на процессор, в течение которого он находится в списке потоков, готовых к исполнению. Возникает по причине присутствия потоков с большими или равными приоритетами, которые

исполняются все это время. Негативный эффект заключается в том, что возникает задержка времени от пробуждения потока до исполнения им следующей важной операции, что задерживает исполнение этой операции, а следом за ней и работу многих других компонентов. Голодание создаёт узкое место в системе и не дает выжать из неё максимальную производительность, ограничиваемую только аппаратно обусловленными узкими местами.

Гонка – недетерминированный порядок исполнения двух путей кода, работающих с одними и теми же данными и исполняемыми в двух различных нитях. Приводит к зависимости порядка и правильности исполнения от случайных факторов. Устраняется добавлением необходимых блокировок и примитивов синхронизации. Обычно является легко устраняемым дефектом.

Инверсия приоритета. Поток L имеет низкий приоритет, поток M – средний, поток H – высокий. Поток L захватывает mutex, и, выполняясь с удержанием mutex'a, преемптивно прерывается потоком M , который пробудился по какой-то причине и имеет более высокий приоритет. Поток H пытается захватить mutex. В полученной ситуации поток H ожидает завершения текущей работы потоком M , ибо, пока поток M исполняется, низкоприоритетный поток L не получает управления и не может освободить mutex.

4.5 Понятие нити и основные отличия от процесса

Точно так же, как многозадачная операционная система может делать несколько вещей одновременно при помощи разных процессов, один процесс может делать много вещей при помощи нескольких нитей [24].

Каждая нить представляет собой независимо выполняющийся поток управления со своим счетчиком команд, регистровым контекстом и стеком.

Понятия процесса и нити очень тесно связаны и поэтому трудноотличимы, нити даже часто называют легковесными процессами.

Основные отличия процесса от нити заключаются в том, что, каждому процессу соответствует своя, независимая от других, область памяти, таблица открытых файлов, текущая директория и прочая информация уровня ядра. Нити же не связаны непосредственно с этими сущностями. Ситуацию с нитями хорошо иллюстрирует рисунок 52:

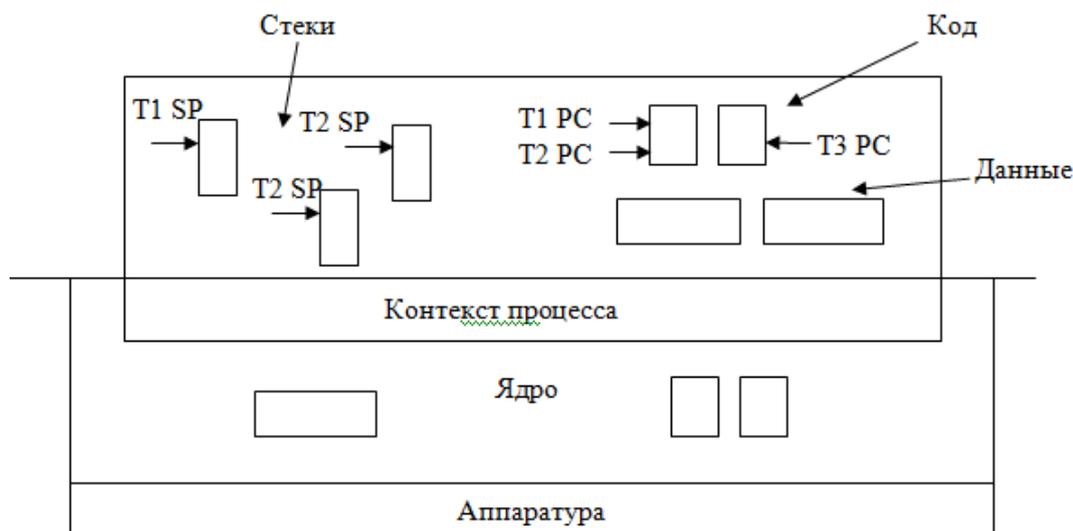


Рисунок 52 – Многозадачная операционная система с нитями

У всех нитей, принадлежащих данному процессу, всё выше перечисленное общее, поскольку принадлежит этому процессу. Кроме того, процесс всегда является сущностью уровня ядра, то есть ядро знает о его существовании, в то время как нити зачастую являются сущностями уровня пользователя и ядро может ничего не знать о ней. В подобных реализациях все данные о нити хранятся в пользовательской области памяти, и соответственно такие процедуры, как порождение или переключение между нитями, не требуют обращения к ядру и занимают на порядок меньше времени.

Хорошо видно, что все нити данного процесса, разделяют состояние процесса, данные и код, но каждая нить имеет свой счетчик команд и стек, видимый, впрочем, другим нитям.

4.6 Контекст и дескриптор процесса

На протяжении существования процесса его выполнение может быть многократно прервано и продолжено. Для того, чтобы возобновить выполнение процесса, необходимо восстановить состояние его операционной среды. Состояние операционной среды отображается состоянием регистров и программного счетчика, режимом работы процессора, указателями на открытые файлы, информацией о незавершенных операциях ввода-вывода, кодами ошибок выполняемых данным процессом системных вызовов и т. д. Эта информация называется *контекстом процесса*.

Кроме этого, операционной системе для реализации планирования процессов требуется дополнительная информация: идентификатор процесса, состояние процесса, данные о степени привилегированности процесса, место нахождения кодового сегмента и другая информация. В некоторых ОС (например, в ОС UNIX) информацию такого рода, используемую ОС для планирования процессов, называют *дескриптором процесса*.

Дескриптор процесса по сравнению с контекстом содержит более оперативную информацию, которая должна быть легко доступна подсистеме планирования процессов. Контекст процесса содержит менее актуальную информацию и используется операционной системой только после того, как принято решение о возобновлении прерванного процесса.

Очереди процессов представляют собой дескрипторы отдельных процессов, объединенные в списки. Таким образом, каждый дескриптор, кроме всего прочего, содержит по крайней мере один указатель на другой дескриптор, соседствующий с ним в очереди. Такая организация очередей позволяет легко их переупорядочивать, включать и исключать процессы, переводить процессы из одного состояния в другое.

Программный код только тогда начнет выполняться, когда для него операционной системой будет создан процесс.

Создать процесс – это значит:

- 1) создать информационные структуры, описывающие данный процесс, то есть его дескриптор и контекст;
- 2) включить дескриптор нового процесса в очередь готовых процессов;
- 3) загрузить кодовый сегмент процесса в оперативную память или в область свопинга.

Контекстом процесса является его состояние, определяемое текстом, значениями глобальных переменных пользователя и информационными структурами, значениями используемых машинных регистров, значениями, хранимыми в позиции таблицы процессов и в адресном пространстве задачи, а также содержимым стеков задачи и ядра, относящихся к данному процессу. Текст операций системы и ее глобальные информационные структуры совместно используются всеми процессами, но не являются составной частью контекста процесса.

Говорят, что при запуске процесса система исполняется в контексте процесса. Когда ядро системы решает запустить другой процесс,

оно выполняет переключение контекста, с тем, чтобы система исполнялась в контексте другого процесса. Ядро осуществляет переключение контекста только при определенных условиях, что мы увидим в дальнейшем. Выполняя переключение контекста, ядро сохраняет информацию, достаточную для того, чтобы позднее переключиться вновь на первый процесс и возобновить его выполнение. Аналогичным образом, при переходе из режима задачи в режим ядра, ядро системы сохраняет информацию, достаточную для того, чтобы позднее вернуться в режим задачи и продолжить выполнение с прерванного места. Однако, переход из режима задачи в режим ядра является сменой режима, но не переключением контекста. Если обратиться еще раз к рисунку 53, можно сказать, что ядро выполняет переключение контекста, когда меняет контекст процесса А на контекст процесса В; оно меняет режим выполнения с режима задачи на режим ядра и, наоборот, оставаясь в контексте одного процесса, например, процесса А.

Ядро обрабатывает прерывания в контексте прерванного процесса, пусть даже оно и не вызывало никакого прерывания. Прерванный процесс мог при этом выполняться как в режиме задачи, так и в режиме ядра. Ядро сохраняет информацию, достаточную для того, чтобы можно было позже возобновить выполнение прерванного процесса, и обрабатывает прерывание в режиме ядра. Ядро не порождает и не планирует порождение какого-то особого процесса по обработке прерываний.

	Процессы			
	А	В	С	Д
Режим ядра	Я	.	.	Я
Режим задачи	.	З	З	.

Рисунок 53 – Процессы и режимы их выполнения

4.7 Классификация методов распределения данных

Все алгоритмы распределения памяти разделены на два класса: алгоритмы, в которых используется перемещение сегментов процессов между оперативной памятью и диском, и алгоритмы, в которых внешняя память не привлекается [21] (рисунок 54).



Рисунок 54 – Классификация методов распределения памяти

Распределение памяти фиксированными разделами состоит в том, что память разбивается на несколько областей фиксированной величины, называемых разделами. Такое разбиение может быть выполнено вручную оператором во время старта системы или во время ее установки. После этого границы разделов не изменяются. Очередной новый процесс, поступивший на выполнение, помещается либо в общую очередь, либо в очередь к некоторому разделу.

Распределение памяти перемещаемыми разделами состоит в перемещении всех занятых участков в сторону старших и младших адресов, так, чтобы вся свободная память образовала единую свободную область. При завершении процесса, или когда не хватает памяти для загрузки очередного процесса, выполняется процедура сжатия.

Динамическое распределение памяти заключается в том, что каждому вновь поступающему на выполнение приложению на этапе создания процесса выделяется вся необходимая ему память. После завершения процесса память освобождается, и на это место может быть загружен другой процесс.

Страничная виртуальная память организует перемещение данных между ОП и внешней памятью страницами – частями виртуального адресного пространства фиксированного и сравнительно небольшого размера. Виртуальное адресное пространство делится на виртуальные страницы, размерами, равными степени 2 (1024, 4096 и т. д.).

Сегментная виртуальная память предусматривает перемещение данных сегментами – частями виртуального адресного пространства

произвольного размера. Виртуальный адрес определяется номером сегмента и смещением внутри сегмента; физический – сложением базового адреса сегмента и смещения в сегменте. Сегменты имеют разные размеры, базовые адреса не являются степенями 2.

В сегментно-страничном распределении виртуальное адресное пространство делится на сегменты, а затем сегменты делятся на страницы. Единица перемещения – страница. Виртуальное адресное пространство процесса разделено на виртуальные сегменты. Все сегменты образуют одно виртуальное пространство. Виртуальный адрес получается сложением начального виртуального адреса сегмента со смещением, а физический – механизмом перехода от виртуальных страниц к физическим.

4.8 Распределение памяти фиксированными разделами

Простейший способ управления ОП состоит в том, что память разбивается на несколько областей фиксированной величины, называемых разделами. Такое разбиение может быть выполнено вручную оператором во время старта системы или во время ее установки. После этого границы разделов не изменяются.

Очередной новый процесс, поступивший на выполнение, помещается либо в общую очередь, либо в очередь к некоторому разделу (рисунок 55).

Подсистема управления памятью выполняет следующие задачи:

- сравнивает объем памяти, требуемый для вновь поступившего процесса, с размерами свободных разделов и выбирает подходящий раздел;

- осуществляет загрузку подпрограммы в один из разделов и настройку адресов; на этапе трансляции разработчик программы может задать раздел, в котором ее следует выполнять; это позволяет сразу получить машинный код, настроенный на конкретную область памяти.

При очевидном преимуществе – простоте реализации – этот метод имеет существенный недостаток – жесткость. Так как в каждом разделе может выполняться только одна программа, то уровень мультипрограммирования заранее ограничен числом разделов независимо от того, какой размер имеют программы. Даже если программа имеет небольшой объем, она будет занимать весь раздел, что приводит к неэффективному использованию памяти. С другой

стороны, даже если объем оперативной памяти машины позволяет выполнить некоторую программу, разбиение памяти на разделы не даст возможности сделать этого.

Такой способ управления памятью применялся в ранних мультипрограммных ОС. Однако и сейчас метод распределения памяти фиксированными разделами находит применение в системах реального времени, в основном благодаря небольшим затратам на реализацию. Детерминированность вычислительного процесса систем реального времени (заранее известен набор выполняемых задач, их требования к памяти, а иногда и моменты запуска) компенсирует недостаточную гибкость данного способа управления памятью (рисунок 55).

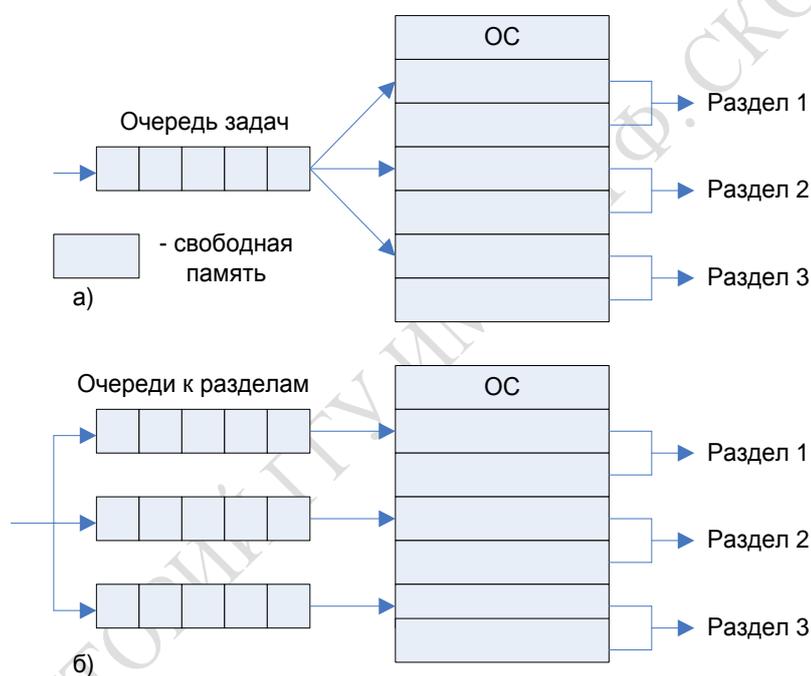


Рисунок 55 – Распределение памяти фиксированными разделами:
а) с общей очередью; б) с отдельными очередями

4.9 Распределение памяти разделами переменной величины (динамически)

Вначале вся память, отводимая для приложений, свободна. Каждому вновь поступающему на выполнение приложению на этапе создания процесса выделяется вся необходимая ему память. После завершения процесса память освобождается, и на это место может быть загружен другой процесс. Таким образом, в произвольный

момент времени ОП представляет собой случайную последовательность занятых и свободных участков произвольного размера.

На рисунке 56 показано состояние памяти в различные моменты времени при использовании динамического распределения. Так, в момент t_0 , в памяти находится только ОС, а к моменту t_1 память разделена между пятью процессами, причем процесс П4, завершаясь, покидает память. На освободившееся от процесса П4 место загружается процесс П6, поступивший в момент t_3 .

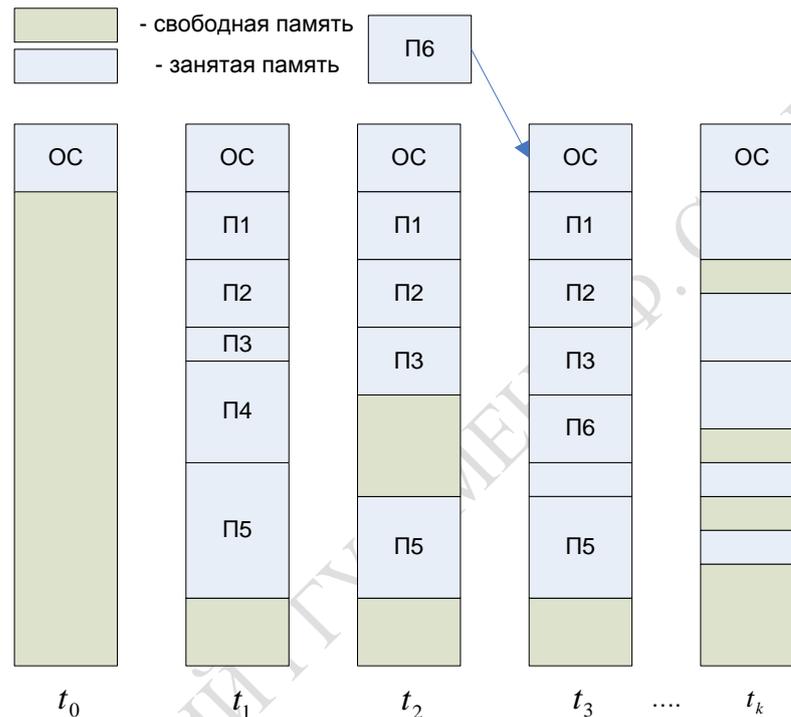


Рисунок 56 – Распределение памяти разделами переменной величины (динамически)

Функции операционной системы, предназначенные для реализации данного метода управления памятью, перечислены ниже.

1 Введение таблиц свободных и занятых областей, в которых указываются начальные адреса и размеры участков памяти.

2 При создании нового процесса – анализ требований к памяти, просмотр таблицы свободных областей и выбор раздела, размер которого достаточен для размещения кодов и данных нового процесса. Выбор раздела может осуществляться по разным правилам, например: «первый попавшийся раздел достаточного размера», «Раздел, имеющий достаточный размер» или «Раздел, имеющий небольшой достаточный размер».

3 Загрузка программы в выделенный ей раздел и корректировка таблиц свободных и занятых областей. Данный способ предполагает, что программный код не перемещается во время выполнения, а значит, настройка адресов может быть проведена одновременно во время загрузки.

4 После завершения процесса корректировка свободных и занятых областей памяти.

По сравнению с методом распределения памяти фиксированными разделами данный метод обладает гораздо большей гибкостью, но ему присущ очень серьезный недостаток – фрагментация памяти.

Фрагментация – это наличие большого числа несмежных участков свободной памяти очень маленького размера (фрагментов). Настолько маленького, что ни одна из вновь поступающих программ не может поместиться ни в одном из участков, хотя суммарный объем фрагментов может составить значительную величину, намного превышающую требуемый объем памяти.

Распределение памяти динамическими разделами лежит в основе подсистем управления памятью многих мультипрограммных систем 60–70-х годов, в частности такой ОС, как OS/360.

4.10 Распределение памяти перемещаемыми разделами

Одним из методов борьбы с фрагментацией является перемещение всех занятых участков в сторону старших и младших адресов так, чтобы вся свободная память образовала единую свободную память (рисунок 57). Операционная система должна время от времени копировать содержимое разделов из одного места памяти в другое, корректируя таблицы свободных и занятых областей. Эта процедура называется сжатием. Сжатие может выполняться либо при каждом завершении процесса, либо тогда, когда для вновь создаваемого процесса нет свободного раздела достаточного размера.

В первом случае требуется меньше вычислительной работы при корректировке таблиц свободных и занятых областей, а во втором – реже выполняется процедура сжатия. Другими словами, невозможно выполнить настройку адресов с помощью перемещаемого загрузчика. Здесь более подходящим оказывается динамическое образование адресов. Так как программы перемещаются по оперативной памяти в ходе своего выполнения, то невозможно выполнить настройку

адресов с помощью перемещаемого загрузчика. Здесь более подходящим оказывается динамическое образование адресов.

Хотя процедура сжатия и приводит к более эффективному использованию памяти, она может потребовать значительного времени, что часто превышает преимущества данного метода.

Концепция сжатия применяется и при использовании других методов распределения памяти, когда отдельному процессу выделяется не одна сплошная область памяти, а несколько несмежных участков памяти произвольного размера (сегментов). Такой подход был использован в ранних версиях OS/2, в которых память распределялась сегментами, а возникшая при этом фрагментация устранялась путем периодического перемещения сегментов.

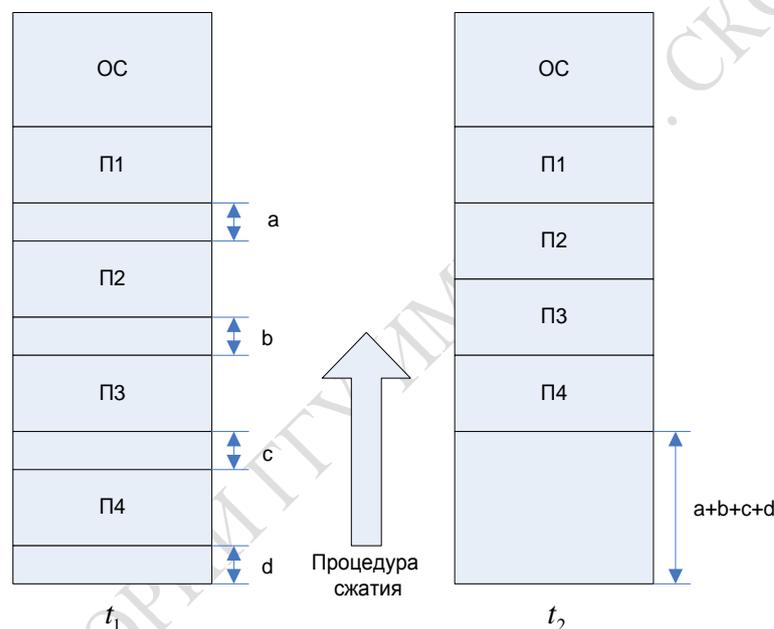


Рисунок 57 – Распределение памяти перемещаемыми разделами

4.11 Понятие виртуальной памяти

Уже достаточно давно пользователи столкнулись с проблемой размещения в памяти программ, размер которых превышал имеющуюся в наличии свободную память. Решением было разбиение программы на части, называемые оверлеями [13]. 0-й оверлей начал выполняться первым. Когда он заканчивал свое выполнение, то вызывал другой оверлей. Все оверлеи хранились на диске и перемещались между памятью и диском средствами операционной системы. Однако разбиение программы на части и планирование

их загрузки в оперативную память должен был осуществлять программист.

Развитие методов организации вычислительного процесса в этом направлении привело к появлению метода, известного под названием *виртуальная память*. Виртуальным называется ресурс, который пользователю или пользовательской программе представляется обладающими свойствами, которые в действительности ему не принадлежат. Так, например, пользователю может быть предоставлена виртуальная оперативная память, размер которой превосходит всю имеющуюся в системе реальную оперативную память. Пользователь пишет программы так, как будто в его распоряжении имеется однородная оперативная память большого объема, но в действительности все данные, используемые программой, хранятся на одном или нескольких разнородных запоминающих устройствах, обычно на дисках, и при необходимости частями отображаются в реальную память.

Таким образом, виртуальная память – это совокупность программно-аппаратных средств, позволяющих пользователям писать программы, размер которых превосходит имеющуюся оперативную память; для этого виртуальная память решает следующие задачи:

- размещает данные в запоминающих устройствах разного типа, например, часть программы в оперативной памяти, а часть на диске;
- перемещает по мере необходимости данные между запоминающими устройствами разного типа, например, подгружает нужную часть программы с диска в оперативную память;
- преобразует виртуальные адреса в физические.

Все эти действия выполняются автоматически, без участия программиста, то есть механизм виртуальной памяти является прозрачным по отношению к пользователю.

4.12 Страничное распределение

На рисунке 58 показана схема страничного распределения памяти. 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 является кратным размеру страницы, поэтому последняя страница каждого процесса дополняется фиктивной областью.

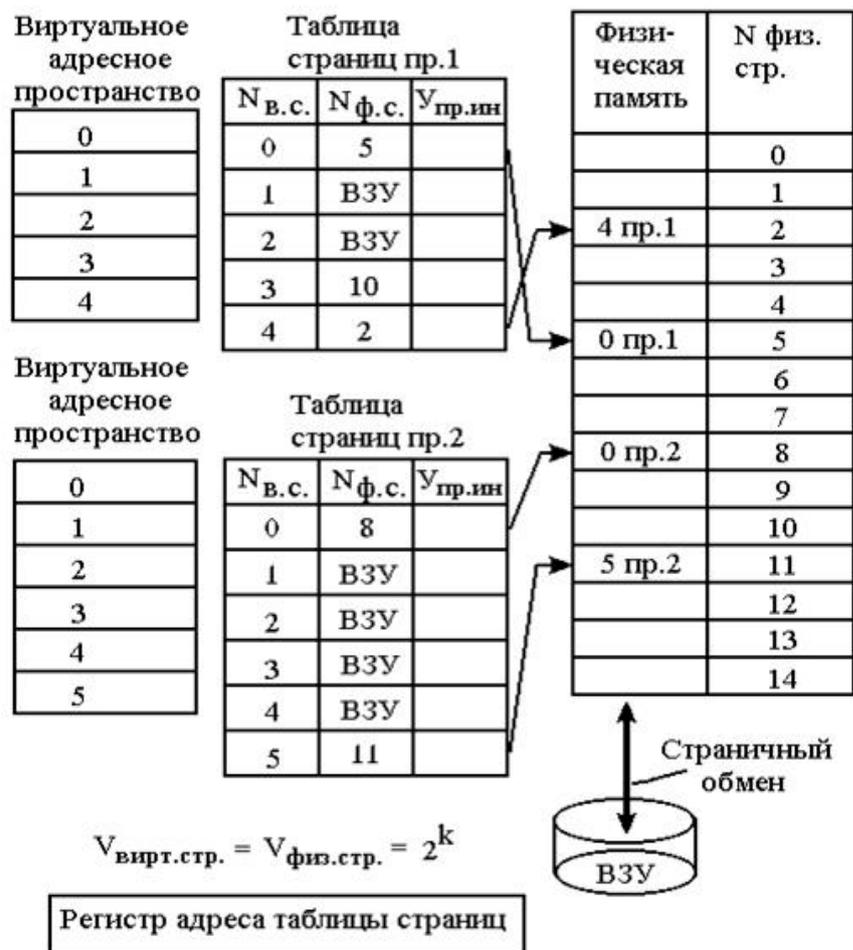


Рисунок 58 – Страничное распределение памяти

Вся оперативная память машины также делится на части такого же размера, называемые физическими страницами (или блоками). Размер страницы обычно выбирается равным степени двойки: 512, 1024 и так далее, это позволяет упростить механизм преобразования адресов.

При загрузке процесса часть его виртуальных страниц помещается в оперативную память, а остальные – на диск. Смежные виртуальные страницы не обязательно располагаются в смежных физических страницах. При загрузке операционная система создает для каждого процесса информационную структуру – таблицу страниц, в которой устанавливается соответствие между номерами виртуальных и физических страниц для страниц, загруженных в оперативную память, или делается отметка о том, что виртуальная страница выгружена на диск. Кроме того, в таблице страниц содержится управляющая информация, такая как признак модификации страницы, признак невыгружаемости (выгрузка некоторых страниц может

быть запрещена), признак обращения к странице (используется для подсчета числа обращений за определенный период времени) и другие данные, формируемые и используемые механизмом виртуальной памяти.

При активизации очередного процесса в специальный регистр процессора загружается адрес таблицы страниц данного процесса.

При каждом обращении к памяти происходит чтение из таблицы страниц информации о виртуальной странице, к которой произошло обращение. Если данная виртуальная страница находится в оперативной памяти, то выполняется преобразование виртуального адреса в физический. Если же нужная виртуальная страница в данный момент выгружена на диск, то происходит так называемое страничное прерывание. Выполняющийся процесс переводится в состояние ожидания, и активизируется другой процесс из очереди готовых. Параллельно программа обработки страничного прерывания находит на диске требуемую виртуальную страницу и пытается загрузить ее в оперативную память. Если в памяти имеется свободная физическая страница, то загрузка выполняется немедленно, если же свободных страниц нет, то решается вопрос, какую страницу следует выгрузить из оперативной памяти.

В данной ситуации может быть использовано много разных критериев выбора, наиболее популярные из них следующие:

- дольше всего не использовавшаяся страница,
- первая попавшаяся страница,
- страница, к которой в последнее время было меньше всего обращений.

В некоторых системах применяется понятие рабочего множества страниц. Рабочее множество определяется для каждого процесса и представляет собой перечень наиболее часто используемых страниц, которые должны постоянно находиться в оперативной памяти и поэтому не подлежат выгрузке.

После того как выбрана страница, которая должна покинуть оперативную память, анализируется ее признак модификации (из таблицы страниц). Если выталкиваемая страница с момента загрузки была модифицирована, то ее новая версия должна быть переписана на диск. Если нет, то она может быть просто уничтожена, то есть соответствующая физическая страница объявляется свободной.

Рассмотрим механизм преобразования виртуального адреса в физический при страничной организации памяти (рисунок 59).

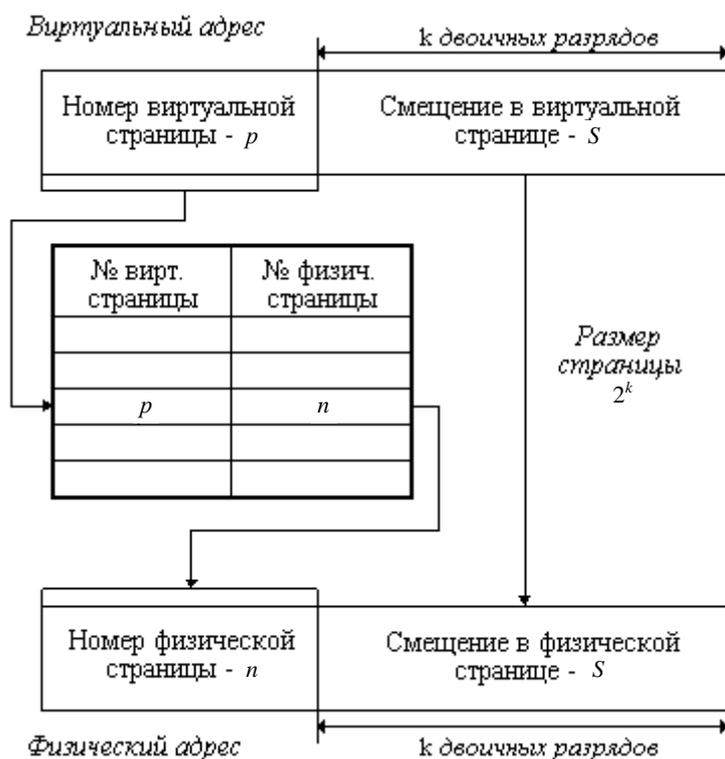


Рисунок 59 – Механизм преобразования виртуального адреса в физический при страничной организации памяти

Виртуальный адрес при страничном распределении может быть представлен в виде пары (p, S) , где p – номер виртуальной страницы процесса (нумерация страниц начинается с 0), а S – смещение в пределах виртуальной страницы. Учитывая, что размер страницы равен 2 в степени k , смещение S может быть получено простым отделением k младших разрядов в двоичной записи виртуального адреса. Оставшиеся старшие разряды представляют собой двоичную запись номера страницы p .

При каждом обращении к оперативной памяти аппаратными средствами выполняются следующие действия:

- 1) на основании начального адреса таблицы страниц (содержимое регистра адреса таблицы страниц), номера виртуальной страницы (старшие разряды виртуального адреса) и длины записи в таблице страниц (системная константа) определяется адрес нужной записи в таблице;
- 2) из этой записи извлекается номер физической страницы;
- 3) к номеру физической страницы присоединяется смещение (младшие разряды виртуального адреса).

Использование в пункте (3) того факта, что размер страницы равен степени 2 , позволяет применить операцию конкатенации

(присоединения) вместо более длительной операции сложения, что уменьшает время получения физического адреса, а значит повышает производительность компьютера.

На производительность системы со страничной организацией памяти влияют временные затраты, связанные с обработкой страничных прерываний и преобразованием виртуального адреса в физический. При часто возникающих страничных прерываниях система может тратить большую часть времени впустую, на свопинг страниц. Чтобы уменьшить частоту страничных прерываний, следовало бы увеличивать размер страницы. Кроме того, увеличение размера страницы уменьшает размер таблицы страниц, а значит, уменьшает затраты памяти. С другой стороны, если страница велика, значит велика и фиктивная область в последней виртуальной странице каждой программы. В среднем на каждой программе теряется половина объема страницы, что в сумме при большой странице может составить существенную величину. Время преобразования виртуального адреса в физический в значительной степени определяется временем доступа к таблице страниц. В связи с этим таблицу страниц стремятся размещать в «быстрых» запоминающих устройствах. Это может быть, например, набор специальных регистров или память, использующая для уменьшения времени доступа ассоциативный поиск и кэширование данных.

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 оперативной памяти. Такой вариант страничной организации, хотя и не предоставляет пользователю виртуальной памяти, но почти исключает фрагментацию за счет того, что программа может загружаться в несмежные области, а также того, что при загрузке виртуальных страниц никогда не образуется остатков.

4.13 Сегментное распределение памяти

Виртуальное адресное пространство процесса делится на части – сегменты, размер которых определяется с учетом смыслового значения содержащейся в них информации (рисунок 60). Максимальный размер сегмента определяется разрядностью виртуального адреса, например при 32-разрядной организации процесса он равен 4 Гбайт. При этом максимально возможное виртуальное пространство

процесса представляет собой набор из N виртуальных сегментов, каждый размером по 4 Гбайта. Виртуальный адрес задается номером сегмента и линейным виртуальным адресом внутри сегмента.

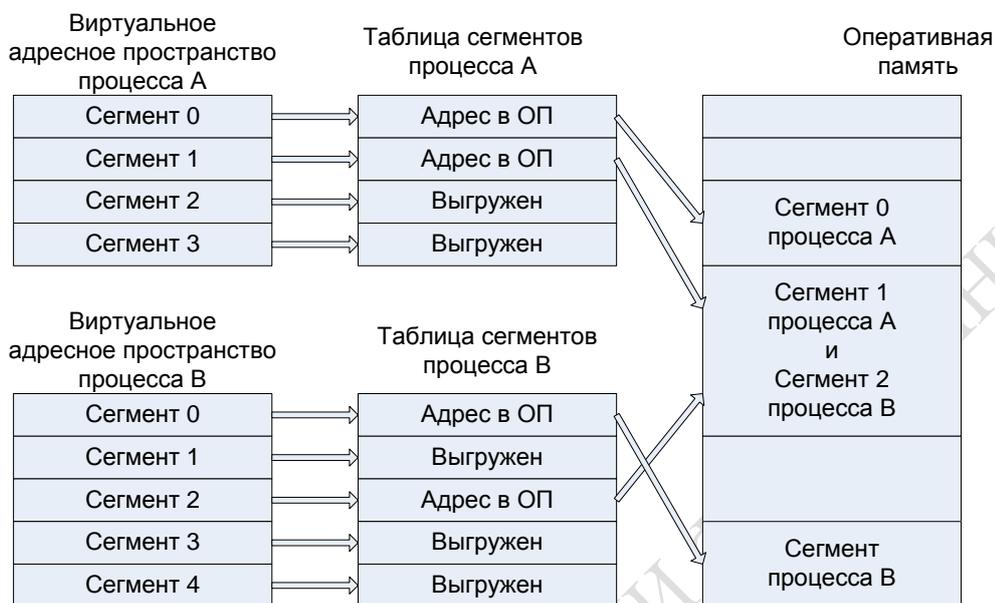


Рисунок 60 – Сегментное распределение памяти

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 достаточного размера. Если во время выполнения процесса происходит обращение по виртуальному адресу, относящемуся к сегменту, который в данный момент отсутствует в памяти, то происходит прерывание. ОС приостанавливает активный процесс, запускает на выполнение следующий процесс из очереди, а параллельно организует загрузку нужного сегмента с диска.

При отсутствии в памяти места, необходимого для загрузки сегмента, ОС выбирает сегмент на выгрузку.

На этапе создания процесса во время загрузки его образа в ОП система создает таблицу сегментов процесса, в которой для каждого сегмента указывается:

- базовый физический адрес сегмента в ОП;
- размер сегмента;
- правила доступа к сегменту;
- признаки модификации, присутствия и обращения к данному сегменту, а также некоторая другая информация.

Если виртуальные адресные пространства нескольких процессов включают один и тот же сегмент, то в таблицах сегментов этих процессов делаются ссылки на один и тот же участок ОП, в которой данный сегмент загружается в единственном экземпляре.

Недостатком указанного метода распределения памяти является фрагментация на уровне сегментов и более медленное по сравнению со страничной организацией преобразование адреса.

4.14 Сегментно-страничное распределение памяти

Данный метод представляет собой комбинацию страничного и сегментного распределения памяти и вследствие этого сочетает в себе достоинства обоих подходов. Виртуальное пространство процесса делится на сегменты, а каждый сегмент, в свою очередь, делится на виртуальные страницы. Виртуальные страницы нумеруются не в пределах всего адресного пространства процесса, а в пределах сегмента. Виртуальный адрес в этом случае выражается тройкой (номер сегмента, номер страницы, смещение в странице).

Загрузка процесса выполняется ОС постранично, при этом часть страниц размещается в ОП, а часть – на диске. Для каждого процесса создается собственная таблица сегментов, а для каждого сегмента – своя таблица страниц. Адрес таблицы сегментов загружается в специальный регистр процессора, когда активируется соответствующий процесс.

Таблица страниц содержит дескрипторы страниц. Таблица сегментов состоит из дескрипторов сегментов, которые содержат описание расположения таблиц страниц в физической памяти.

На рисунке 61 показана схема преобразования виртуального адреса в физический для данного метода.

1 По номеру сегмента, заданному в виртуальном адресе, из таблицы сегментов извлекается физический адрес соответствующей таблицы страниц.

2 По номеру виртуальной страницы, заданному в виртуальном адресе, из таблицы страниц извлекается дескриптор, в котором указан номер физической страницы.

3 К номеру физической страницы пристыковывается младшая часть виртуального адреса – смещение.

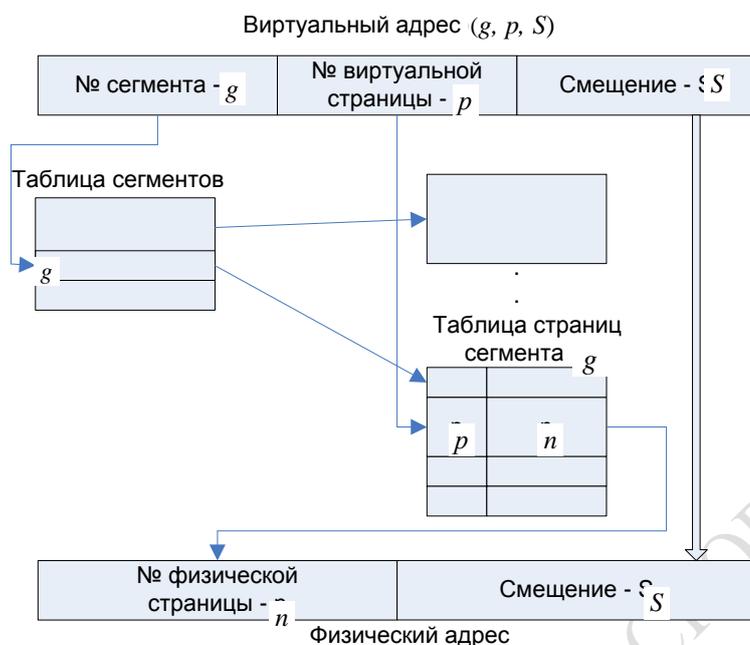


Рисунок 61 – Странично-сегментное распределение памяти

4.15 Борьба с фрагментацией памяти

При работе в мультипрограммном режиме может сложиться ситуация, когда между программами образуются незанятые участки памяти очень маленького размера. Такое состояние называется фрагментацией реальной памяти. Оно характерно для систем со статическим перемещением (рисунок 62).

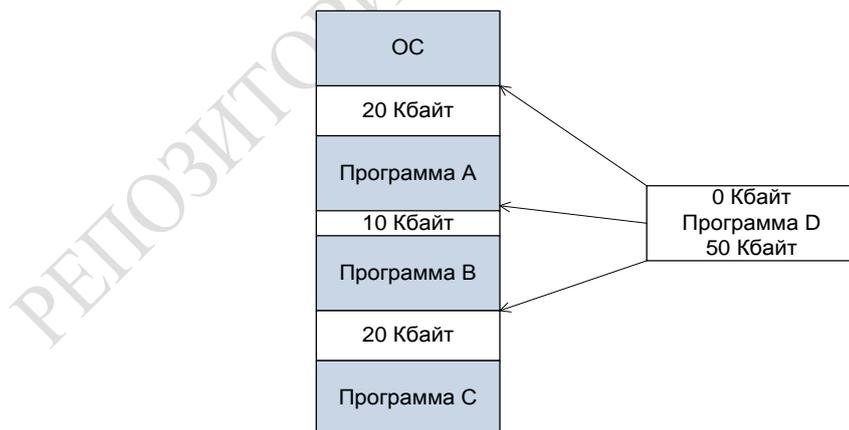


Рисунок 62 – Фрагментация памяти

В системах с динамическим перемещением программ перемещающий загрузчик размещает программу в свободной части памяти и допускает использование несмежных ее участков (рисунок 63).

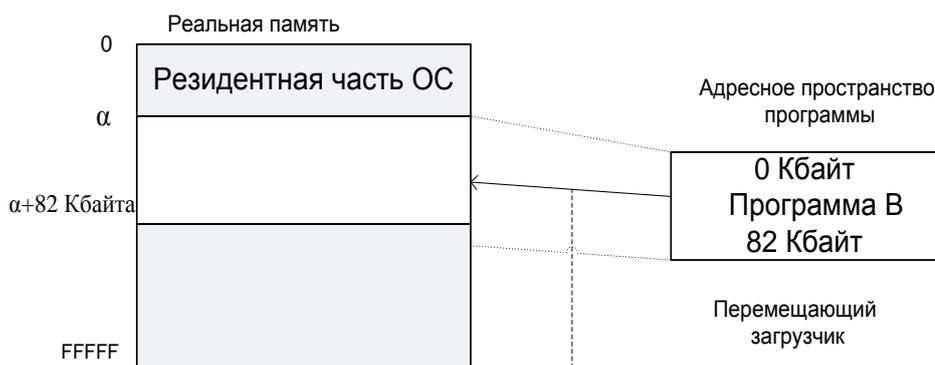


Рисунок 63 – Динамическое перемещение программ

В этом случае имеется больше возможностей для организации мультипрограммной работы, а следовательно, и для более эффективного использования временных ресурсов ЭВМ.

При больших размерах реализуемых программ возникают некоторые противоречия в организации мультипрограммного режима работы и трудности динамического распределения ресурсов.

В настоящее время разработано несколько способов решения этих противоречий. Например, для борьбы с фрагментацией основной памяти адресное пространство программы может быть разбито на отдельные сегменты, слабо связанные между собой. Тогда программа, которая имеет длину, превышающую размеры свободных участков памяти, может быть представлена в виде ряда сегментов, загружаемых в различные области ОП. Это позволяет использовать реальную память, теряемую из-за фрагментации.

Адреса в каждом сегменте начинаются с **0h**. При статическом перемещении программы в процессе загрузки ее в *основную память* адреса должны быть привязаны к конкретному месту в памяти. На это уходит много времени, отвлекаются вычислительные ресурсы. Более эффективной является динамическая трансляция адресов (ДТА), которая заключается в том, что сегменты загружаются в *основную память* без трансляции адресного пространства (то есть без изменения адресов в программе с учетом физического размещения в памяти команд и данных), а трансляция адресов каждой команды производится в процессе ее выполнения. Этот тип трансляции называется динамическим перемещением и осуществляется специальными аппаратными средствами ДТА (рисунок 64).

Для динамической трансляции адресов (то есть при определении абсолютных адресов по известным относительным, содержащим номер сегмента и смещение) операционная система строит

специальные таблицы, устанавливающие соответствие между сегментируемым адресным пространством программы и действительными адресами сегментов в реальной памяти (рисунок 65).

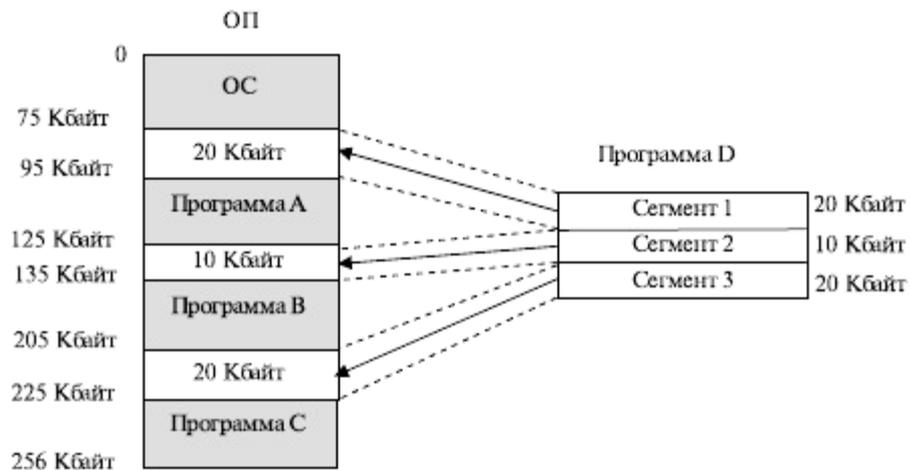


Рисунок 64 – Динамическое перемещение программ

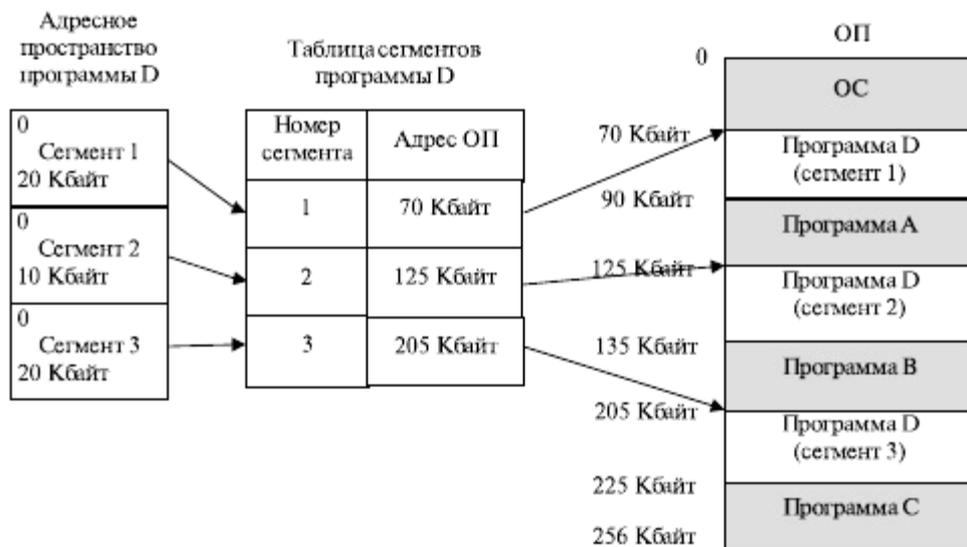


Рисунок 65 – Динамическая трансляция адресов

Каждая строка таблицы сегментов содержит адрес начала сегмента в реальной памяти. Для каждого сегмента имеется одна строка таблицы. Таблицу сегментов содержит каждая выполняемая программа.

В дополнение к таблице сегментов для динамической трансляции адреса применяется специальный управляющий регистр, называемый регистром начала таблицы сегментов (RHTC, или STOR

(segment table origin register)). В этот регистр занесен адрес таблицы сегментов выполняемой в данный момент программы. В МП Pentium в качестве РНТС используются 3 регистра: GDTR, LDTR и IDTR.

Использованием *сегментации* программ достигается уменьшение фрагментации основной памяти, но полностью фрагментация не устраняется – остаются фрагменты, длина которых меньше длины сегментов программы.

Если *сегменты разделить* на одну или несколько единиц, называемых страницами, которые имеют фиксированный размер, то, поскольку размер страницы достаточно мал по сравнению с обычным размером сегментов, неиспользуемые фрагменты ОП значительно сокращаются в объеме, будет происходить так называемая фрагментация внутри страниц. Следовательно, потери все-таки останутся, но они будут существенно меньше. Такая организация использования ресурсов называется сегментно-страничной.

Формирование сегментно-страничной структуры выполняется автоматически с помощью операционной системы.

Все преимущества динамического перемещения с применением *сегментации* и *страничной организации* достигаются благодаря аппаратуре и программному обеспечению, а не пользователям системы. Специальные программы во время загрузки разбивают адресное пространство программы на сегменты и страницы, строят таблицы сегментов и страниц. Средства ДТА автоматически транслируют адрес в процессе выполнения программы.

4.16 Управление программными ресурсами

Различают аппаратные и программные ресурсы ЭВМ [11]. К аппаратным ресурсам относятся микропроцессор (процессорное время), оперативная память и периферийные устройства; к программным ресурсам – доступные пользователю программные средства для управления вычислительными процессами и данными. Важнейшими программными ресурсами являются программы, входящие в систему программирования; средства программного управления периферийными устройствами и файлами; библиотеки системных и прикладных программ; средства, обеспечивающие контроль и взаимодействие вычислительных процессов (задач).

Программные ресурсы – это все программы, установленные в компьютере. Часто называют программным обеспечением (ПО).

ПО современных компьютеров включает множество разнообразных программ, которое можно условно разделить на три группы (рисунок 66):

- 1 Системное программное обеспечение (системные программы).
- 2 Прикладное программное обеспечение (прикладные программы).
- 3 Инструментальное обеспечение (инструментальные системы).



Рисунок 66 – Категории программного обеспечения

5 ВИРТУАЛИЗАЦИЯ ОПЕРАЦИОННЫХ СИСТЕМ

5.1 Концепция виртуальной машины

Виртуальная машина – программная или аппаратная среда, исполняющая некоторый код (например, байт-код, шитый код, р-код или машинный код реального процессора), или спецификация такой системы [12] (например: «виртуальная машина языка программирования Си»).

Виртуальная машина – это полностью изолированный программный контейнер, способный выполнять собственную операционную систему и приложения. Виртуальная машина работает абсолютно так же, как физический компьютер, и содержит собственные виртуальные (то есть программные) ЦП, ОЗУ, жесткий диск и сетевую интерфейсную карту (NIC).

Проще говоря, виртуальная машина – это программа, которую вы запускаете из своей операционной системы. Программа эмулирует реальную машину. На виртуальные машины, как и на реальные, можно ставить операционные системы. У неё есть BIOS, отведенное место на вашем жестком диске, сетевые адаптеры для соединения с реальной машиной, сетевыми ресурсами или другими виртуальными машинами.

Процесс или система, которые выполняются на виртуальной машине, называются гостем, платформа, поддерживающая виртуальную машину, – хостом. Программное обеспечение, реализующее процессную виртуальную машину, называют рабочей средой, а программное обеспечение виртуализации системной виртуальной машины – монитором виртуальной машины.

Конфигурация виртуальной машины отличается от конфигурации реальной ЭВМ. На одной реальной ЭВМ может функционировать несколько виртуальных машин. Программное обеспечение виртуальных машин, основой которого является МВМ, представляет собой систему виртуальных машин (СВМ).

Система виртуальных машин обеспечивает новый принцип параллелизма (мультипрограммирования) в работе реальной ЭВМ. Каждому пользователю предоставляется отдельная ВМ, в которой может функционировать:

- версия операционной системы;

- собственная системно-независимая программа;
- подсистема СВМ, например подсистема диалоговой обработки (ПДО);
- подсистема операционной системы, основанная на концепции VM.

Пользователи отдельных VM полностью независимы.

Виртуальные машины, такие как Virtualbox, используются для эмуляции виртуального оборудования и запуска нескольких операционных систем на компьютере. Чем лучше будет у вас CPU и чем больше будет оперативной памяти, тем быстрее будут выполняться виртуальные машины на вашем компьютере (рисунок 67).



Рисунок 67 – Система виртуальных машин

Система виртуальных машин решает проблемы «традиционных» операционных систем:

1 Введение концепции виртуальной машины позволяет уменьшить взаимное влияние различных режимов использования вычислительной системы. Упрощается управляющая программа.

2 Пользователь имеет иллюзию обладания отдельной ЭВМ. В действительности же он обладает отдельной виртуальной машиной. Управляющая программа в большей степени «прозрачна».

3 Виртуальная основная память позволяет снизить остроту проблемы «фрагментации» памяти.

4 Уровень реального мультипрограммирования повышается. В результате возрастает интенсивность использования ресурсов, в первую очередь ЦП и оперативной памяти.

5 Виртуализация расширяет объем ресурсов, в первую очередь основной памяти.

6 На одной реальной ЭВМ можно выполнить одновременно несколько операционных систем, их версий, подсистем, системно-независимых программ.

5.2 Концепция виртуального ресурса

Границы, определяющие область действия понятия «ресурс», достаточно условны. Поэтому будем полагать, что всякий потребляемый объект (независимо от формы его существования), обладающий некоторой практической ценностью для потребителя, является ресурсом.

Одним из важнейших свойств ресурса является «реальность существования». В этом смысле ресурсы разделяют на физические и виртуальные (мнимые). Под физическим понимают ресурс, который реально существует и при распределении его между пользователями обладает всеми присущими ему физическими характеристиками. Виртуальный ресурс схож многими своими характеристиками с некоторым физическим, но по многим свойствам и отличен. По сути – это некоторая модель физического ресурса. Виртуальный ресурс не существует в том виде, в котором он проявляет себя пользователю.

Свойство виртуализации ресурсов – одно из важнейших при построении систем управления ресурсами. По значимости – это одна из важнейших концепций при построении современных ОС.

Построение каждого виртуального ресурса проводится на базе некоторого физического [27]. Имея всего один физический ресурс, можно построить на его основе несколько виртуальных. Это дает возможность существенно экономичнее использовать соответствующий физический ресурс, а также увеличивать гибкость политики распределения ресурсов, исключая, в большинстве случаев, конфликтные ситуации.

Так, например, пользователю может быть предоставлена виртуальная оперативная память, размер которой превосходит всю имеющуюся в системе реальную оперативную память. Пользователь пишет программы так, как будто в его распоряжении имеется однородная оперативная память большого объема, но в действительности все данные, используемые программой, хранятся на одном или

нескольких разнородных запоминающих устройствах, обычно на дисках, и при необходимости частями отображаются в реальную память. Пример виртуализации дискового пространства приведён на рисунке 68.



Рисунок 68 – Процесс преобразования физических дисков в виртуальный пул, а затем разбиения виртуального пула на виртуальные диски

5.3 Классификация систем управления образами виртуальных машин

В классификации виртуальных машин (ВМ) их можно разделить на процессные и системные [12].

Процессная виртуальная машина – это виртуальная платформа для выполнения отдельного процесса. Она предназначена для поддержки процесса, создаётся при его активации и «умирает» после его окончания.

Системная виртуальная машина – полнофункциональная, постоянно действующая системная среда, служащая для поддержки операционной системы вместе с большим количеством её пользовательских процессов; она обеспечивает «гостевой» операционной системе доступ к виртуальным аппаратным средствам, в том числе к процессору и памяти, устройствам ввода/вывода, а иногда – и к графическому интерфейсу.

Дальше систематизация происходит в зависимости от того, какие системы имеет виртуальная машина и её базовая платформа, а точнее совпадают ли они (рисунок 69).



Рисунок 69 – Классификация виртуальных машин

Среди процессных виртуальных баз ISA можно выделить многозадачные системы, поддерживаемые большинством современных компьютеров.

В раздел «Процессные VM» – разные ISA входят динамические трансляторы. У них интерфейс чаще всего определяется на уровне ABI.

В категории системных машин с одинаковыми ISA входят классические и вложенные системные виртуальные машины, служащие для репликации изолированной системной среды. Различие же между ними – в реализации монитора виртуальных машин, но не в функциональных возможностях.

В интегральных системных виртуальных машинах производительность играет не первую роль, на первом месте точность воспроизведения функциональных возможностей.

Виртуальных машин существует великое множество. И каждая из них имеет свои плюсы и свои минусы.

6 КОМПЬЮТЕРНЫЕ СЕТИ

6.1 Способы и цели объединения компьютерных сетей и систем

Компьютерная сеть – это множество компьютеров, соединенных линиями связи и работающих под управлением специального программного обеспечения [14].

Главной целью объединения компьютеров в сеть является предоставление пользователям возможности доступа к различным информационным ресурсам (например, документам, программам, базам данных и т. д.), распределенным по этим компьютерам, и их совместного использования.

Существует достаточно много способов объединения компьютеров. Наибольшее распространение получили сетевые топологии «шина», «звезда» и «кольцо», поэтому именно их мы и рассмотрим [18].

Топология «шина» (рисунок 70) представляет собой общий кабель (называется шина или магистраль), к которому подсоединены все рабочие станции. На концах кабеля находятся терминаторы, для предотвращения отражения сигнала. Отправляемое какой-либо рабочей станцией сообщение распространяется на все компьютеры сети. Каждая машина проверяет, кому адресовано сообщение, если сообщение адресовано ей, то обрабатывает его.

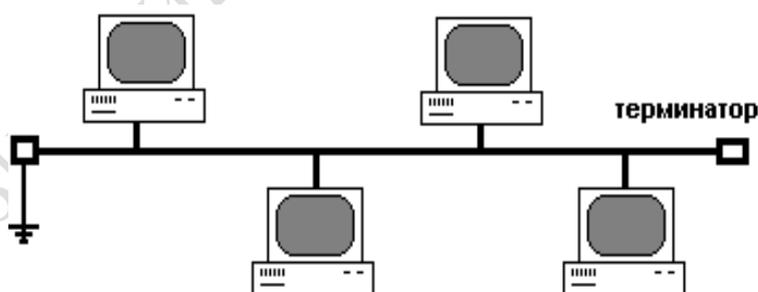


Рисунок 70 – Топология «шина»

Достоинства: небольшое время установки сети; дешевизна (требуется кабель меньшей длины и меньше сетевых устройств); простота настройки; выход из строя одной рабочей станции не отражается на работе всей сети.

Недостатки: неполадки в сети, такие как обрыв кабеля или выход из строя терминатора, полностью блокируют работу всей сети;

затрудненность выявления неисправностей; с добавлением новых рабочих станций падает общая производительность сети.

Топология «звезда» (рисунок 71) – все компьютеры сети присоединены к центральному узлу (обычно коммутатор), образуя физический сегмент сети. Весь обмен информацией идет исключительно через центральный компьютер, на который возлагается очень большая нагрузка, поэтому ничем другим, кроме сети, он заниматься не может.

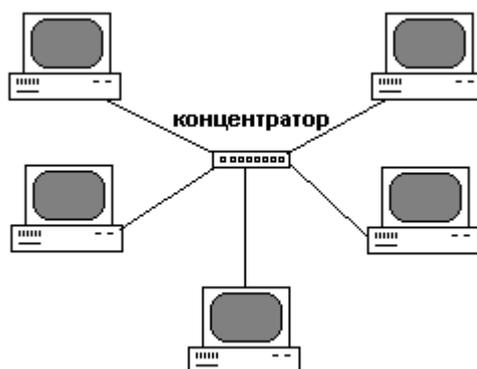


Рисунок 71 – Топология «звезда»

Достоинства: выход из строя одной рабочей станции не отражается на работе всей сети в целом; хорошая масштабируемость сети; лёгкий поиск неисправностей и обрывов в сети; высокая производительность сети (при условии правильного проектирования); гибкие возможности администрирования.

Недостатки: выход из строя центрального концентратора обернётся неработоспособностью сети (или сегмента сети) в целом; для прокладки сети зачастую требуется больше кабеля, чем для большинства других топологий; конечное число рабочих станций в сети (или сегменте сети) ограничено количеством портов в центральном концентраторе.

Топология «кольцо» – это топология, в которой каждый компьютер соединён линиями связи только с двумя другими: от одного он только получает информацию, а другому только передаёт (рисунок 72). На каждой линии связи, как и в случае звезды, работает только один передатчик и один приёмник. Это позволяет отказаться от применения внешних терминаторов.

Достоинства: простота установки; практически полное отсутствие дополнительного оборудования; возможность устойчивой работы без существенного падения скорости передачи данных при

интенсивной загрузке сети, поскольку использование маркера исключает возможность возникновения коллизий.

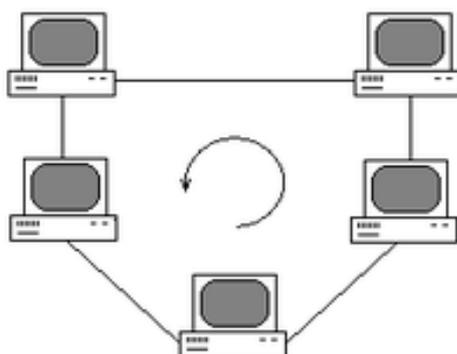


Рисунок 72 – Топология «кольцо»

Недостатки: выход из строя одной рабочей станции и другие неполадки (обрыв кабеля) отражаются на работоспособности всей сети; сложность конфигурирования и настройки; сложность поиска неисправностей; необходимость иметь две сетевые платы, на каждой рабочей станции; добавление/удаление станции требует временной остановки работы сети.

6.2 Понятие информационного потока в компьютерной сети

Информационным потоком называют последовательность данных, объединенных набором общих признаков, который выделяет эти данные из общего сетевого трафика. Данные могут быть представлены в виде последовательности байтов или объединены в более крупные единицы данных – пакеты, кадры, ячейки.

Например, все данные, поступающие от одного компьютера, можно определить как единый поток, а можно представить как совокупность нескольких подпотоков, каждый из которых в качестве дополнительного признака имеет адрес назначения. Каждый из этих подпотоков, в свою очередь, можно разделить на еще более мелкие подпотоки данных, например, относящихся к разным сетевым приложениям – электронной почте, копированию файлов, обращению к Web-серверу.

В задаче коммутации, суть которой – передача данных из одного конечного узла в другой, при определении потоков в роли обязательных

признаков потока, очевидно, должны выступать адрес отправителя и адрес назначения данных. Тогда каждой паре конечных узлов будет соответствовать один поток и один маршрут.

Однако не всегда достаточно определить поток только парой адресов. Если на одной и той же паре конечных узлов выполняется несколько взаимодействующих по сети приложений, которые предъявляют к ней свои особые требования, поток данных между двумя конечными узлами должен быть разделен на несколько подпотоков, так чтобы для каждого из них можно было проложить свой маршрут. В таком случае выбор пути должен осуществляться с учетом характера передаваемых данных. Например, для файлового сервера важно, чтобы передаваемые им большие объемы данных направлялись по каналам с высокой пропускной способностью, а для программной системы управления, которая посылает в сеть короткие сообщения, требующие обязательной и немедленной отработки, при выборе маршрута важнее надежность линии связи и минимальный уровень задержек. В таком примере набор признаков потока должен быть расширен за счет информации, идентифицирующей приложение.

Кроме того, даже для данных, предъявляющих к сети одинаковые требования, может прокладываться несколько маршрутов, чтобы за счет распараллеливания добиться одновременного использования различных каналов и тем самым ускорить передачу данных. В данном случае необходимо «пометить» данные, которые будут направляться по каждому из этих маршрутов.

Признаки потока могут иметь глобальное или локальное значение. В первом случае они однозначно определяют поток в пределах всей сети, а во втором – в пределах одного транзитного узла. Пара уникальных адресов конечных узлов для идентификации потока – это пример глобального признака. Примером признака, локально определяющего поток в пределах устройства, может служить номер (идентификатор) интерфейса устройства, с которого поступили данные.

Например, узел 1 (рисунок 73) может быть сконфигурирован так, что он передает все данные, поступившие с интерфейса А, на интерфейс С, а данные, поступившие с интерфейса D, на интерфейс В. Такое правило позволяет разделить два потока данных – поступающий из узла 2 и поступающий из узла 7 – и направлять их для транзитной передачи через разные узлы сети, в данном случае данные из узла 2 через узел 8, а данные из узла 7 – через узел 5.

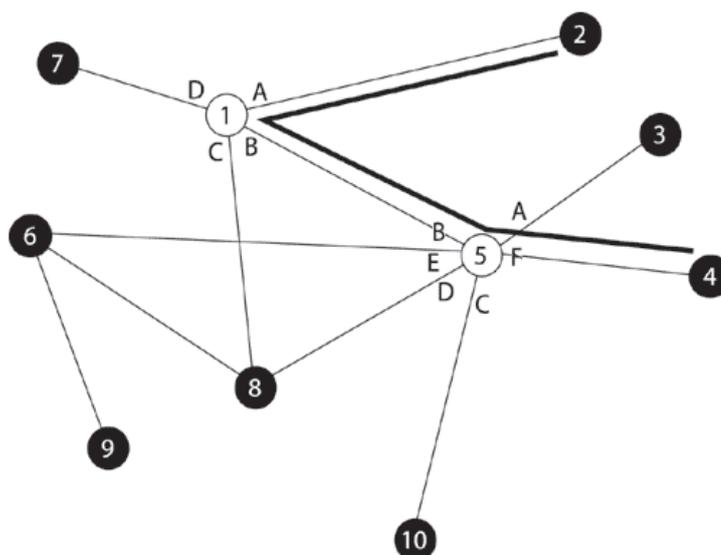


Рисунок 73 – Коммутация абонентов через сеть транзитных узлов

6.3 Выбор эффективной версии межсетевого экрана для реализации информационной защиты

Межсетевой экран или сетевой экран – комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами [19].

Основной задачей сетевого экрана является защита компьютерных сетей или отдельных узлов от несанкционированного доступа. Также сетевые экраны часто называют фильтрами, так как их основная задача – не пропускать (фильтровать) пакеты, не подходящие под критерии, определённые в конфигурации.

Поддерживаемый уровень сетевой модели OSI является основной характеристикой при классификации межсетевых экранов. Различают следующие типы межсетевых экранов:

- 1 Управляемые коммутаторы (канальный уровень).
- 2 Сетевые фильтры сетевого уровня (stateless). Фильтрация статическая осуществляется путём анализа IP-адреса источника и приёмника, протокола, портов отправителя и получателя.
- 3 Шлюзы сеансового уровня (circuit-level proxy). В сетевой модели TCP/IP нет уровня, однозначно соответствующего сеансовому уровню OSI, поэтому к шлюзам сеансового уровня относят фильтры, которые невозможно отождествить ни с сетевым, ни с транспортным, ни с прикладным уровнем:

– шлюзы, транслирующие адреса (NAT, PAT) или сетевые протоколы (транслирующий мост);

– фильтры контроля состояния канала. К фильтрам контроля состояния канала связи нередко относят сетевые фильтры сетевого уровня с расширенными возможностями (stateful), которые дополнительно анализируют заголовки пакетов и умеют фильтровать фрагментированные пакеты);

– шлюзы сеансового уровня. Наиболее известным и популярным шлюзом сеансового уровня является посредник SOCKS.

4 Шлюзы прикладного уровня (application-level proxy), часто называемые прокси-серверами. Они делятся на прозрачные (transparent) и непрозрачные.

5 Брандмауэр SPI (Stateful Packet Inspection, SPI), или иначе брандмауэры с динамической фильтрацией пакетов (Dynamic Packet Filtering), являются по сути шлюзами сеансового уровня с расширенными возможностями. Инспекторы состояния оперируют на сеансовом уровне, но «понимают» протоколы прикладного и сетевого уровней. В отличие от шлюза прикладного уровня, открывающего два виртуальных канала TCP (один – для клиента, другой – для сервера) для каждого соединения, инспектор состояния не препятствует организации прямого соединения между клиентом и сервером.

Существует также понятие «межсетевой экран экспертного уровня». Сетевой экран данного типа базируются на посредниках прикладного уровня или инспекторах состояния, но обязательно комплектуются шлюзами сеансового уровня и сетевыми фильтрами, иногда понимая и сетевой уровень. Зачастую имеют систему протоколирования событий и оповещения администраторов, средства поддержки удаленных пользователей (например, авторизация), средства построения виртуальных частных сетей и т. д. К нему относятся почти все имеющиеся на рынке брандмауэры. На рисунке 74 представлена иллюстрация расположения межсетевого экрана в сети.

Теория поиска оптимального решения в практике выбора МСЭ.

Чтобы выбрать наилучший по функциональным характеристикам МСЭ, нужно последовательно решить следующие задачи.

1 Определить критерии выбора межсетевого экрана, то есть основные функциональные требования к нему.

2 Присвоить каждому критерию весовой коэффициент (сумма всех весовых коэффициентов должна была быть равна единице), указывающий на степень его важности.

3 Проанализировать наличие и качество реализации необходимых функций в подмножестве межсетевых экранов (отобранном, например, в определенном диапазоне цен). На этом этапе каждому МСЭ выставляется балл по десятибалльной шкале.

4 Перемножить весовые коэффициенты из п. 2 и полученный для каждого критерия (см. п. 3) балл. Результат даст итоговую оценку для каждого МСЭ, а максимальное значение укажет на наилучший продукт из подмножества.

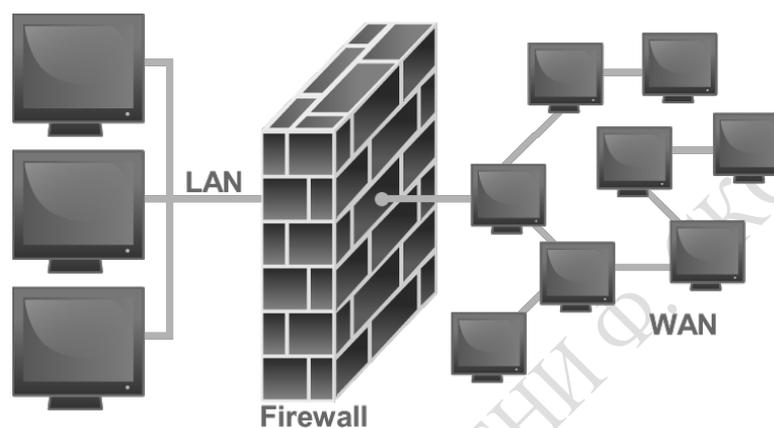


Рисунок 74 – Расположение межсетевого экрана в сети

6.4 Обмен данными мобильных устройств с ПК

На сегодняшний день существует пять способов соединения мобильных устройств и компьютеров: инфракрасный порт, data-кабель, технологии Bluetooth, Wi-Fi и GPRS [10].

Инфракрасный порт. Многие устройства могут соединяться по оптическим линиям связи, таким как Infrared. Этот метод связи для передачи голоса и данных между устройствами использует инфракрасный диапазон волн. При этом типе связи сигнал должен иметь свободную, прямую траекторию распространения от одного устройства к другому, а расстояние между устройствами не должно превышать одного метра. Технология связи Infrared предназначена для передачи данных или синхронизации файлов только для соединенный point-to-point.

Data-кабель. Название data-кабель (от англ. data-cable – «кабель для передачи данных»). Data-кабель – это устройство, при помощи которого можно соединить мобильный телефон с компьютером. Кабель выполняет такие функции, как загрузка в телефон файлов,

то есть мелодий, картинок, игр, тем рабочего стола, звуковых эффектов и др. Кроме того, data-кабель (при условии, что телефон это поддерживает) позволяет использовать телефон как модем и выходить через него в Интернет при помощи услуги GPRS. Соединение с помощью Data-кабеля – это в первую очередь очень высокая скорость приема/передачи данных (до 12 Мбит/с). Для соединения мобильного устройства и компьютера не понадобится ничего, кроме диска с драйверами и самого Data-кабеля.

Bluetooth. Подключение по Bluetooth имеет как преимущества, так и недостатки. Основное преимущество – возможность соединения устройств с помощью радиочастот вне пределов прямой видимости (в среднем на расстоянии до 10 метров). Недостатком является низкая скорость передачи данных, несмотря на то, что в спецификации Bluetooth 3.0 + HS теоретически поддерживается скорость передачи данных до 24 Мбит/с. В мобильных устройствах нет смысла использовать данную технологию из-за высокого энергопотребления. Стоит отметить, что до сих пор является нерешенной проблема подключения телефонов отдельных производителей, например, Nokia, к ПК с операционными системами UNIX или Mac OS.

Wi-Fi. Под аббревиатурой Wi-Fi (от англ. Wireless Fidelity – «высокая точность беспроводной передачи данных») в настоящее время развивается целое семейство стандартов передачи цифровых потоков данных по радиоканалам. Соединение с помощью Wi-Fi – высокоскоростной способ передачи данных на радиочастотах. Действует в радиусе до 300 метров на открытых пространствах и до 30 метров в помещениях. Оснащаются им новые КПК, смартфоны и мобильные последнего поколения. Скорость передачи данных – до 54 Мбит/с.

GPRS (General Packet Radio Service). С помощью GPRS пользователи могут работать со своей электронной почтой, с обычными Web-серверами. Основное достоинство GPRS-сетей состоит в том, что пользователь оплачивает только объем передаваемой/получаемой информации, а не время нахождения в сети. В GPRS максимально возможная скорость передачи данных составляет 171,2 Кбит/с.

У каждой из рассмотренных технологий, с помощью которых можно соединить мобильное устройство с компьютером (рисунок 75), есть свои преимущества и недостатки. Оптимальным из рассмотренных является соединение при помощи Wi-Fi – самая высокая скорость передачи данных, мобильные устройства с поддержкой данной технологии находятся в среднем ценовом сегменте рынка.



Рисунок 75 – Связь компьютера и телефона

6.5 Защита межсетевого трафика, межсетевые экраны

Межсетевой трафик – объем информации, который пользователь Интернета получает из сети на свой компьютер и отправляет в сеть [23].

Межсетевой экран или сетевой экран – комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами.

Межсетевой экран (от немец. брандмауэр, от англ. фаервол) – это защитный барьер между вашим компьютером и сетью к которой он подключен. Когда вы заходите в интернет, то ваш компьютер становится видимым для внешнего мира. Вы видимы через нечто, что называется порт.

Порт – это идентифицируемый определенным номером системный ресурс, выделяемый приложению, которое выполняется на некотором сетевом хосте (компьютер или другое сетевое устройство), для связи с приложениями, которые выполняются на других сетевых хостах.

Суть брандмауэра в том, чтобы *закрыть порты*, которые вы не используете (рисунок 76). В противном случае через них злоумышленник или вредоносная программа (вирус, троян) могут проникнуть на ваш ПК.

Что может случиться, если вы не будете использовать брандмауэр?

Если после установки операционной системы вы не активируете брандмауэр или если он не будет активирован по умолчанию, то ваша система может быть атакована спустя несколько минут после выхода в интернет. Не стоит также считать, что наличие только фаервола поможет уберечься от всех бед. Не стоит забывать об установке антивируса с последующим регулярным обновлением баз, а также о регулярном скачивании и установке обновлений безопасности для вашей Windows.

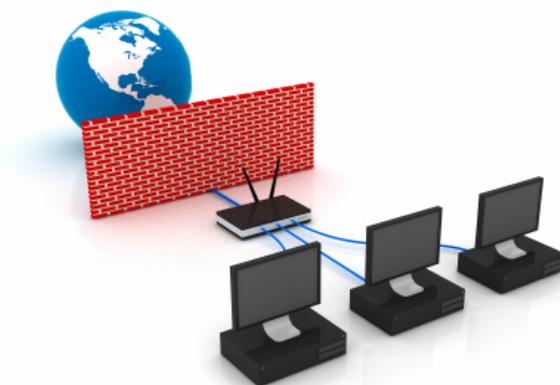


Рисунок 76 – Защита ПК межсетевым экраном

Как брандмауэр помогает защитить ваш ПК?

Большинство межсетевых экранов, включая встроенные в Windows, будут оповещать вас о подозрительном входящем трафике. Но хороший брандмауэр должен оповещать и о подозрительном исходящем трафике. Например, встроенный брандмауэр Windows XP не умеет это делать, поэтому его лучше заменить на программу от стороннего производителя. Наличие подозрительного исходящего трафика поможет вам понять, что ваш компьютер уже заражен троянским или шпионским ПО.

Виды межсетевых экранов.

Брандмауэры можно поделить на две простые категории: аппаратные и программные.

Аппаратным фаерволом может быть маршрутизатор, который находится между вашим ПК и сетью Интернет. В таком случае к нему можно подключить несколько компьютеров, и все они будут защищены брандмауэром, который является частью маршрутизатора.

Программный межсетевой экран – это специализированное ПО, которое пользователь устанавливает себе на компьютер.

Даже если у вас уже есть маршрутизатор со встроенным межсетевым экраном, вы можете также установить программный фаервол

на каждый компьютер в отдельности. Тогда злоумышленнику будет значительно тяжелее проникнуть в вашу систему.

6.6 Применение фильтров доступа к внешним сетевым ресурсам

Access Control List или ACL – список контроля доступа, который определяет, кто или что может получать доступ к конкретному объекту и какие именно операции разрешено или запрещено этому субъекту проводить над объектом [1].

В типичных ACL каждая запись определяет субъект воздействия и операцию: например, запись (Vasya, delete) в ACL для файла XYZ даёт возможность пользователю Vasya удалить файл XYZ.

В системе с моделью безопасности, основанной на ACL, когда субъект запрашивает выполнение операции над объектом, система сначала проверяет список разрешённых для этого субъекта операций и только после этого даёт (или не даёт) доступ к запрошенному действию.

При централизованном хранении списков контроля доступа можно говорить о матрице доступа, в которой по осям размещены объекты и субъекты, а в ячейках – соответствующие права. Однако в большом количестве систем списки контроля доступа к объектам хранятся отдельно для каждого объекта, зачастую непосредственно с самим объектом.

Традиционные ACL системы назначают права индивидуальным пользователям, и со временем и ростом числа пользователей в системе списки доступа могут стать громоздкими. Одним из вариантов решения этой проблемы является назначение прав группам пользователей, а не персонально. Другой вариант – «Управление доступом на основе ролей», где функциональные подмножества прав к ряду объектов объединяются в «роли», и эти роли назначаются пользователям. Однако в первом варианте группы пользователей также часто называются ролями.

В файловых системах для реализации ACL используется идентификатор пользователя процесса (UID в терминах POSIX).

Список доступа представляет собой структуру данных (обычно таблицу), содержащую записи, определяющие права индивидуального пользователя или группы на специальные системные объекты, такие как программы, процессы или файлы.

Эти записи также известны как ACE (англ. Access Control Entries) в операционных системах Microsoft Windows и OpenVMS. В операционной системе Linux и Mac OS X большинство файловых систем имеют расширенные атрибуты, выполняющие роль ACL. Концепции ACL в разных операционных системах различаются, несмотря на «стандарт» POSIX.

В сетях ACL представляют список правил, определяющих порты служб или имена доменов, доступных на узле или другом устройстве третьего уровня OSI, каждый со списком узлов и/или сетей, которым разрешен доступ к сервису.

6.7 Использование Проxy-серверов

Proxy-сервер (от англ. представитель, уполномоченный) – служба (комплекс программ) компьютерных сетей, позволяющая клиентам выполнять косвенные запросы к другим сетевым службам (рисунок 77). Сначала клиент подключается к Proxy-серверу и запрашивает какой-либо ресурс (например, e-mail), расположенный на другом сервере. Затем Proxy-сервер либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного кэша (в случаях, если Proxy имеет свой кэш). В некоторых случаях запрос клиента или ответ сервера может быть изменен Proxy-сервером в определенных целях. Также Proxy-сервер позволяет защищать компьютер клиента от некоторых сетевых атак и помогает сохранять анонимность клиента.

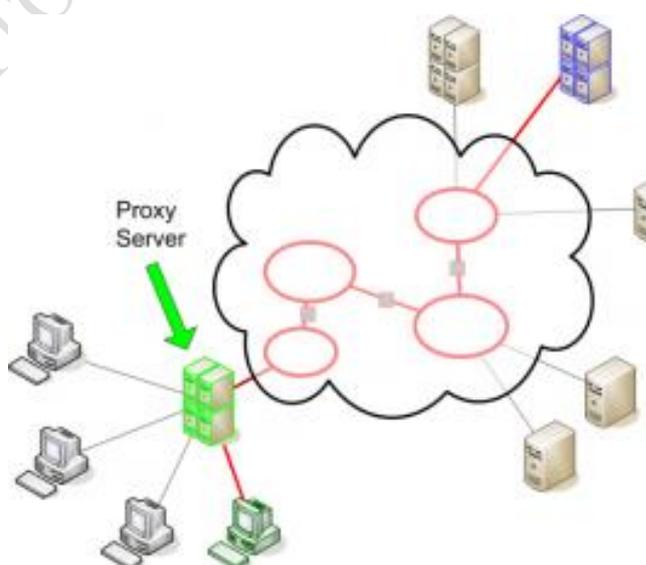


Рисунок 77 – Proxy-сервер

Чаще всего Прoxy-серверы применяются для следующих целей:

- для обеспечения доступа с компьютеров локальной сети Интернет;

- для кэширования данных: если часто происходят обращения к одним и тем же внешним ресурсам, то можно держать их копию на Прoxy-сервере и выдавать по запросу, снижая тем самым нагрузку на канал во внешнюю сеть и ускоряя получение клиентом запрошенной информации;

- сжатие данных: Прoxy-сервер загружает информацию из Интернета и передает её конечному пользователю в сжатом виде. Такие Прoxy-серверы используются в основном с целью экономии внешнего сетевого трафика клиента или внутреннего – компании, в которой установлен Прoxy-сервер.

- защита локальной сети от внешнего доступа: например, можно настроить Прoxy-сервер так, что локальные компьютеры будут обращаться к внешним ресурсам только через него, а внешние компьютеры не смогут обращаться к локальным вообще (они видят только Прoxy-сервер);

- ограничение доступа из локальной сети к внешней: например, можно запретить доступ к определенным web-сайтам, ограничить использование Интернета каким-то локальным пользователям, устанавливать квоты на трафик или полосу пропускания, фильтровать рекламу и вирусы;

- Прoxy-сервер популярен среди пользователей стран, где доступ к некоторым ресурсам ограничен законодательством и фильтруется;

- анонимизация доступа к различным ресурсам. Прoxy-сервер может скрывать сведения об источнике запроса или пользователе. В таком случае целевой сервер видит лишь информацию о Прoxy-сервере, например, IP-адрес, но не имеет возможности определить истинный источник запроса. Существует также искажающие Прoxy-серверы, которые передают целевому серверу ложную информацию об истинном пользователе.

Прoxy-сервер, к которому может получить доступ любой пользователь сети Интернет, называется открытым.

7 ФАЙЛОВЫЕ СИСТЕМЫ

7.1 Файловая система и примеры распространенных файловых систем

Файловая система – это часть ОС, назначение которой состоит в том, чтобы обеспечить пользователю удобный интерфейс при работе с данными, хранящимися на диске, и совместное использование файлов несколькими пользователями и процессами [24].

В широком смысле понятие «файловая система» включает:

- совокупность всех файлов на диске;
- наборы структур данных, используемых для управления файлами, такие, например, как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске;
- комплекс системных программных средств, реализующих управление файлами, в частности: создание, уничтожение, чтение, запись, именованное, поиск и другие операции над файлами.

Функционирование любой файловой системы можно представить многоуровневой моделью (рисунок 78), в которой каждый уровень предоставляет некоторый интерфейс вышележащему уровню, а сам, в свою очередь, для выполнения своей работы использует интерфейс нижележащего уровня.

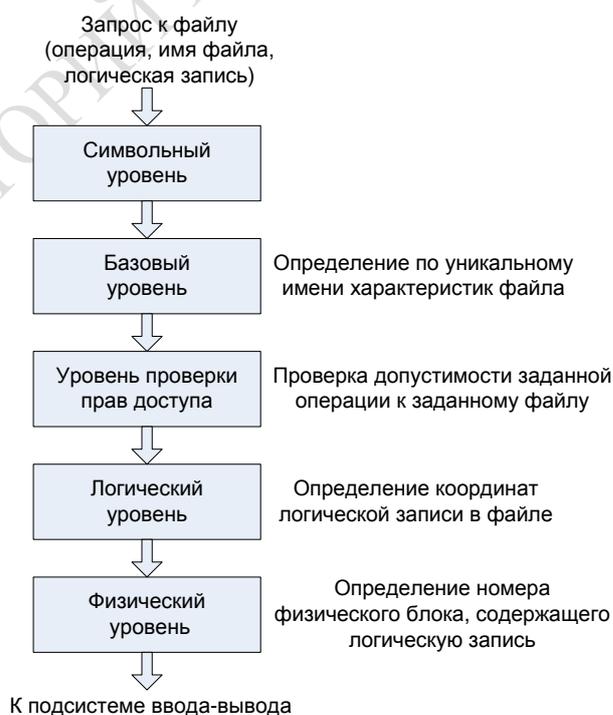


Рисунок 78 – Общая модель файловой системы

Задачи, решаемые файловой системой, зависят от способа организации вычислительного процесса в целом. Самый простой тип – это файловая система в однопользовательских и однопрограммных ОС, к числу которых относится, например, MS-DOS. Основные функции в такой файловой системе нацелены на решение следующих задач:

- именование файлов;
- программный интерфейс для приложений;
- отображение логической модели файловой системы на физическую организацию хранилища данных;
- устойчивость файловой системы к сбоям питания, ошибкам аппаратных и программных средств.

Задачи файловой системы усложняются в операционных однопользовательских мультипрограммных ОС, которые, хотя и предназначены для работы одного пользователя, но дают ему возможность запускать одновременно несколько процессов. Одной из первых ОС этого типа стала OS/2. К перечисленным выше задачам добавляется новая задача совместного доступа к файлу из нескольких процессов. Файл в этом случае является разделяемым ресурсом, а значит, файловая система должна решать весь комплекс проблем, связанных с такими ресурсами. В частности, в файловой системе должны быть предусмотрены средства блокировки файла и его частей, предотвращения гонок, исключение тупиков, согласование копий и т. п.

В многопользовательских системах появляется еще одна задача: защита файлов одного пользователя от несанкционированного доступа другого пользователя.

Еще более сложными становятся функции файловой системы, которая работает в составе сетевой ОС.

Файловая система FAT (File Allocation Table) – таблица размещения файлов. Этот термин относится к линейной табличной структуре со сведениями о файлах, их атрибутах и другими данными, определяющими местонахождение файлов в среде FAT. Элемент FAT определяет фактическую область диска, в которой хранится начало физического файла. Есть следующие разновидности файловой системы FAT: VFAT (виртуальная FAT) и FAT32.

Файловая система HPFS (High Performance File System) – высокопроизводительная файловая система. Файловая система HPFS стремится, прежде всего, к тому, чтобы расположить файл в смежных кластерах, или, если такой возможности нет, расширить его на

диске таким образом, чтобы фрагменты файла физически были как можно ближе друг к другу.

Файловая система NTFS (New Technology File System) содержит ряд значительных усовершенствований и изменений, существенно отличающих ее от других файловых систем. При проектировании системы NTFS особое внимание было уделено следующим характеристикам: надежность, расширенная функциональность, поддержка POSIX, гибкость.

7.2 Имена файлов в различных файловых системах

Наиболее важной характеристикой любого механизма абстракции является именование управляемых объектов. Правила именования файлов меняются от одной ОС к другой, но, обычно, все современные ОС поддерживают использование в качестве имен файлов 8-символьные текстовые строки. Часто в именах разрешается использование цифр и специальных символов. В некоторых файловых системах различаются прописные и строчные символы, тогда как в других, например, MS-DOS, – нет.

Во многих ОС имя файла состоит из двух частей, разделенных точкой. Часть имени после точки называется расширением файла и обычно означает его тип. Так, в MS-DOS имя файла может содержать от 1 до 8 символов, а расширение от 0 до 3.

В некоторых ОС, например, Windows, расширение указывает на программу, создавшую файл. Другие ОС, например, UNIX, не принуждают пользователя строго придерживаться расширений. Некоторые типичные расширения файлов приведены ниже (рисунок 79).

Расширение	Значение
file.bak	Резервная копия файла
file.cpp	Исходный текст программы на C++
file.gif	Изображение формата GIF
file.hlp	Файл справки
file.html	Документ в формате HTML
file.jpg	Неподвижное изображение стандарта JPEG
file.mp3	Музыка в формате MPEG-1 уровень 3
file.mpg	Фильм в формате MPEG
file.obj	Объектный файл

Рисунок 79 – Некоторые расширения файлов

В иерархически организованных файловых системах обычно используются три типа имен файлов: простые, составные и относительные.

Простое символьное имя идентифицирует файл в пределах одного каталога. Несколько файлов могут иметь одно и то же простое имя, если они принадлежат разным каталогам. В системе FAT длина имени ограничивается схемой 8.3; в системе s5, поддерживаемой многими версиями ОС Unix, – не более 14 символов; в файловых системах NTFS и FAT32, входящих в состав ОС Windows NT, – до 255 символов.

Составное символьное имя представляет собой цепочку, содержащую имя диска и имена всех каталогов, через которые проходит путь от корневого каталога до данного файла.

Относительное имя файла определяется через текущий каталог, то есть каталог, в котором в данный момент времени работает пользователь. Таким образом, относительных имен у файла может быть достаточно много, и все они являются частью полного имени.

7.3 Атрибуты файлов и права доступа к файлу

Понятие «файл» включает не только хранимые им данные и имя, но и атрибуты [28]. Атрибуты файла – это информация, описывающая свойства файла. Примеры возможных атрибутов файла:

- тип файла (обычный файл, каталог, специальный файл и т. п.);
- владелец файла;
- создатель файла;
- пароль для доступа к файлу;
- информация о разрешенных операциях доступа к файлу;
- времена создания последнего доступа и последнего изменения;
- текущий размер файла;
- максимальный размер файла;
- признак «только для чтения»;
- признак «скрытый файл»;
- признак «системный файл»;
- признак «архивный файл»;
- признак «двоичный/символьный»;
- признак «временный» (удалить после завершения процесса);
- признак блокировки;

- длина записи в файле;
- указатель на ключевое поле в записи;
- длина ключа.

Набор атрибутов файла определяется спецификой файловой системы: в файловых системах разного типа для характеристики файлов могут использоваться разные наборы атрибутов.

Пользователь может получать доступ к атрибутам, используя средства, предоставленные для этих целей файловой системой. Определить права доступа к файлу – значит определить для каждого пользователя набор операций, которые он может применить к данному файлу. Обычно разрешается читать значения любых атрибутов, а изменять – только некоторые.

Значения атрибутов файлов могут непосредственно содержаться в каталогах, как это сделано в файловой системе MS-DOS. На рисунке 80 представлена структура записи в каталоге, содержащая простое символьное имя и атрибуты файла. Здесь буквами обозначены признаки файла: R – только для чтения, A – архивный, H – скрытый, S – системный.

8		3				1	4				
Имя файла		Расширение				R	A	H	S	Резервное	
Резервное		В р е м я	Дата				№ резервного кластера		Размер		

Рисунок 80 – Структура записи каталога MS-DOS (32 байта)

Другим вариантом является размещение атрибутов в специальных таблицах, когда в каталогах содержатся только ссылки на эти таблицы. Такой подход реализован, например, в файловой системе ufs ОС UNIX. Запись о каждом файле содержит короткое символьное имя файла и указатель на индексный дескриптор файла, так называется в ufs таблица, в которой сосредоточены значения атрибутов файла (рисунок 81).

2		14	
№ индексного дескриптора		Имя файла	

Рисунок 81 – Структура записи каталога ОС UNIX

В том и другом вариантах каталоги обеспечивают связь между именами файлов и собственно файлами.

7.4 Файловые системы ОС семейства Unix

Все файлы, с которыми могут манипулировать пользователи, располагаются в файловой системе, представляющей собой дерево, промежуточные вершины которого соответствуют каталогам, и листья – файлам и пустым каталогам. Примерная структура файловой системы ОС UNIX показана на рисунке 82. Реально на каждом логическом диске располагается отдельная иерархия каталогов и файлов. Для получения общего дерева в динамике используется «монтирование» отдельных иерархий к фиксированной корневой файловой системе.

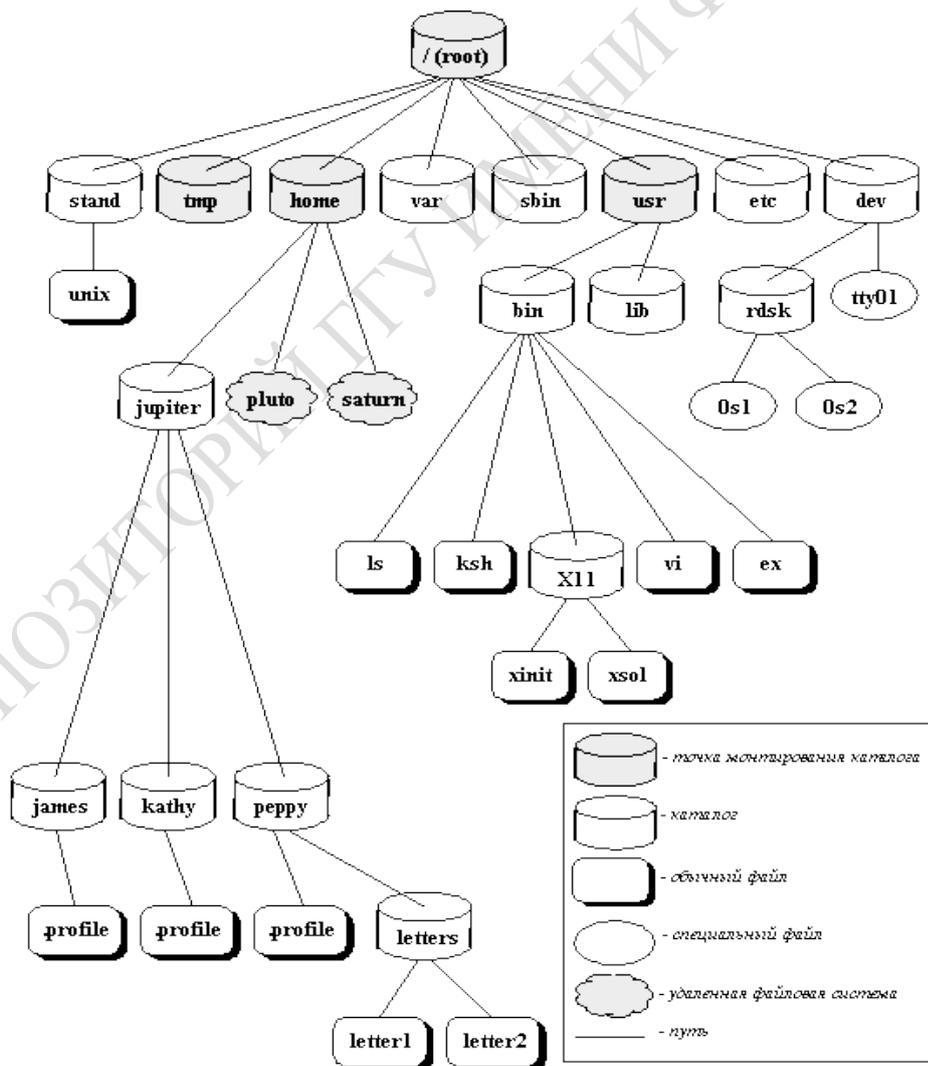


Рисунок 82 – Структура файловой системы ОС Unix

Каждый каталог и файл файловой системы имеет уникальное полное имя (в ОС UNIX это имя принято называть full pathname). Каталог, являющийся корнем файловой системы, в любой файловой системе имеет предопределенное имя «/» (слэш). Полное имя файла, например, /bin/sh означает, что в корневом каталоге должно содержаться имя каталога bin, а в каталоге bin должно содержаться имя файла sh. Коротким или относительным именем файла (relative pathname) называется имя, задающее путь к файлу от текущего рабочего каталога.

В каждом каталоге содержатся два специальных имени, имя «.» именованное сам этот каталог, и имя «..», именованное «родительский» каталог данного каталога, то есть каталог, непосредственно предшествующий данному в иерархии каталогов.

Файловая система обычно размещается на дисках или других устройствах внешней памяти, имеющих блочную структуру. Кроме блоков, сохраняющих каталоги и файлы, во внешней памяти поддерживается еще несколько служебных областей.

В семействе UNIX существует несколько разных видов файловых систем со своей структурой внешней памяти. Наиболее известны традиционная файловая система UNIX System V (s5) и файловая система семейства UNIX BSD (ufs). Файловая система s5 состоит из четырех секций (рисунок 83а). В файловой системе ufs на логическом диске находится последовательность секций файловой системы (рисунок 83б).

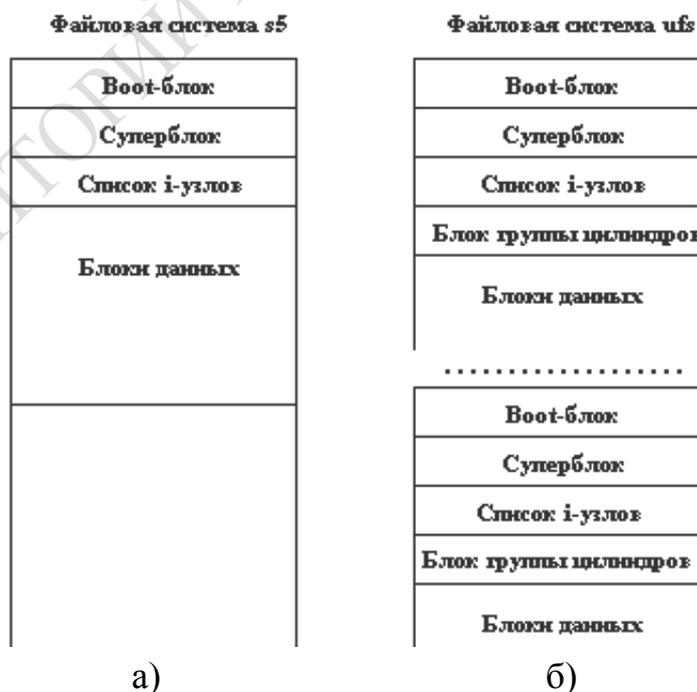


Рисунок 83 – а) файловая система s5; б) файловая система ufs

7.5 Особенности работы файловой системы NTFS

Файловая система NTFS (New Technology File System) является самой популярной среди систем подобного рода и применяется практически на всех современных компьютерах. Всё дело в том, что эта система установлена почти на всех компьютерах, где используется ПО Microsoft, во всех ОС, созданных на базе Windows NT. Сегодня NTFS – одна из самых надежных и сложных файловых систем.

Одна из особенностей этой файловой системы – это то, что локальные диски, на которых она установлена, могут быть абсолютно любого размера. У файловой системы NTFS нет ограничения по размерам.

Файловая система NTFS представляет собой реляционную базу данных с очень сложной архитектурой. По своему строению она очень похожа на запутанный египетский лабиринт.

Файловая система NTFS делится на тома и разделы, ведущим разделом в которой является том, совпадающий с разделом. Тома включают в себя несколько разделов. Отличительной чертой NTFS является то, что служебные структуры не привязаны к определенному месту, а могут находиться в совершенно разных частях тома, что даёт им возможность при необходимости разделять себя на несколько частей посредством фрагментирования.

Самым главным и важным файлом среди служебных, в этой файловой системе служит \$MFT – Master File Table. Эта таблица является базой данных, в которой хранится информация об именах файлов, их атрибутах, а также месторасположении на диске. Помимо этого таблица содержит файлы, находящиеся в дочерних директориях диска. Также имеются и разные дополнительные файлы вспомогательного назначения, их обозначает значок доллара, который является признаком информации дополнительного характера (рисунок 84).

Локальный диск файловой системы NTFS делится на две зоны. Первая зона занимает 12 % дискового пространства и называется MFT-зоной. Эта зона является зоной хранения метафайла (MFT), и поэтому она защищена от записи файлов. Оставшиеся 88 % дискового пространства служат для хранения обычных файлов.

Файловая система NTFS также содержит и директории со ссылками на файлы, каталоги и блоки. Каждый файл в директории разделен на несколько блоков, в которых хранятся имена файлов, их базовые атрибуты и ссылка на их MFT-элемент.

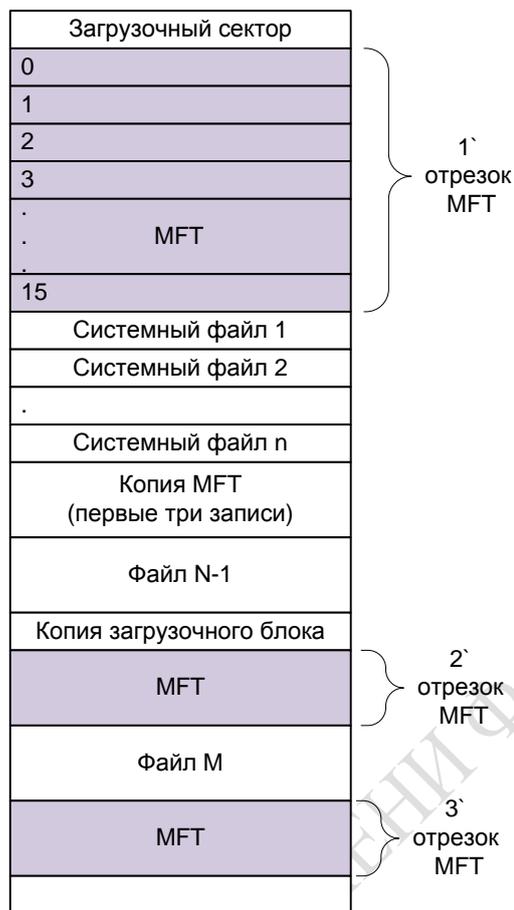


Рисунок 84 – Структура тома NTFS

Ещё одной уникальной чертой системы NTFS является её способность автоматически восстанавливать нормальное функционирование после сбоя.

В случае непредвиденного отключения питания система запоминает своё состояние и при включении компьютера перезагружается в своём последнем состоянии.

8 КОНФИГУРИРОВАНИЕ ОПЕРАЦИОННЫХ СИСТЕМ

8.1 Настройка файла `autoexec.bat` и `config.sys` в ОС MS-DOS

Файл `Config.sys` является текстовым файлом, в нем могут содержаться строки следующих типов [18]:

1 Команды, задающие параметры конфигурации MS-DOS. Они имеют следующий вид: имя-команды = значение.

2 Комментарии – это строки, которые всегда игнорируются при обработке файла `Config.sys`. Строки комментариев начинаются с символа ‘;’ или с зарезервированной команды `REM` с последующим пробелом.

3 Заголовки блоков – это строки, в первой позиции которых указывается имя блока, заключенное в квадратные скобки. Заголовок блока предваряет строки с командами, относящимися к данному блоку. Команды блока выполняются в зависимости от выбора пользователя в стартовом меню.

Наиболее часто используемые команды файла `Config.sys`:

Device – загрузка драйвера в обычную память: `Device = имя-драйвера (параметры)`.

DeviceHigh – загрузка драйвера в верхнюю память: `DeviceHigh = имя-драйвера (параметры)`.

Install – установка резидентной программы: `Install = имя-программы (параметры)`.

Buffers – установка числа буферов ввода-вывода: `Buffers = число-буферов`.

Files – установка числа открытых файлов: `Files = число-файлов`.

FCBs – задание числа блоков управления файлами: `FCBs = число-блоков-FCB`.

LastDrive – задание последней буквы диска: `LastDrive = буква`.

Stacks – резервирование стеков: `Stacks = количество, размер`.

Shell – задание командного интерпретатора MS-DOS: `Shell = имя-командного-процессора (параметры)`.

Switches – управление начальной загрузкой MS-DOS: `Switches /F /N`.

Break – управление аварийным завершением программ: `Break = on | off`.

Country – задание страны: Country = код-страны, кодовая-страница, имя-файла.

После загрузки ОС MS-DOS автоматически выполняется командный файл Autoexec.bat. Данный файл является обычным командным файлом, поэтому при его создании можно использовать любые команды, используемые в пакетных файлах: Rem, Echo, Goto, IF, For, Choice, и др.

С помощью файла Autoexec.bat можно произвести необходимую настройку ОС, установить удобное для работы с ней окружение, а также задать команды, которые должны выполняться каждый раз после загрузки MS-DOS. Как правило, в файле Autoexec.bat содержатся следующие группы команд:

- 1) команды запуска резидентных программ и других программ, которые целесообразно запускать при каждой загрузке MS-DOS;
- 2) команды SET для установки переменных окружения MS-DOS;
- 3) команда Path для задания списка каталогов, в которых будет производиться поиск запускаемых программ;
- 4) команда Prompt для установки формата приглашения MS-DOS.

Prompt – установка формата приглашения MS-DOS: Prompt (формат-приглашения).

Path – задание путей для поиска программ: Path имя-каталога1 (; имя-каталога2) (; имя-каталога3)...

Set – установка переменных окружения MS-DOS: имя-переменной = значение.

8.2 Автоматическое обслуживание настроек ОС

Майкрософт неоднократно заявляли, что Windows 8 была создана на основе предшествующих разработок. Невозможно отрицать этот факт, действительно, многие компоненты системы построены на старых проверенных решениях, которые делают её быстрой, стабильной и современной, как и предыдущие версии [5].

Один из обновленных компонентов – это автоматическое обслуживание системы, которое следит за обновлениями программного обеспечения, выполняет диагностику системы, следит за безопасностью, за своевременным выполнением операций обслуживания системы, когда вас нет рядом и система простаивает без дела.

Программа для автоматического сопровождения Windows 8, называется MSchedExe.exe, и она находится в каталоге C:\Windows\System32.

Просмотреть данную компоненту можно в разделе «Обслуживание» центра поддержки (рисунок 85).

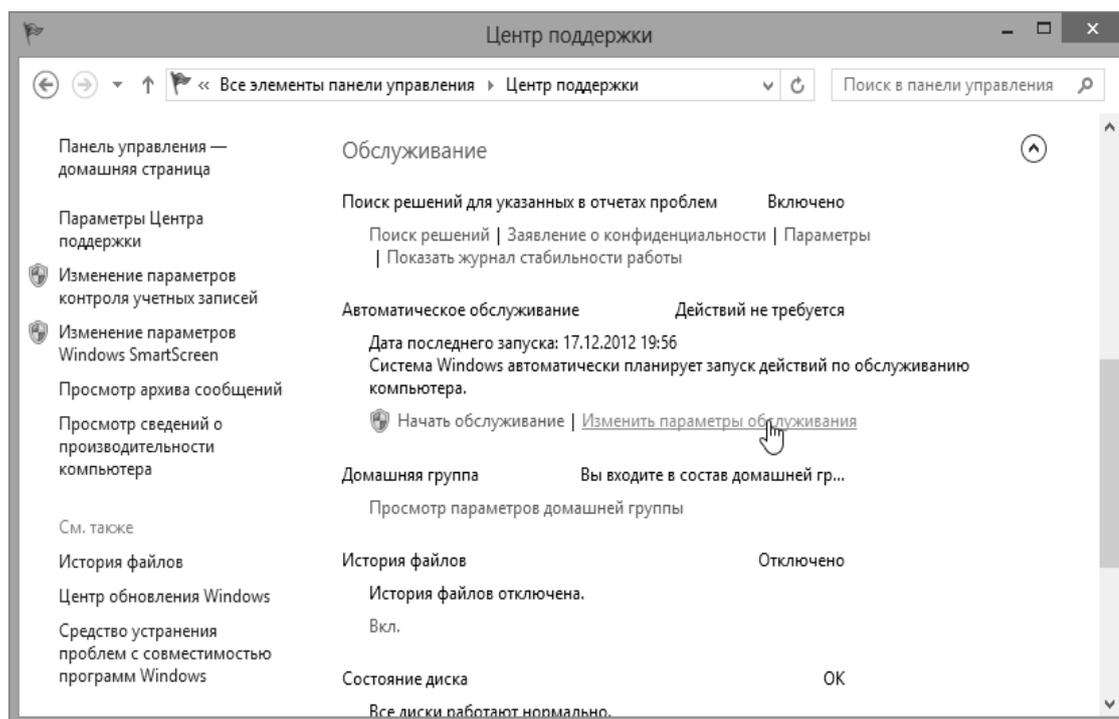


Рисунок 85 – Центр поддержки Windows 8

Время запуска обслуживания происходит в момент, когда система переходит в режим бездействия. Windows проверяет свою активность с помощью службы планировщика заданий каждые 15 минут. Система считается бездействующей, если более 90 % этого времени мышь и клавиатура не использовались, а процессоры и диски не проявляли активности. Если начать взаимодействовать с системой во время ее обслуживания, оно немедленно прекратится.

Автоматическое обслуживание учитывает различия в политике электропитания для мобильных и стационарных ПК, исходные настройки которых отличаются. Например, одной из них является пробуждение по таймеру для выполнения запланированных задач.

Выполнение автоматического обслуживания Windows, как и прежде, осуществляется с помощью планировщика (рисунок 86), однако теперь там появилась новая группа заданий для этой цели.

Внизу списка видно то самое задание *Regular Maintenance*, время ежедневного запуска, которое можно настроить в панели управления. В планировщике можно изменять расписание запуска, например, на еженедельное. Однако делать это вовсе необязательно, ибо в итоге все упирается в бездействие ПК.

Файл	Состояние	Триггеры
Idle Maintenance	Отключе...	При бездействии компьютера
Maintenance Configurator	Готово	Определено несколько триггеров
Manual Maintenance	Готово	
Regular Maintenance	Готово	В 3:00 каждый день

Рисунок 86 – Планировщик задач в Windows 8

Если запустить задание *Manual Maintenance*, система выполнит тот же самый набор задач по обслуживанию, что и при запуске задания по расписанию. Отличие в том, что взаимодействие с системой не прервет процесс оптимизации, поскольку его запустил сам пользователь.

Отключенное задание *Idle Maintenance* любопытно тем, что система сама включает его при выполнении ручного или автоматического обслуживания.

Задание *Maintenance Configurator* определяет список задач, обслуживающих систему.

Ключевое различие в автоматическом обслуживании Windows 7 и Windows 8 в оптимизации энергопотребления операционной системой. Раньше многие задачи по обслуживанию действовали исключительно по своему расписанию и могли выполняться при работе от батареи, а теперь они привязаны к единой задаче, которая запускается в период бездействия только при питании от сети и выполняется как можно быстрее.

Это не означает, что все задания планировщика теперь выполняются только в рамках автоматического обслуживания, поскольку у них может быть определено несколько триггеров. Например, точка восстановления системы обязательно создается при установке обновлений и драйверов. Однако отсутствие триггера в задании указывает на то, что оно выполняется исключительно в рамках автоматического обслуживания.

8.3 Антивирусная профилактика средств мобильной связи

Любое устройство, имеющее установленную операционную систему, должно быть защищено антивирусным ПО [6]. Это поможет

избежать ситуаций, когда устройство с помощью сторонних приложений, без ведома человека использует трафик, а также совершает звонки и другие действия, наносящие вред.

Основные функции антивирусов для средств мобильной связи:

- комплексная защита для Android-телефона или планшета;
- защита от вредоносных программ, шпионского ПО, хакеров и других мобильных угроз;
- обеспечение безопасного хранения личных и финансовых данных на телефоне;
- поддержание оптимизированного состояния системы Android на вашем смартфоне;
- резервное копирование и восстановления контактов между различными платформами;
- определение местоположения, дистанционная блокировка и удаление информации с потерянного или украденного Android-устройства;
- контроль над запущенными приложениями, энергопотреблением и использованием интернет-трафика.
- блокировка вызовов и SMS-сообщений.

Kaspersky Mobile Security. Продукт «Лаборатории Касперского», обеспечивающий комплексную защиту девайсов на базе Windows Mobile и Symbian OS от различного типа угроз (рисунок 87).

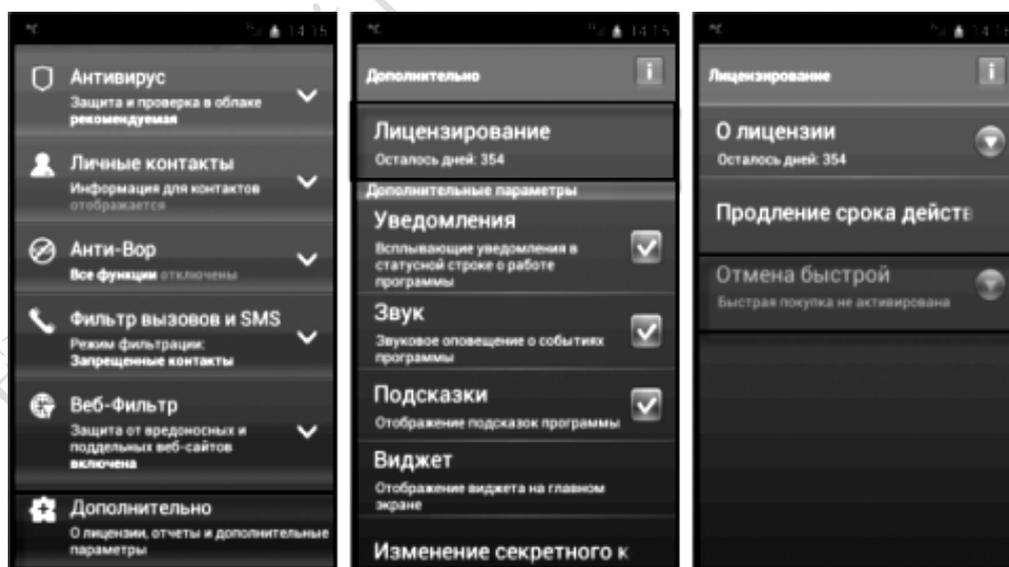


Рисунок 87 – Антивирус Kaspersky Mobile Security

Помимо распознавания мобильных вирусов программа посредством шифрования защищает данные пользователя от попадания

в чужие руки, предоставляет возможность найти потерянный или украденный аппарат, получив ссылку на его точные координаты в системе Google Maps (при наличии в устройстве модуля GPS), умеет блокировать нежелательные звонки и SMS, а также пресекать опасные сетевые соединения посредством брандмауэра. Стремясь заслужить доверие со стороны старшего поколения, специалисты компании оснастили приложение модулем родительского контроля, блокирующим звонки и отправку SMS на определенные номера и позволяющим определять местонахождение ребенка в случае необходимости. Последняя опция, опять-таки, функционирует только в случае, если мобильник поддерживает работу с GPS-навигацией.

ESET NOD32 Mobile. Разработка компании ESET (рисунок 88). Антивирус обеспечивает информационную защиту смартфонов, коммуникаторов и КПК на платформе Windows Mobile от SMS-спама, вирусов, троянских и шпионских модулей и другого опасного и нежелательного программного обеспечения. В основе приложения лежит технология проактивной защиты ThreatSense, по словам разработчика позволяющая точно и оперативно распознавать новое неизвестное вредоносное ПО ещё до его внесения в сигнатурную базу. Эвристический анализатор сканирует полученные через Bluetooth, WiFi, инфракрасный порт, GPRS и EDGE данные на наличие вредоносного кода, предотвращая попытки установки опасных программ на мобильное устройство. Все подозрительные файлы помещаются в раздел «карантин», и пользователь может по своему усмотрению восстановить их или удалить. Сообщения рекламного, мошеннического и другого нежелательного содержания отслеживаются с помощью спам-фильтра и отправляются в соответствующий раздел.

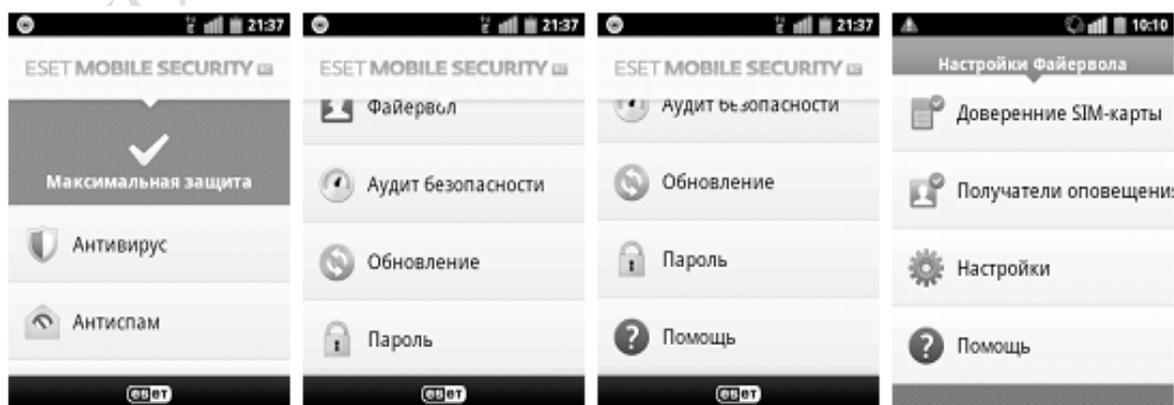


Рисунок 88 – Антивирус ESET NOD32 Mobile

Антивирус *Dr.Web* для Windows Mobile (рисунок 89). Предназначен для защиты мобильных устройств – КПК и коммуникаторов любых моделей – на которых установлена ОС Windows Mobile 2003/2003 SE/5.0./6.0/ 6.1. Антивирус защищает от вирусов и других интернет-угроз, созданных специально для инфицирования мобильных устройств. Продукт не защищает от вирусов и вредоносных программ для ПК.

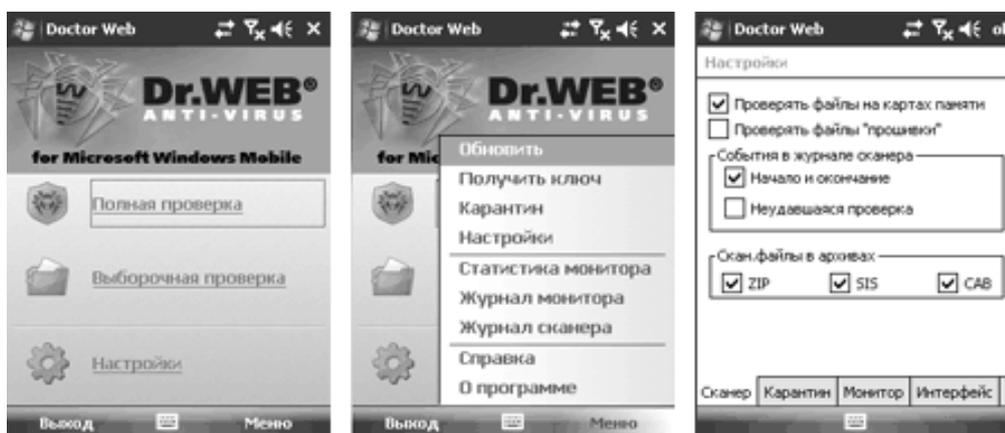


Рисунок 89 – Антивирус Dr. Web

Norton Smartphone Security. Творение корпорации Symantec, добавляющее спокойствия обладателям WM и Symbian-девайсов. Программа стойко защищает принадлежащий пользователю цифровой аппарат от вирусов, троянов, червей, SMS-спама и даже хакерских атак извне (рисунок 90).



Рисунок 90 – Антивирус Norton Smartphone Security

8.4 Понятие групповой политики в сетях Microsoft

Структура многопользовательских операционных систем предполагает возможность создания для отдельного пользователя индивидуального окружения [1]. В окружение пользователя могут входить:

- конфигурации рабочего стола и индивидуальные настройки оболочки;
- доступные пользователю приложения;
- сценарии, выполняющиеся при входе пользователя в систему или выходе из нее;
- ассоциированные с пользователем права и разрешения на доступ к локальным и сетевым информационным ресурсам.

Для управления разрешениями пользователей в доменах Windows используется механизм *групповых политик*.

Под *групповой политикой* понимается совокупность параметров, используемых для конфигурирования рабочего окружения пользователя или компьютера (рисунок 91).

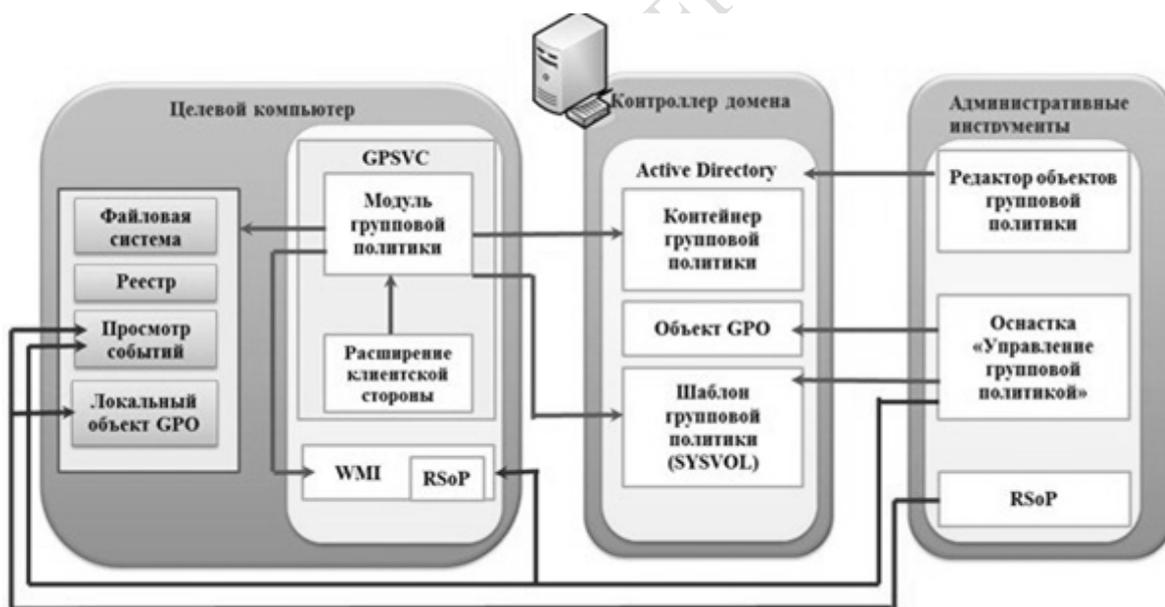


Рисунок 91 – Компоненты групповой политики

Механизм групповых политик – основа централизованного управления конфигурациями пользователей и компьютеров в корпоративной сети.

Групповая политика в доменах Windows применяется к двум основным категориям сетевых объектов – компьютерам и пользователям домена.

Пользователи – групповая политика регламентирует окружение конкретных пользователей независимо от того, на каком компьютере эти пользователи работают.

Компьютеры – групповая политика определяет параметры системы, влияющие на окружение пользователей, для конкретных компьютеров независимо от того, какие пользователи на них работают.

Особенности групповых политик:

- объекты GPO применяются в отношении контейнеров, а не замыкающих объектов;

- один контейнер может быть связан с несколькими объектами GPO;

- объекты GPO, связанные с одним и тем же контейнером, применяются в отношении этого контейнера в том порядке, в котором они были назначены;

- объект GPO включает в себя две составляющие: параметры, относящиеся к компьютеру, и параметры, относящиеся к пользователю;

- обработку любой из этих составляющих можно отключить;

- наследование объектов GPO можно блокировать;

- наследование объектов GPO можно форсировать;

- применение объектов GPO можно фильтровать при помощи списков ACL.

Объекты групповой политики (GPO) – основной элемент групповой политики, выступающий в качестве самостоятельных элементов каталога.

GPO входят как объекты службы каталогов Active Directory.

С каждым объектом групповой политики связан глобальный уникальный идентификатор – GUID. Для управления ими используются специальные инструменты – редакторы групповых политик, программные интерфейсы.

Любой объект групповой политики может быть связан с некоторым объектом контейнерного типа в каталоге, относящемся к одному из трех классов: узел (сайт), домен, организационная единица.

На каждом компьютере, под управлением Windows существует специальный объект групповой политики – локальная групповая политика.

Литература

1. Реймер, С. Active Directory для Windows Server 2003. Справочник администратора / С. Реймер, С. Малкер; пер. с англ. – М. : «СП ЭКОМ», 2004. – 512 с: ил.
2. Пильщиков, В. Assembler. Программирование на языке ассемблера IBM PC / В. Пильщиков. – М. : Диалог-МИФИ, 2005. – 288 с.
3. Intel теряет интерес к настольным компьютерам [Электронный ресурс]. – Режим доступа: <http://mnenia.ru/rubric/tech/intel-teryayet-interes-k-nastolnym-kompyuteram>. – Дата доступа: 26.01.2013. – Загл. с экрана.
4. UNIX [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/UNIX>. – Дата доступа: 30.01.2015. – Загл. с экрана.
5. Windows [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Windows>. – Дата доступа: 30.01.2015. – Загл. с экрана.
6. Антивирусная программа [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Антивирусная_программа. – Дата доступа: 12.04.2015. – Загл. с экрана.
7. Таненбаум, Э. Архитектура компьютера / Э. Таненбаум ; пер. с англ. – Изд. 5-е. – СПб. : Питер, 2007. – 848 с.
8. Паттерсон, Д. Архитектура компьютера и проектирование компьютерных систем / Д. Паттерсон, Дж. Хеннесси. – СПб. : Питер, 2012. – 784 с.
9. Банкомат [Электронный ресурс] // Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Банкомат>. – Дата доступа: 05.02.2015. – Загл. с экрана.
10. Беспроводные технологии [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Беспроводные_технологии. – Дата доступа: 05.03.2015. – Загл. с экрана.
11. Иртегов, Д. Введение в операционные системы / Д. Иртегов. – СПб. : БХВ-Петербург, 2012. – 1040 с.
12. Виртуальная машина [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Виртуальная_машина. – Дата доступа: 15.03.2015. – Загл. с экрана.
13. Виртуальная память [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Виртуальная_память. – Режим доступа: 15.04.2015. – Загл. с экрана.
14. Мелехин, В. Вычислительные машины, системы и сети / В. Мелехин, Е. Павловский. – М. : Academia, 2010. – 560 с.
15. Интерфейс [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Интерфейс>. – Дата доступа: 18.02.2015. – Загл. с экрана.
16. Каталог X-driver / TM-4104R [Электронный ресурс]. – Режим доступа: <http://www.texet.ru/smartphones/tm4104R.htm>. – Дата доступа: 10.06.2015. – Загл. с экрана.

17. Рэндал, Э. Компьютерные системы: архитектура и программирование / Э. Брайант Рэндал, Дэвид Р. О'Халларон ; пер. с англ. : Д. Ежов, С. Шестаков. – СПб. : БХВ-Петербург, 2005. – 1186 с.
18. Воруев, А. В. Лабораторный практикум по операционным системам : учеб. пособие / А. В. Воруев, О. М. Демиденко, А. И. Кучеров. – Барановичи : Баранов. крупн. тип., 2002. – 126 с.
19. Межсетевой экран [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Межсетевой_экран. – Дата доступа: 04.03.2015. – Загл. с экрана.
20. Мобильный компьютер [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Мобильный_компьютер. – Дата доступа: 11.02.2015. – Загл. с экрана.
21. НОУ ИНТУИТ | Лекция | Управление памятью. Методы, алгоритмы и средства [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/631/487/lecture/11057?page=2>. – Дата доступа: 05.04.2015. – Загл. с экрана.
22. Сеницын, С. Операционные системы / С. Сеницын, А. Батаев, Н. Налютин. – М. : Academia, 2012. – 304 с.
23. Дейтел, Х. Операционные системы. Том 2. Распределённые системы, сети, безопасность / Х. Дейтел, П. Дейтел, Д. Р. Чофнес ; пер. с англ. : А. Бутко [и др.]. – СПб. : Бинوم, 2011. – 704 с.
24. Дейтел, Х. Операционные системы. Часть 1. Основы и принципы / Х. Дейтел, П. Дейтел, Д. Р. Чофнес ; пер. с англ. : А. Бутко [и др.]. – СПб. : Бинوم-Пресс, 2011. – 1024 с.
25. Поколения мобильной телефонии [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Поколения_мобильной_телефонии. – Дата доступа: 08.02.2015. – Загл. с экрана.
26. Нортон, П. Программно-аппаратная организация IBM PC / П. Нортон ; пер. с англ. – М. : Радио и связь, 1992. – 336 с.
27. Сервер (аппаратное обеспечение) [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Сервер_\(аппаратное_обеспечение\)](https://ru.wikipedia.org/wiki/Сервер_(аппаратное_обеспечение)). – Дата доступа: 01.03.2015. – Загл. с экрана.
28. Файл [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Файл>. – Дата доступа: 01.04.2015. – Загл. с экрана.

Учебное издание

Воруев Андрей Валерьевич,

Чечет Павел Леонидович

**ОПЕРАЦИОННЫЕ СИСТЕМЫ И СЕТИ :
СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

Учебно-методическое пособие

Редактор *В. И. Шкредова,*

Корректор *В. В. Калугина*

Подписано в печать 16.12.2016. Формат 60×84 1/16.

Бумага офсетная. Ризография. Усл. печ. л. 7,67.

Уч.-изд. л. 8,39. Тираж 15 экз. Заказ 717.

Издатель и полиграфическое исполнение:

учреждение образования

«Гомельский государственный университет

имени Франциска Скорины».

Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/87 от 18.11.2013.

Специальное разрешение (лицензия) № 02330 / 450 от 18.12.2013.

Ул. Советская, 104, 246019, Гомель

А. В. Воруев, П. Л. Чечет

**ОПЕРАЦИОННЫЕ
СИСТЕМЫ И СЕТИ:
СИСТЕМНОЕ ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ**

Гомель
2016

