

28. Параллельный и последовательный порты.

Параллельный и последовательный порты используются не только для подключения принтера и модема, для которых были в свое время разработаны. Простота исполнения и отработанный протокол приема-передачи данных сделали их незаменимыми для подключения к ПЭВМ различных низкоскоростных устройств, применяющихся в промышленности и научных исследованиях.

Параллельный порт (интерфейс Centronics).

Основным назначением интерфейса Centronics (аналог — ИРПР-М) является подключение к компьютеру принтеров различных типов.

Поэтому распределение контактов разъема, назначение сигналов, программные средства управления интерфейсом ориентированы именно на это использование.

В то же время с помощью данного интерфейса можно подключать к компьютеру и другие специально разработанные внешние устройства.

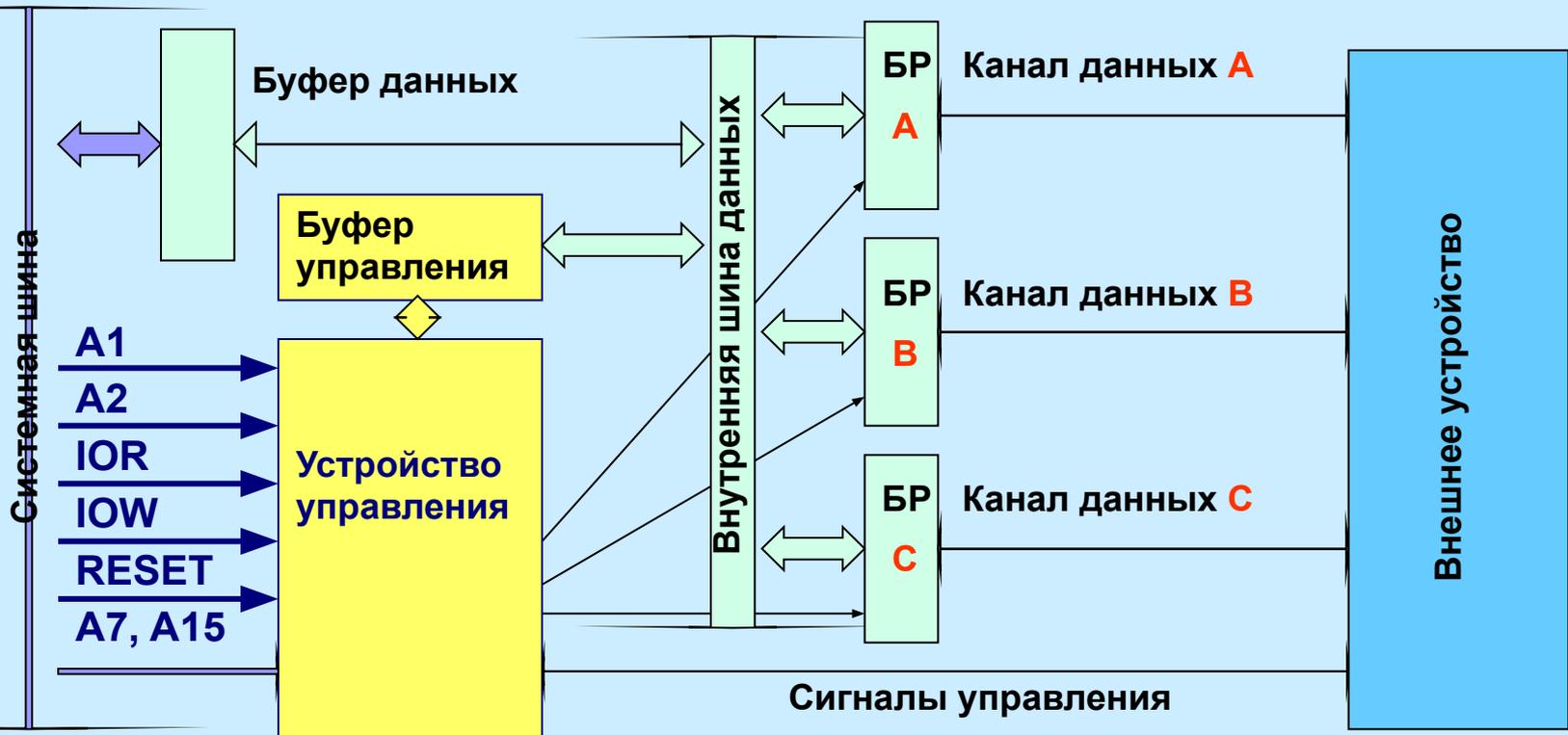
Скорость обмена по интерфейсу Centronics – 129-200 кБ/с.

Стандартный параллельный порт предназначен только для односторонней передачи информации от ПЭВМ к принтеру.

Усовершенствованный порт **EPP (Enhanced Parallel Port)** является двунаправленным, позволяет подключать до 64-х устройств и обеспечивает скорость передачи данных с использованием ПДП до 2 МБ/с.

Расширенный порт **ECP (Extended Capability Port)** позволяет подключить до 128 устройств и поддерживает режим компрессии (сжатия) данных.

Схема контроллера параллельного порта i8255A.



Контроллер параллельного обмена представляет собой 3-канальный байтовый интерфейс и позволяет организовать обмен данными в трех режимах:

Режим 0 – синхронный однонаправленный ввод/вывод (4 порта А, В, С1, С2).

Режим 1 – асинхронный однонаправленный ввод/вывод (2 порта А и В).

Режим 2 – асинхронный двунаправленный ввод/вывод (1 порт А).

Программирование режимов работы каналов контроллера осуществляется передачей в буфер управления соответствующего кода.

Сигналы Centronics имеют следующее назначение:

- D0...D7** — 8-разрядная шина данных для передачи из компьютера в принтер.
- STROBE** — сигнал стробирования (сопровождения) данных.
- ACK** — сигнал подтверждения принятия данных и готовности принтера.
- BUSY** — сигнал занятости принтера обработкой полученных данных и неготовности принять следующие данные.
- AUTO FD** — сигнал автоматического перевода строки (каретки).
- PE** — сигнал конца бумаги (режим ожидания).
- SLCT** — сигнал готовности приемника (принтера).
- SLCT IN** — сигнал принтеру о том, что последует передача данных.
- ERROR** — сигнал ошибки принтера.
- INIT** — сигнал инициализации (сброса) принтера и очистки буфера печати.

Контакт разъема компьютера	Цепь	I/O	Контакт разъема принтера
1	-STROBE	O	1
2 ... 9	DO ... D7	O	2 ... 9
10	-ACK	I	10
11	BUSY	I	11
12	PE	I	12
13	SLCT	I	13
14	-AUTOFD	O	14
15	-ERROR	I	32
16	-INIT	O	31
17	-SLCT IN	O	36
18...25	GND	-	16, 17, 19...30, 33

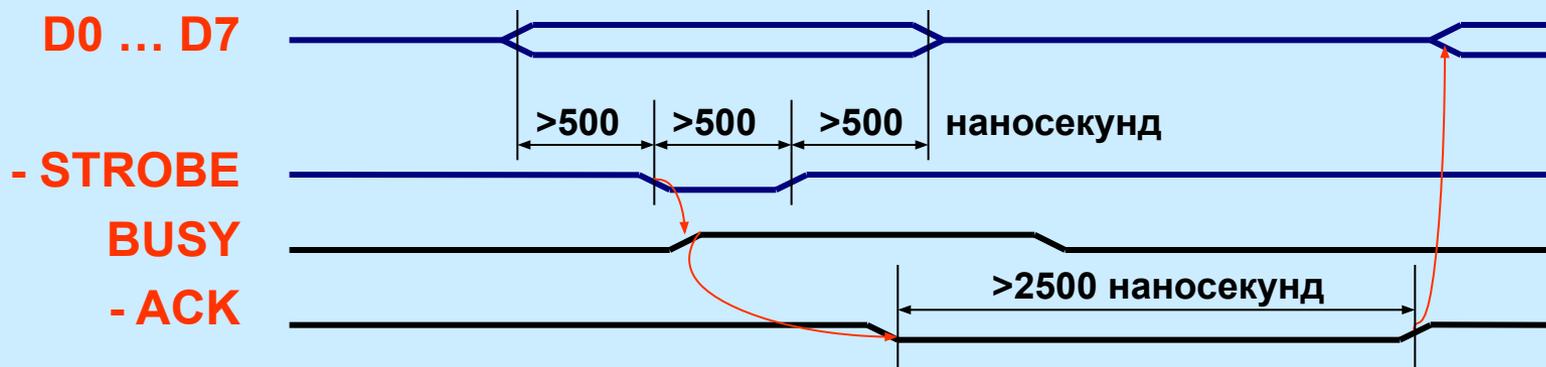
Формирование и прием сигналов интерфейса Centronics производится путем записи и чтения выделенных для него портов ввода/вывода.

В компьютере может использоваться три порта Centronics:

LPT1 (порт 378h, IRQ5), **LPT2** (порт 278h, IRQ7) и **LPT3** (порт 3BCh).

Базовый адрес порта используется для передачи принтеру байта данных. Установленные на линиях данные можно считать из этого же порта в ПЭВМ.

Временная диаграмма цикла передачи данных по интерфейсу Centronics.



Перед началом передачи данных контролируется снятие сигналов **BUSY** и **ASK**. Затем данные выставляются на шину и формируется сигнал **STROB**. За это время принтер должен успеть принять данные и выставить сигнал **BUSY**, а затем и **ASK**.

Максимальная длина соединительного кабеля – 1,8 метра.

В настоящее время стандарты параллельного порта **EPP** и **ECP** включены в стандарт **IEEE 1284** с добавлением еще двух режимов обмена данными: байтового и полубайтового.

Последовательный порт (Интерфейс RS232C).

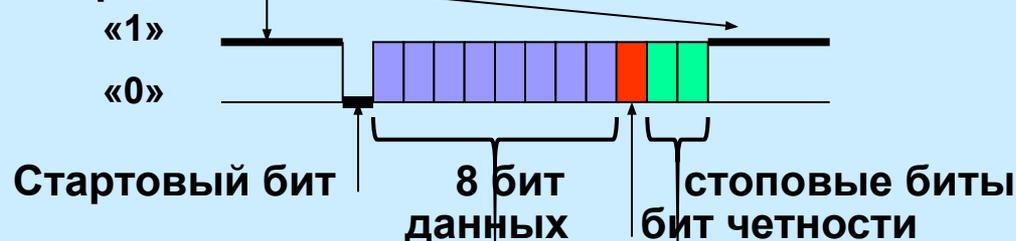
Интерфейс RS-232C предназначен для подключения к компьютеру стандартных внешних устройств (принтера, сканера, модема, мыши и т.д.), а также для связи компьютеров между собой.

Основными преимуществами использования RS-232C по сравнению с Centronics являются возможность передачи на значительно большие расстояния и гораздо более простой соединительный кабель.

В то же время работать с ним несколько сложнее: данные в RS-232C передаются в последовательном коде побайтно, а каждый байт обрамляется стартовым и стоповыми битами.

Формат передаваемых данных последовательного порта:

отсутствие передачи

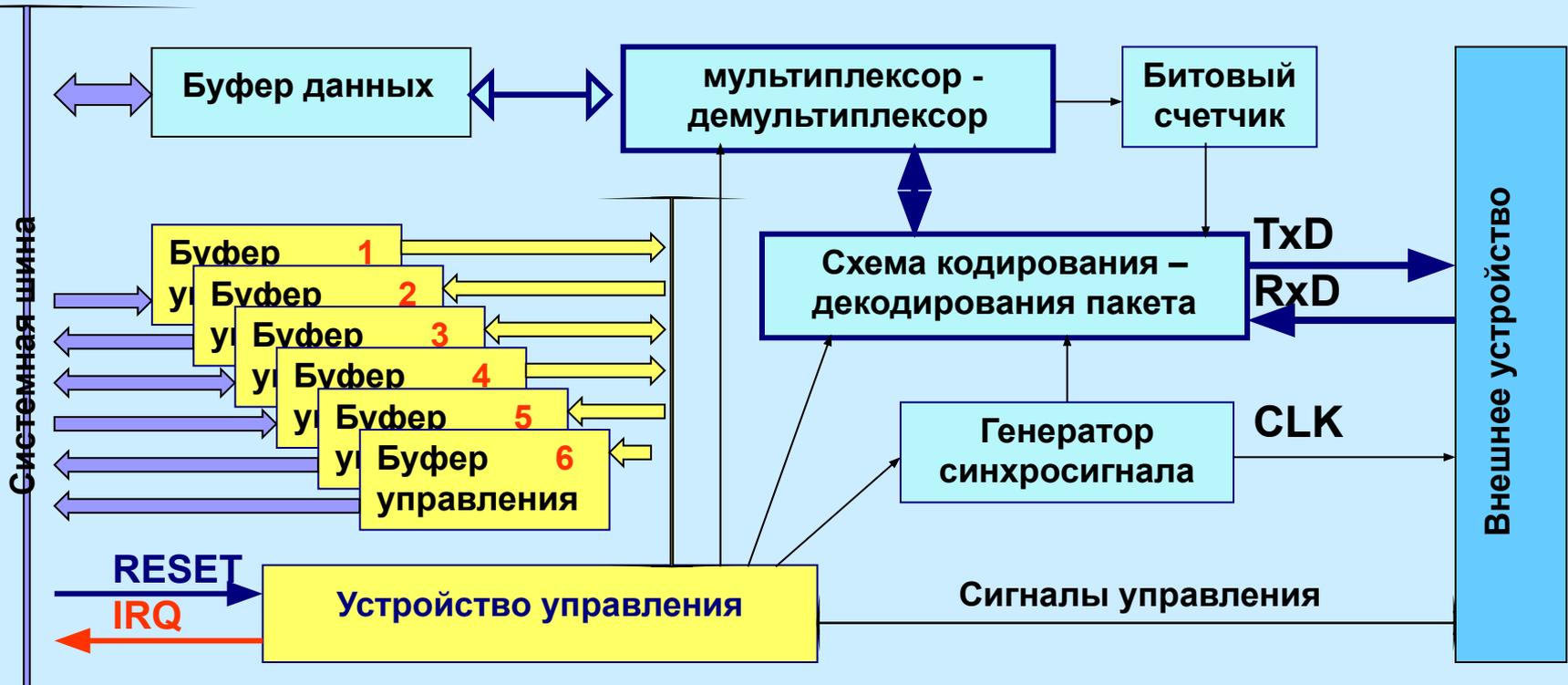


Данные могут передаваться как в одну (полудуплексный режим), так и в обе стороны (дуплексный режим).

Обмен по интерфейсу RS-232C осуществляется по специально выделенным для этого последовательным портам:

- COM1** (адреса 3F8h...3FFh, прерывание IRQ4),
- COM2** (адреса 2F8h...2FFh, прерывание IRQ3),
- COM3** (адреса 3E8h...3EFh, прерывание IRQ10),
- COM4** (адреса 2E8h...2EFh, прерывание IRQ11).

Схема контроллера последовательного порта i8250.



В состав ПЭВМ могут входить до четырех последовательных портов, работающих в стандарте RS-232C (отечественный аналог - стык С2).

Каждое из устройств RS-232C представляет собой самостоятельный контроллер i8250, оснащенный 25- или 9- штырьковым разъемом.

Контроллер порта RS-232C является полностью программируемым Устройством.

Ему можно задать следующие параметры обмена: количество битов данных и стоп-битов, вид четности и скорость обмена в бодах (бит/с).

Назначение сигналов обращений:

FG — защитное заземление (экран).

-TxD — данные, передаваемые компьютером в последовательном коде.

-RxD — данные, принимаемые компьютером в последовательном коде.

RTS — сигнал запроса передачи. Активен во все время передачи.

CTS — сигнал сброса (очистки) для передачи. Активен во все время передачи. Говорит о готовности приемника.

DSR — готовность данных. Используется для задания режима модема.

SG — сигнальное заземление, нулевой провод.

DCD — обнаружение несущей данных (принимаемого сигнала).

DTR — готовность выходных данных.

RI — индикатор вызова. Говорит о приеме модемом сигнала вызова по телефонной сети.

Компьютер обычно имеет 9-контактный (DB9P) или 25-контактный (DB25P) разъем для подключения интерфейса RS-232C.

Назначение контактов разъема приведено в таблице

сигнал	25-контактный разъем	9-контактный разъем	I/O
FG	1	-	-
-TxD	2	3	O
-RxD	3	2	I
RTS	4	7	O
CTS	5	8	I
DSR	6	6	I
SG	7	5	-
DCD	8	1	I
DTR	20	4	O
RI	22	9	I

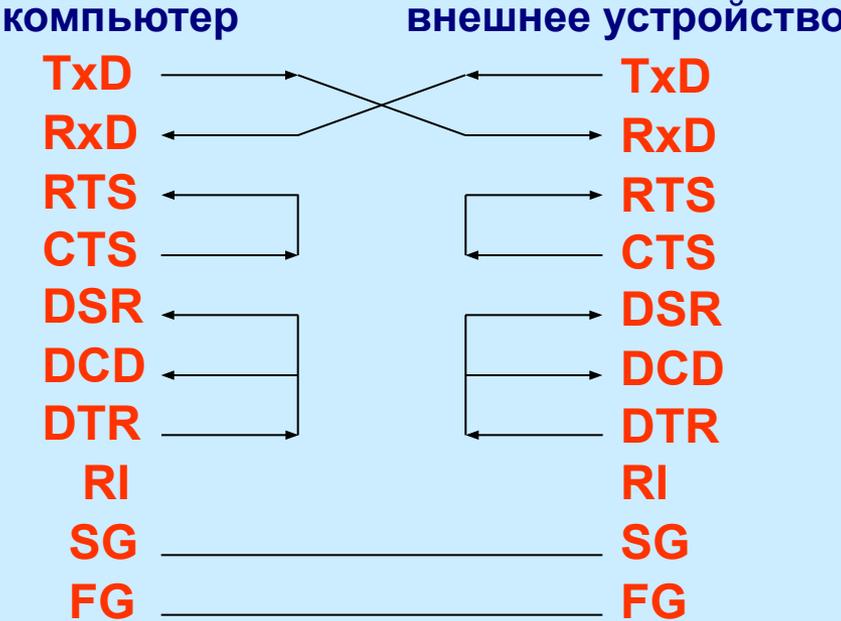
Конкретные форматы обращений по этим портам можно найти в описаниях микросхем контроллеров последовательного обмена UART (Universal Asynchronous Receiver/ Transmitter), например, для i8250.

Наиболее часто используются трех- или четырехпроводная связь (для двунаправленной передачи).

Для двухпроводной линии связи в случае только передачи из компьютера во внешнее устройство используются сигналы SG и TxD.

Все 10 сигналов интерфейса задействуются только при соединении компьютера с модемом.

Схема 4-х проводной линии связи для интерфейса RS232C



Примечание: тактовые частоты приемника и передатчика должны быть одинаковыми (расхождение – не более 10%) для этого скорость передатчика (ПЭВМ) может выбираться из ряда: 150, 300, 600, 1200, ... 57600, 115200 бит/с.

29. Подсистема ввода-вывода.

Назначение, структура и задачи BIOS.

BIOS (Basic Input Output System) – часть программного обеспечения ПЭВМ, содержащая управление адаптерами внешних устройств, экранными операциями, тестирование, а затем начальную загрузку OS.

BOIS обеспечивает стандартный интерфейс, поддерживающий переносимость OS для ПЭВМ с совместимыми процессорами.

BOIS состоит из основных компонент:

- 1. POST– процедуры проверки системных устройств и их ресурсов.**
- 2. ROM-Scan – программа сканирования ОЗУ.**
- 3. SETAP-программный интерфейс просмотра и корректировки констант.**
- 4. Константы BIOS в CMOS, таблица 256-ти символов ASCII 8x8.**

Все адреса констант документированы и должны сохраняться в последующих версиях BIOS для данной системной платы.

Компоненты BOIS записаны в специальной ПЗУ на системной плате объемом 64 кБ и обычно рассматриваются, как неотъемлемая часть ПЭВМ, встраиваемая в адресное пространство ОЗУ с адреса F000:0000.

Основной принцип организации системы ввода/вывода:

ЦПУ и ОЗУ образуют ядро ПЭВМ, а различные периферийные устройства, к которым можно отнести любое другое устройство, которое не входит в состав ядра ПЭВМ, сопрягаются с ядром системы с помощью интерфейсов (совокупности шин, сигналов, электрических схем, протоколов передачи данных и команд), входящих в состав ядра OS для организации обмена информацией.

Структура подсистемы POST.

Состояние процессора после включения питания predetermined –
EFLAGS = 00000002h; **EIP** = 0000FFF0h; **CS** = 0F000h; **PE(CR0)** = 0.

по этому адресу находится команда **JMP** перехода на процедуру **POST (Power On Self Test)** самотестирования и инициализации базовых устройств ПЭВМ:



Основные функции подсистемы POST.

Процедуры POST служат для пуска самотестирования устройств на системной плате, сравнения их статуса с данными CMOS и инициализации:

- каналы системного интервального таймера (слышен гудок),
- контроллер прерываний,
- контроллер прямого доступа,
- контроллер клавиатуры (загораются индикаторы на клавиатуре),
- контроллер памяти и т.д.,

затем инициализируются процедуры самотестирования устройств, имеющих собственную BIOS:

- видеоконтроллер (на мониторе появляется первое сообщение),
- контроллеры дисковых накопителей (появляется сообщение),
- контроллер USB (клавиатура или мышь USB становятся активными)
- звуковой адаптер,
- сетевой адаптер и т.д.

Выполняется сканирование оперативной памяти (отображается на мониторе)

После успешного завершения процедур тестирования осуществляется поиск boot-устройств, содержащих в буфере данных байт **80h** (1000000), т.е. неисполняемую команду, используемую в данном контексте POST для подтверждения активности устройства загрузки OS.

Приоритет загрузочных устройств определяется в SETUP CMOS.

При выполнении POST могут генерироваться сообщения об ошибках.

Описания кодов ошибок можно найти по адресу: www.earthweb.com

или http://burks.bton.ac.uk/burks/pcinfo/hardware/bios_sq/bios_sq.htm

Доступ к переменным и константам BIOS.

Осуществляется через меню **SETUP BIOS**, доступное в начальные моменты загрузки BIOS с помощью клавиш, указанных в сообщениях.

Меню **SETUP** состоит из основных разделов:

- 1. Стандартные переменные: дата, время, параметры ОЗУ и накопителей.**
- 2. Дополнительные переменные и установленные модули программного обеспечения BIOS для тестирования аппаратных средств, устройств ядра ПЭВМ и ближайшего окружения.**
- 3. Параметры остальных устройств, интегрированных в состав системной (материнской) платы ПЭВМ и варианты распределения ресурсов (слотов расширений шин, прерываний и каналов прямого доступа).**
Здесь же определяется порядок опроса периферийных устройств, которые могут содержать загрузочные модули операционных систем.
- 4. Параметры интерфейсов устройств, подключаемых к системной плате, определяющих варианты энергосбережения.**
- 5. Размеры констант, определяющих параметры центрального процессора (частота, напряжение энергопитания ядра центрального процессора и ОЗУ), а также их предельные величины для сигнализации или отключения.**
- 6. Набор параметров BIOS, загружаемый по умолчанию (в случае ошибок ручного набора параметров пользователем).**
- 7. Ввод пароля для входа в редактор переменных – SETUP BIOS.**
- 8. Ввод пароля для продолжения загрузки BIOS и выполнения POST-процедур после включения ПЭВМ (пользовательский пароль).**

Значение паролей может быть снято системным обнулением BIOS.

Карта ввода / вывода.

0000 – 00FF – 256 8-разрядных портов предназначены для устройств, расположенных на системной (материнской) плате ПЭВМ,

0100 – 03FF – 768 8-разрядных портов отведены для контроллеров периферийных устройств, подключаемых к шинам системной платы ПЭВМ.

Внутренние устройства		Периферийные устройства	
Диапазон	Наименование устройства	Диапазон	Наименование устройства
000 - 01F	Контроллер ПДП №1	3B0 – 3DF	VGA
020 – 03F	Контроллер ПКП №1	378 – 37B	LPT 1
040 – 05F	Интервальный таймер	37C – 37F	LPT 2
060 – 06F	Контроллер клавиатуры	278 – 27B	LPT 3
070 – 07F	Часы CMOS и константы	3F0 – 3F7	FDD
080	Диагностический регистр	3F8 – 3FF	COM №1
081 – 08F	Контроллер страниц ПДП	2F8 – 2FF	COM №2
0A0 – 0BF	Контроллер ПКП №2	3E8 – 3EF	COM №3
0C0 – 0DF	Контроллер ПДП №2	2E8 – 2EF	COM №4

F000 – FFFF – 4096 дополнительных 8-разрядных портов, отведенных для различных виртуальных устройств, подключаемых к внешним шинам (USB, mini USB, SCSI, eSATA, HDD IDE ATA/ATAPI, PCI Express и т.д.).

Функции BIOS при работе с портами

Для прямого обращения ЦПУ к портам ввода/вывода в системе команд предусмотрено всего 2 команды: **IN** и **OUT**.

Но сам процесс обмена данными с периферийными устройствами сложен и должен учитывать ширину порта (порядок опроса нескольких 8-ми разрядных портов) и протокол обмена, определяющий порядок оценки готовности этих устройств к обмену данными по той или иной шине, а также свойства самой шины.

По этой причине в OS Windows прямой доступ к портам ограничен. Ввод прямых обращений к портам в приложениях может вызвать зависание программы или ее аварийное завершение из-за превышения привилегий.

Для этой цели в составе ядра OS имеется большой набор типовых обработчиков – функций BIOS, связанных с тем или иным общепользовательским прерыванием.

Поскольку количество прерываний ограничено, то обычно прерывание имеет несколько функций, а отдельные функции и подфункции BIOS:

функции прерывания **10h** – работа с портами видеоконтроллера VGA

функции прерывания **13h** – работа с дисковыми системами и DMA

функции прерывания **14h** – работа с последовательным портом COM

функции прерывания **15h** – работа с манипулятором мышь

функции прерывания **16h** – работа с клавиатурой

функции прерывания **17h** – работа с параллельным портом LPT

функции прерывания **1Ah** – работа с таймером, времязадающие функции

функции прерывания **80h** – работа со звуковым процессором

30. Система Plug & Play автоопределения устройств ПЭВМ.

Основные принципы построения системы PnP были сформулированы и частично внедрены в 1974-м году для шины MCA (Micro Channel Architecture).

Основные принципы построения системы PnP:

- 1. Ресурсы ядра ПЭВМ (порты доступа и их разрядность, номера прерываний, адресное пространство ОЗУ для обмена информацией, каналы прямого доступа) не являются жестко распределенными, а присваиваются по требованию.**
- 2. Каждое периферийное (по отношению к ядру системы) устройство имеет описание набора требований в своем BIOS.**
- 3. В составе BIOS PnP имеется программа – системный конфигуратор, которая присваивает номера периферийным устройствам, составляет паспорта (описания) этих устройств и выделяет необходимые ресурсы, с учетом недопущения конфликтов, при необходимости производит оптимизацию (перераспределение) ресурсов.**

Паспорта устройств сохраняются в реестре.
- 4. После загрузки операционной системы для периферийных устройств загружаются соответствующие системные драйвера. Осуществляется повторная проверка безконфликтной работы устройств.**
- 5. При отключении периферийного устройства или подключения нового операционная система автоматически перераспределяет освободившиеся ресурсы, определяет параметры нового устройства, проверяет его и предоставляет необходимые ресурсы без перезагрузки OS (на лету).**

Современная система PnP состоит из следующих компонентов:

1. BIOS стандарта PnP.

Уведомления – сообщение пользователю об обнаружении нового устройства

Конфигурирование – изоляция устройства до присвоения ID.

Поддержка данных- информация завершения POST на специальной RAM.

2. Система драйверов-энумераторов шин.

Контроллер шины получает информацию из RAM об устройстве или из реестра для устройств «не PnP» и присваивает уникальный номер.

3. Дерево аппаратных средств и реестр.

Ветвь в реестре OS под названием «HKEY_LOCAL_MACHINE\HARDWARE», которая состоит из типов аппаратных устройств.

4. Windows 95 и выше (или другая OS PnP).

Фирма Intel предлагает спецификацию PnP всем разработчикам OS.

5. Драйверы устройств PnP.

Спецификация PnP предполагает не только наличие доступной для BIOS информации об устройстве в RAM этого устройства, но и динамически подгружаемый драйвер этого устройства. Существует интерфейс прикладного программирования (API) для создания таких драйверов для новых устройств стандарта PnP. Загрузка таких драйверов должна регистрироваться диспетчером конфигурации и отвечать за выделенные ресурсы (сдавать их при выгрузке).

Подсистема ввода/вывода ПЭВМ и ядро OS решает следующие задачи:

1. Реализация вычислительной системы переменной конфигурации.
2. Параллельная работа программ в памяти и процедур ввода/вывода.
3. Упрощение процедур ввода/вывода, обеспечения их программной независимости от конфигурации конкретного периферийного устройства.
4. Обеспечение автоматического распознавания ядром ЭВМ периферийных устройств, многообразия их состояний (готовности, отсутствия носителя, ошибок чтения/записи и т.д.).
5. Интеллектуализация интерфейса, налаживание диалога между ядром и периферийными устройствами.
6. Переносимость и независимость OS от аппаратной платформы и ядра ПЭВМ.

Пути решения этих задач:

1. **Модульность** - новые периферийные устройства не вызывают существенных изменений архитектуры и вписываются в существующее адресное пространство, каналы и порты доступа.
2. **Унификация по формату передаваемых данных и команд** вне зависимости от используемых внутренних машинных языков микроопераций.
3. **Унифицированный интерфейс** по разрядности шины, набору линий сигналов управления и протоколам обмена.
4. **Унифицированные по адресному пространству**, доступному ядру ПЭВМ, и каналам доступа к нему со стороны центрального процессора для операций ввода/вывода информации в пределах этого адресного пространства,.

6. Арбитр ресурсов (служба OS PnP).

Основные функции:

- Обновление реестра, помещая туда новейшую информацию о выделении ресурсов на стадии загрузки,
- Переназначение ресурсов «на лету» любым устройства PnP, конфигурация которых изменилась.

Арбитр ресурсов работает в контакте с диспетчером конфигурации, который в любой момент может запросить у арбитра ресурсов освобождения ресурса с последующим предоставлением его другому устройству.

7. Диспетчер конфигурации (служба OS PnP).

Отвечает за процесс конфигурирования всей системы в целом.

Диспетчер конфигурации непосредственно взаимодействует, как с BIOS, так и с реестром, координируя процесс конфигурирования в ходе событий:

- когда BIOS отправляет ему список устройств «не PnP» на системной плате при загрузке, которые имеют жестко закрепленные за ними ресурсы,
- когда он получает извещение об изменении конфигурации от BIOS или от эnumераторов шин, которую он использует для идентификации всех устройств на конкретной шине, а также требования каждого устройства о выделении ресурсов. Эта информация заносится в реестр.

8. Пользовательский интерфейс (API).

Основное требование для пользовательских приложений, запускаемых в OS PnP – они не должны иметь явных обращений к ресурсам устройств (портам ввода/вывода, прерываниям или дискам) необходимо заменять эти обращения формальными обращениями к соответствующим устройствам.

Распределение ресурсов ПЭВМ между устройствами.

Для реализации принципов построения системы PnP, учитывая, что в составе архитектуры ПЭВМ встречается достаточно много устаревших устройств с жестко закрепленными за ними системными ресурсами, которые изменять нецелесообразно (контроллер клавиатуры, системный интервальный таймер, контроллеры ПДП и т.д.), в реальной системе PnP используется следующий порядок распределения ресурсов:

1. При проверке POST определяются устройства «не PnP».
2. Устройствам «не PnP» ресурсы выделяются в первую очередь согласно спецификационным требованиям, т.к. эти устройства неперенастраиваемые.
3. При обнаружении конфликтов BIOS PnP генерирует уведомление о необходимости устранения конфликтов вручную.
4. Затем осуществляется итерационное конфигурирование устройств PnP.
5. Используются методы изоляции устройств друг от друга (присваивается идентификатор и серийный номер), после этого устройству присваивается дескриптор (Handle).

Присвоение идентификатора связано с используемой устройством шиной и осуществляется специальной программой из состава OS – эnumератором шины, которая является новым типом драйвера контроллера шины. Номера идентификаторов являются уникальными для каждого устройства и неизменными для каждой последующей перезагрузки OS, например, **PnP 0000** – контроллер прерываний АТ, **PnP 0100** – системный интервальный таймер, **PnP 0C04** – матсопроцессор, **PnP 0A03** – контроллер шины PCI и т.д.

PnP- спецификация архитектуры аппаратных средств ПЭВМ, используемая соответствующими операционными системами для их конфигурирования и исключения конфликтов устройств между собой.

Основной компонент – все оборудование, подключаемое к шинам, содержит энергонезависимые регистры POS (Programmable Option Select), где хранится конфигурация устройства и требуемые ресурсы.

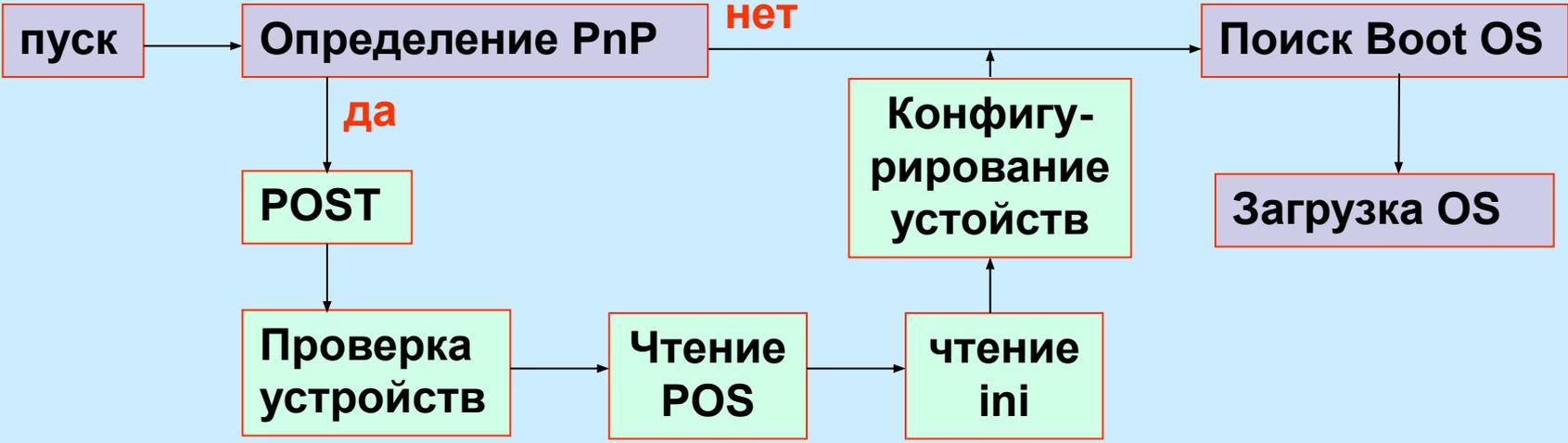
Дополнительный компонент – файлы OS описания устройств, драйверов к ним и требуемых ресурсов (ini –файлы или реестр OS).

Программы BIOS PnP, бесконфликтно распределяющие ресурсы.

Обычная загрузка системы:



Загрузка BIOS PnP:



31. Общие сведения об операционных системах.

Общие понятия об операционной системе.

Чтобы полностью овладеть всеми возможностями своего компьютера, необходимо знать и понимать его операционную систему.

Назначение операционной системы заключается в обеспечении удобства управления компьютером.

Любая операционная система, в полном смысле этого термина, является первой и наиболее важной программой любого компьютера. Как правило, она является и наиболее сложной, используемой только для управления самим компьютером.

Основная часть работы операционной системы заключается в выполнении огромного количества рутинных операций контроля, проверки достоверности, вычисления значений физических адресов и т.д. и т.п. и предназначена, чтобы скрыть от пользователей большое количество сложных и ненужных им деталей процесса управления аппаратной частью.

Как правило, операционная система состоит из нескольких частей:

Первая часть - это система BIOS в ПЗУ ПЭВМ.

Вторая часть - главная загрузочная запись.

Третья часть – аппаратный загрузчик операционной системы.

Четвертая часть – сканер и конфигуратор аппаратных средств ПЭВМ.

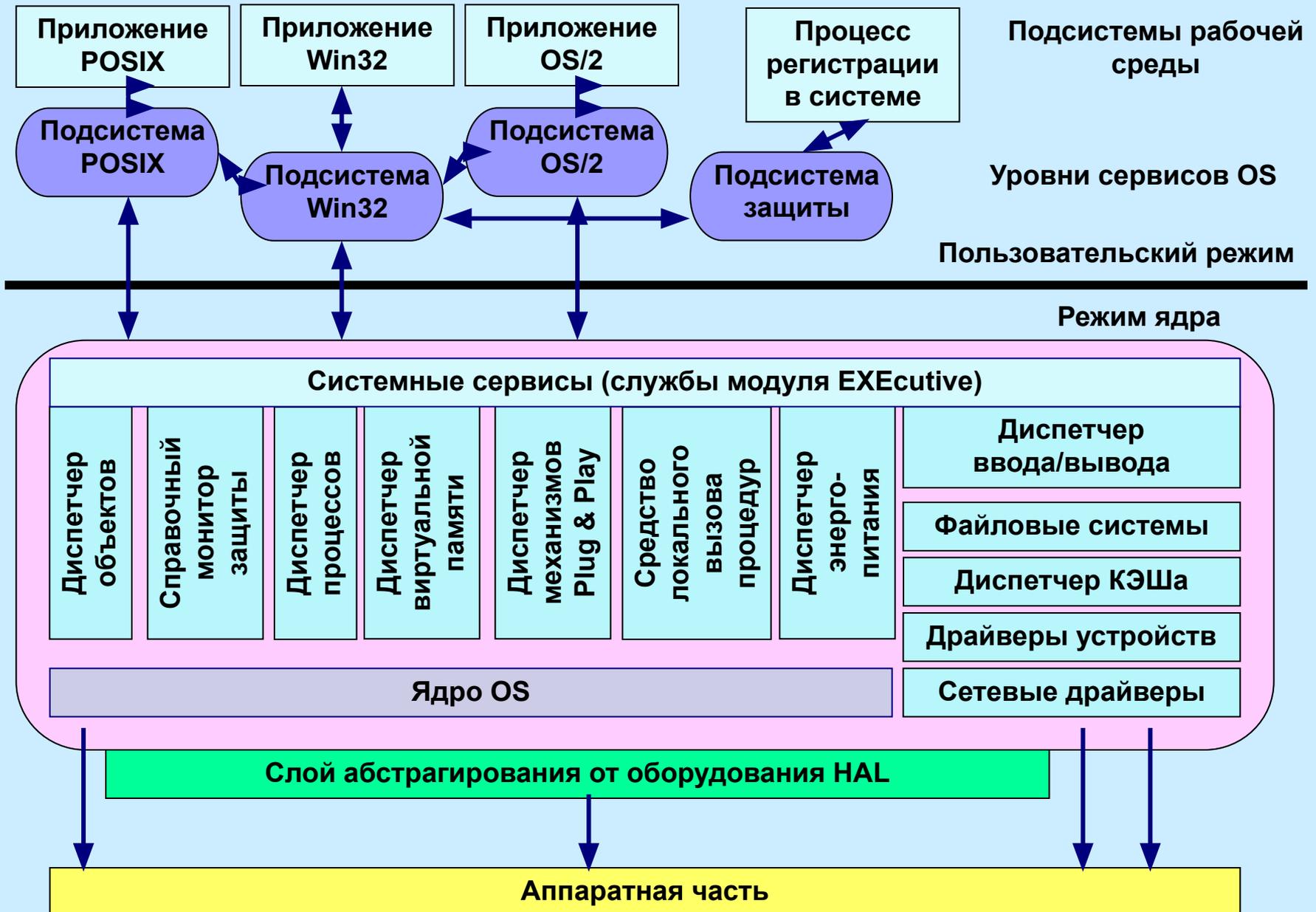
Пятая часть – ядро операционной системы и командный монитор.

Шестая часть – файлы конфигурации или реестр OS.

Седьмая часть – Диспетчеры объектов и устройств.

Восьмая часть – драйверы устройств.

Архитектура операционной системы Windows NT/2000/XP.



Пояснения к архитектуре операционной системы.

Пользовательский режим.

Большая часть приложений, запускаемых пользователем, работает в пользовательском режиме. Все эти приложения обладают ограниченным доступом к операционной системе, благодаря чему при возникновении неполадок в программе приложения ядро ОС остается надежно защищенным и продолжает нормально функционировать.

Пользовательское приложение работает в рамках изолированного адресного пространства, предоставляемого операционной системой.

При обращении к аппаратным устройствам (принтеру) службами ядра (диспетчером ввода/вывода) запускается соответствующий драйвер.

Службы Windows (например, Task Scheduler, Messenger, Alerter и др.) работают в пользовательском режиме в специальном пользовательском контексте безопасности индивидуального рабочего окружения каждой конкретной прикладной задачи.

Режим ядра.

Процессы, работающие в этом режиме, обладают наивысшим уровнем привилегий. Они работают в одном и том же адресном пространстве и могут напрямую обращаться к оборудованию компьютера, включая такие важные устройства, как центральный процессор или видеоадаптер.

В этом же режиме функционируют драйверы устройств, все системные диспетчеры, модуль Microkernel, все службы модуля EXEcutive, а также системные сервисы уровня абстракции оборудования HAL (Hardware Abstraction Layer).

Модуль EXEcuteve.

Так обозначаются в Windows NT/2000/XP программные компоненты, работающие в режиме ядра.

Эти компоненты включают в себя жизненно-важные службы ОС такие, как управление памятью, вводом/выводом, системой безопасности, механизмами взаимодействия процессов, кэшированием, а также системой управления объектами.

Модуль EXEcutive загружается в процессе начальной загрузки ОС и является частью файла Ntoskrnl.exe.

Модуль Microkernel.

Управляет переключением процессора между выполнением разных потоков, а также обрабатывает системные прерывания и исключения.

Этот модуль синхронизирует работу нескольких процессоров на многопроцессорных аппаратных платформах.

В отличие от остального кода ОС этот модуль никогда не перемещается в виртуальную память, т.к. его компоненты должны иметь фиксированные физические адреса.

Модуль Microkernel также является частью файла Ntoskrnl.exe.

Уровень абстракции оборудования HAL (Hardware Abstraction Layer).

Позволяет ОС работать на аппаратных платформах различной конфигурации и количества процессоров, не требуя ее перекомпилирования.

Обычно HAL разрабатывается производителем аппаратной платформы.

Модуль HAL находится в файле hal.dll и загружается в процессе начальной загрузки ОС.

Процессы и потоки.

Процесс – это программа, которую можно запустить в рабочей среде ОС.

Каждый процесс обладает своим адресным пространством, одним или несколькими программными потоками, а также идентификатором безопасности SID (Security ID), соответствующим учетной записи в контексте безопасности которой этот процесс функционирует.

Поток – это составляющая часть процесса (что-то вроде процесса внутри процесса), т.е. программный код, выполнением которого занят процессор.

В любой момент времени один процессор может выполнять только один программный поток. Переключение процессора между выполнением нескольких программных потоков осуществляется средствами ОС.

Каждый процесс может включать в себя несколько программных потоков. Например, работая в программе Explorer, можно открыть новое окно, для управления этим окном создается новый поток. Это означает, что в системе существует только один экземпляр процесса Windows Explorer, но в рамках этого процесса работает одновременно несколько программных потоков.

Симметричная многопроцессорная архитектура и наращиваемость ОС.

ОС Windows NT/2000/XP обладают встроенной поддержкой многопроцессорных систем SMP (Symmetric Multiprocessing). Это означает, что если в системе установлено больше одного процессора, то поток, требующий выполнения, будет выполняться первым освободившимся процессором. Переключение процессоров между потоками осуществит модуль Microkernel.

Службы и приложения пользовательского режима.

В пользовательском режиме работают процессы 3-х типов с различным уровнем привилегий.

1. Системные процессы. Это процессы управляющие рабочей средой пользовательского режима: Winlogon (подключение пользователей к системе), Service Controller (сервис подключения служб), Session Manager (диспетчер сеанса). Эти процессы запускаются модулем Ntoskrnl.exe в процессе начальной загрузки и используют учетную запись LocalSystem.

2. Службы Windows. Службы (Alerter, Computer Browser и др.) запускаются процессом Service Controller (файл services.exe) для прикладной задачи. Эти службы функционируют, как отдельные потоки в рамках процесса services.exe и не отображаются диспетчером задач в качестве отдельных процессов.

Большинство служб не использует контекст безопасности LocalSystem.

3. Пользовательские прикладные программы. Т.е. прикладные программы запускаемые пользователем в пользовательском режиме. Каждая такая программа функционирует в рамках своего собственного виртуального адресного пространства.

Процесс ассоциируется с подсистемой рабочего окружения environment subsystem, который принимает от него вызовы, адресуемые API (Application Program Interface) и преобразует их в команды, адресованные модулю EXEcutive.

Для поддержки DOS служит подсистема времени исполнения Csrss.exe, которая запускается автоматически при формировании вызова.

Порядок загрузки операционной системы.

Загрузка любой операционной системы после завершения операций инициализации аппаратных средств предусматривает следующие этапы:

- инициализация загрузчика OS (Boot loader process),
- выбор операционной системы (если предусмотрен выбор),
- повторное сканирование аппаратных средств,
- загрузка ядра OS и его инициализация.

После завершения POST BIOS передает управление на первое внешнее устройство (согласно установленному приоритету), имеющее в буфере данных байт 80h. Считывание следующего байта из этого порта активирует программу аппаратной загрузки OS: т.е. будет выполнена процедура, записанная в начале раздела MBR, которая позволит найти адрес первого активного раздела в таблице разделов (по значению байта 80h), и загрузит следующую за ним команду в табличной записи **JMP(xx)** на исполнение.

Адрес команды **JMP** является адресом аппаратного загрузчика **NTLDR** (для операционной системы Windows NT/2000/XP), **IBMBIO.COM** (для DOS) или аналогичной программы для других OS (например UNIX, Linux и т.д.).

NTLDR загрузит режим плоского 32-хразрядного адресного пространства и запустит минифайловую систему, совместимую с FAT16, FAT32 и NTFS. Затем читает **Boot.ini** в корневом каталоге и предлагает выбор OS к загрузке.

После выбора Windows XP выполняет программу **Ntdetect.com**, чтобы собрать информацию о всех физических устройствах, подключенных к ЦП.

Затем NTLDR загружает в ОЗУ и запускает ядро операционной системы – программу **Ntoskrnl.exe**, которая принимает данные сканирования.

Файлы, необходимые для успешного запуска ОС Windows XP.

Процедура запуска системы закончится неудачей, если хотя бы один из указанных ниже файлов не будет найден или окажется поврежденным.

Наименование файла	Расположение файла
NTLDR	Корневой каталог Windows
Boot.ini	Корневой каталог Windows
Bootsect.dos (для выбора ОС)	Корневой каталог Windows
Ntdetect.com	Корневой каталог Windows
Ntbootdd.sys (только для SCSI)	Корневой каталог Windows
Ntoskrnl.exe	%SystemRoot%\System32
Hal.dll	%SystemRoot%\System32
Улей реестра \system	%SystemRoot%\System32\Config
драйверы устройств	%SystemRoot%\System32\Drivers

Файл **Bootsect.dos** содержит копию первого сектора раздела альтернативной ОС (в данном случае DOS), аналогично можно построить загрузчики для ОС UNIX и Linux, а затем отредактировать файл **Boot.ini**:

```
C:\BOOTSECT.UNX="UNIX"
```

```
C:\BOOTSECT.LNX="Linux"
```

Дальнейшее развитие архитектуры операционных систем.

Загрузочная запись ДОС версии 1.00 имеет одно незначительное отличие от всех остальных версий: вместо имени Роберта О'Рира – разработчика этой версии ДОС, которую он сделал по прототипу (Key DOS) за 2 месяца, во всех последующих версиях стоит название фирмы - "Майкрософт".

В настоящее время разработка новой версии ОС занимает 6-10 лет и в ее реализации участвует коллектив 200-300 человек.

Фирма Microsoft стала практически монополистом на рынке ОС ПЭВМ. Из альтернативных ОС для ПЭВМ можно указать лишь Unix и Linux.

Основное совершенствование, которое реализовано в настоящее время – для многоядерных процессоров нового поколения архитектуры Itanium создан новый интерфейс между ОС и встроенным программным обеспечением аппаратных платформ – **EFI (Extensible Firmware Interface)**, который предназначен для замены системы POST.

Для этой системы фирмой Microsoft разработан новый загрузчик для операционной системы Windows XP, вошедший и в состав ОС Windows 7, **IA64ldr.efi** вместо **NTLDR**, файлы **Boot.ini** и **Ntdetect.com** также исключены из системы, т.к. информация, которая хранилась там, находится в **NVRAM**, а все аппаратное окружение полностью соответствует спецификации ACPI.

Кроме этого для поддержки операций с плавающей точкой разработан новый объект **FPSWA.efi**, который иницируется при загрузке ОС.

В состав ОС введен новый каталог **EFI**, содержащий инструментальные средства спецификации EFI.

Соответствующим образом откорректирован реестр Windows.