

Введение в операционные системы

Операционные системы М.П. Чубрикова.

Опер. С. - обеспечивает управление всеми аппаратными ресурсами машины и позволяет отделить системные классы программ от непосредственного общения с аппаратурой (кроме отдельных специализированных случаев).

ОС обычно стандартизированы и число их не велико - не более десятка.

Отечественные: В простых машинах или системах функцион. ОС обычно выполняет следующие функции: управление клавиатурой, дисководом, монитором или видеокамерой (ПЗУ) интерпретатор языка Basic (как в Sinclair это преимущество неудобно т.к. все всегда скрыто в теле) хотя кто кроме BASIC ничего не знает и пишет лишь в шрифт - это приложение но машина теряет б.в. лучшее качество отформатированной архитектуры.)

На больших машинах обычно устанавливаются самодостаточные стандартные широко распространенные системы ориентированные на бизнес-тип микропроцессоров. Наиболее распространенными являются CP/M (Control Program for Microcomputer), MS-DOS, UNIX.

На высокопроизводительных машинах обычно устанавливаются ОС и ее разновидности. Для этой ОС создано мощное ПО транслятора почти всех языков высокого уровня, текстовые редакторы, СУБД и т.д. Распространению системной оболочки способствуют ее простота и мобильность. Несмотря на малое количество функций и составных частей - в след. лекции.

Михайлов

Несколько слов о системах с закрытой и открытой архитектурой.

Предусмотреть все варианты использования МПС не возможно поэтому для того чтобы использовать /какие-то еще/ модули аппаратного расширения, необходимо их перед-то подключить, для этого все основное изделие "узнают" и подсоединяют в цепь расширения к которой подключаются разъемы (слоты) подключаемые наращивать систему.

Такие системы открытые. Обычно в них предусматривается и возможность за- /соединя/ действий программ через карту прерываний или окно ввода п.п.

Эти системы - открытого типа. Существуют хорошие и популярные системы, не обладающие такой возможностью напр. некоторые компьютеры. Apple, Macintosh, Syntair.

Такие системы обладают закрытой архитектурой (картмодули и аппаратура расширения не к цепи).

Лекция №2 Операционная система (ОС)

Задачи ОС:

- поддержка работы всех программ, обеспечивающих взаимодействие с аппаратурой;
- предоставление пользователю возможностей общего управления машиной.

В рамках 1-й задачи взаимодействие программ с Вн. Уст. и друг с другом, распределение оперативной памяти, выявление различных состояний системы и реакция на них (выбросы и т.д.)

- 2-я задача - управление осуществляется на основе командного языка (языка директив) с помощью которых реализуются: расстановка дисков, копирование программ, растекание каталогов, загрузка, печать, принтер, установка режимов работы и т.д.

Составные части ОС. В полной конфигурации ОС для ПЭВМ профессионального компьютера должна содержать следующие основные компоненты:

- файловую систему;
- драйверы внешних устройств;
- интерпретатор командного языка.

Файловая система: - хранилище программ и данных. Файл - именованная область памяти - место постоянного хранения информации программы, данных их работ, текстов, закодированных изображений.

Файлы реализуются как участки памяти на носителях информации: шлейфы, магниты, виртуальной памяти и т.д.

Файл имеет имя зарегистрированное в каталоге — описании файлов.

Каталог (директория) доступен пользователю через командный язык ОС. - его идентифицируют, переименовывают в нём файлы, переносят их с места на место, удаляют. Каталог имеет собственное имя и хранится в другом каталоге наряду с файлами (тогда он называется каталогом нулевого уровня).

Так образуются иерархические (от старшинства) файловые структуры (они же древовидные и разветвленные). Файловой системе имеют доступ к ресурсам периферийных устройств на данной ЭВМ, при помощи функциональных процедур. Пользователь может быть обеспечен на любой системе или потребителем информации в машине, напр. принтер, дисплей, клавиатура и т.д.

Управляемые внешними устройствами: Стандартные ВУ это дисплей, клавиатура, гибкие диски, жесткие диски, принтер. Дополнительно возможно подключение следующих устройств: графопостроители, планшет, сканер, джойстик типа "мышь", джойстик, модем для связи по телефонным кабелям, контроллеры локальных сетей, АЦП, ЦАП и т.д.

Каждое ВУ характеризуется определенным протоколом обмена, пропускной способностью и структурой передаваемых (принимаемых) данных;

Поддержка широкого набора ВУ - одна из важнейших функций DOS. Для ее осуществления введено понятие драйвера - программа специального типа, ориентированная на управление внешним устройством. Каждому устройству образуют свой драйвер. Драйверы стандартных ввода-вывода (BIOS) которая обычно заносится в ПЗУ системного блока ПЭВМ. Драйверы можно добавлять дополнительно в процессе работы.

Командный язык - обеспечивает взаимодействие пользователя с операционной системой, позволяет оперативно при помощи директив (просто вводимых команд) осуществлять обращение к различным внешним носителям, запуск программ и т.д.

Анализ и исполнение команд (директив) пользователя всегда выполняется готовых программ из файлов в оперативную память ПЭВМ и их запуск осуществляется

Командными процессорами ОС. - на этой стадии ОС несет поддержку взаимодействия с пользователем.

Одним из команд можно считать меню, которое является составной частью сложной программы, которая к примеру позволяет оперативно создавать удобную для пользователя обстановку (сразу войти редактор, BHSIK и т.д.). Не путать с графическими оболочками - которые являются удобными визуальными средствами над ОС.

Граф. оболочки типа MS-DOS, Windows и т.д. не являются составной частью ОС. В рамках ОС могут существовать графические оболочки терминалов, которые позволяют пользователю работать и передавать в ЭВМ информацию, обычно более мощную, через канал связи. В этом случае машина непосредственного пользователя выполняет роль терминала. В этом случае основная задача программистов-системщиков перенести файлы между машинами.

ОС. семейство СРМ.

СРМ разработана в 1974г. и явилась началом и прототипом для многих последующих ОС. Характерна для 8-разрядных машин. Для СРМ разработано значительное ПО трансляторы логики всех видов высокоуровневого языка, процессоры, БД, графические пакеты, оладки и т.д.

Система довольно проста и компактна легко переносится на отдельные типы МЭВМ, первая версия занимала 4К. памяти, что было вынуждено в условиях ограниченных ресурсов памяти.

В дальнейшем, после реконструкции системы, её структура была усовершенствована с помощью внешнего текстового файла, описывающего конфигурацию внеш. устр-в. Т.о. система стала более гибкой и конкретное включение конфигурации ИП. системы.

СРМ полностью хранится на диске и состоит из трёх системных частей: базовой системы ввода-вывода, базовой дисковой операционной системы, координатного процессора. Эти части содержатся в отдельных файлах с именами BIOS, BDOS, CCP. По включению машины указанные файлы загружаются в оперативную память по указанным в них адресам, остаются в памяти до завершения работы программы, загружаемой с диска. Присущая для работы памяти величина — 46 Кбайт.

BIOS — базовая система ввода/вывода — содержит основные программы-драйверы внешних устройств. Это единственная аппаратно-зависимая часть СРМ, требует настройки на конкретное оборудование. Таблица описывающая характеристики и параметры дисков, клавиатуры и мыши, параметры периферийного канала, формируется при загрузке BIOS с использованием специального файла CONFIG.SRM. При установке нового устройства информация вносится в файл CONFIG.SRM, скопировать BIOS и систему снова готова к работе.

BDOS — содержит функции управления файловой системой и общего управления машиной (около 40). Файловая система СРМ довольно ограничена. Система всех файлов хранится в одном едином каталоге который можно разбить на несколько нумерованных пользовательских областей.

Процессор командных команд (CP) имеет средства для обработки лишь нескольких ветровых команд. Сюда относятся команды настройки на рабочий диск.

A>, B>, C>, ...

USER - выбор области пользователя;

DIR - вывод каталога;

TYPE - вывод содержимого файла на экран.

REN - переименование файла;

ERA - удаление файла

Основные команды:

Загружаемые команды или транзитные.

PIP - переименование содержимого файла

STAT - просмотр и изменение атрибутов файлов.

FORMAT - форматирование или начисление разметки и многие другие, до нескольких сотен.

Запуск программы:

A>[имя прог. ин], [параметры] ввод(ENTER)

система автоматически находит программу, ее загружает в память и передает управление. Возможно создание файлов из команд и их последов. выполнения. Система сообщает что сделало с файлами при помощи расширений и имен.

Имя (TXT, COM, EXE и т.д.)

После входа в CP/М одна из программ выполняется по номеру: умножением на 3 и суммированием с базовой адресом.

CP/М получила образом и основой для целого класса системных ОС как для 8- так и для 16-разрядных машин. Основой из них MS-DOS - на следующей лекции.

Лекция №2 Структура и функции DOS.

п.1. Основные модули DOS

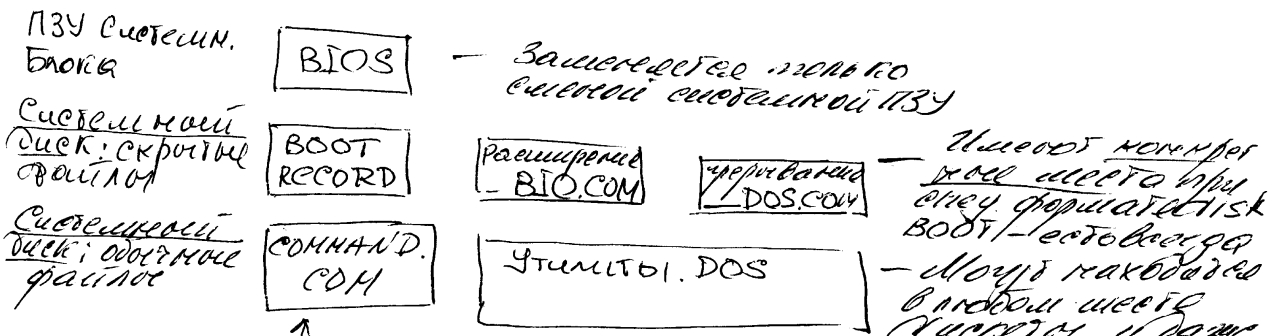
xxx Лекция не претендует на всю полноту описания DOS т.к. это объект большой книги, а является в некотором смысле ознакомительной и представляет DOS как пример из общего класса ОС для различных машин.

а) DOS - модульная система что позволяет можно индивидуализировать её расширять функции и т.д. путём замены одного или нескольких модулей, которые можно разделить грубо на:

- 1 - базовая система ввода/вывода (BIOS)
- 2 - блок начальной загрузки (BOOT RECORD (Sector 0))
- 3 - модуль расширения (BIOS) ... BIOS.COM (IO.SYS)
- 4 - модуль обработки прерываний (DOS.COM) (Rain.sys)
- 5 - командный процессор (COMMAND.COM)
- 6 - утилиты DOS (FORMAT.COM, CHKDSK.COM и др.)

(Начало сист. блоков обозначают группу разработчика Икогда (и это в нек-е смысле справедливо) к DOS относятся только модули (1, 2, 3, 4) т.к. BIOS - устанавливая разраб. boot-ка машины, а утилиты по сути можно отнести к отдельным приложениям.

Разделение модулей DOS:



Подобная структура DOS является гибкой и расширяемой. Рассмотрим функции отдельных частей.

п.2. Функции BIOS

BIOS можно рассматривать как составную часть аппаратных средств (сист. блок), её задачи:

а) Автоматизирование основных аппаратных возможностей при включении (ОЗУ, Порт, клавиатура...) при наличии ошибки выводит код на экран или если это не возможно звуковой сигналом. Работа машины прекращается по завершении проверки и перезапуска.

б) Возврат блока начальной загрузки DOS после завершения тестирования. Процесс идёт в 2 этапа сначала ищет BOOT-сектор, начиная с диска А и т.д. если безуспешно, если утешительно сформатированный диск не найден, система и в дальнейшем будет работать на нем.

BOOT-Record получив управление после загрузки операционной системы, загрузку и передачу управления DOS т.к. «он знает это».

Если попытка найти BOOT-сектор системы не удалась, то передается управление ветромикому BASIC из ПЗУ. Если он отсутствует, то машина просит вставить системную дискету, и осуществляется повторный запуск.

В) Обслуживание системных вызовов или прерываний. Системные вызовы генерируются программами или аппаратными устройствами в целях выполнения необходимых операций.

Суть механизма прерываний: текущая работа микропроцессора приостанавливается одним из сигналов, который указывает на возникновение события, требующего немедленной обработки, если конечно этот сигнал ранее не запрещен.

Прерывания делятся на 3 группы: аппаратное, исключение и программное.

а) Аппаратные прерывания вызываются: падением напряжения питания, нажатие клавиши или импульс от датчиков, накопителей на жестких и гибких дисках...

б) Позиционные или прочисленные прерывания возникают при выполнении операций в работе CPU: деление на ноль, переполнение, признак останова и т.д.

в) Программные прерывания - вызываются программой или вызовом другой программы - (например, CPM, где подпрограммы связаны со взаимодействием аппаратными средствами).

Прерывание может иметь многократное вложение и можно запретить такую возможность.

Итого: в аппаратных и программных ресурсах могут выполняться прерывания, которые могут отсутствовать полностью. 31 (00-1F) прерывание низкого уровня обслуживает BIOS эти прерывания требуют непосредственного вмешательства с аппаратными частями и по сути являются простейшими подпрограммами по управлению дисками, клавиатурой, НЧД, МЧД, принтером и т.д. Прерывание BIOS не интеллектуальное, выдает байт на экран, на принтер, управляет головкой ЧД. Т.е. BIOS - своеобразная программная оболочка вокруг аппаратных средств позволяющая обращаться к аппаратным ресурсам через прерывание BIOS.

Следующие (20-3FH) прерывания являются прерываниями DOS и они более развитые: создание файлов, переименование, удаление, поиск по каталогам, и т.д.

п.3.1 Функции расширения BIOS.COM.

ПЗУ BIOS является неотъемлемой частью, характерной для данной машины, её изменение задано аппаратно и трудная т.к. ПЗУ ориентировано на конкретную машину. Если возникает необходимость расширения BIOS, то для BIOS нужны или дополнительные устройства не являющиеся стандартными то расширение шлобы расширением, расширяющей некоторую ф-цию BIOS и добавляющей стратегическую поддержку к существующей набору функциональных драйверов устройств например, мышь и т.д. Это позволяет максимально адаптировать систему под задачи и приходы конкретного пользователя.

Все информации передается DOS через специальный файл конфигурации, находящийся на дискетке, имеет - CONFIG.SYS - это текстовый файл, его BIOS и обрабатывает осуществляя настройку системы:

DEVICE = MOUSE.SYS - включить драйвер. мыши
BREAK = ON → шотко прерывать программу комбинацией CTRL + Break.
FILES = 12 (шотко открывающихся файлов (STAND = 8)
BUFFERS = 8 (8 буферов обмена, информации с TMD (STAND = 2)
SHELL = START.COM (замена командного процессора) и т.д.

BIOS.COM кроме того завершает загрузку системы передав начало управление уже загрузившейся DOS.COM, который установит свои вектора прерываний и подготовит все к загрузке COMMAND.COM. Далее BIOS.COM загружает COMMAND.COM с диска и передает ему управление.

Отступление о структуре памяти в 16-раз. системе.
Процессоры типа 80 (88) способны адресовать 1Мбайт ОЗУ, хотя выбор ячейки осуществляется через 16-бит адреса, а они разрешают только 64Кбайт. Для семнадцатой адресации используется специальный регистр CS - 16 раз., отсюда шотно заключить что 2¹⁶ - адресуют около 1 млрд. байт, но это не так. CS - это регистр сегмента и указывает адрес начала сегмента с интервалом 16 байт. За адресацию внутри сегмента размером 64К отвечает регистр IP. т.е. Сегмент + смещение, "Окно" как бы "скользит" по памяти которую процессор считает непрерывной цепью из 1 байта на байт. Кроме того существуют DS - рег. сегмента данных, SS - регистр сегмента стека, ES - регистр дополнительного сегмента. т.е. в отдельной шотности времени процессору непосредственно доступны 256К DS и SS - определяют допустимые области памяти и стека. Абсолютный адрес = CS × 16 + IP.

п.5 Командный процессор. COMMAND.COM

это бинарный файл, расположенный на диске обязательно и именуется по зарезервированному имени.

Основные функции:

- прием и разбор команд, полученных с клавиатуры или из командного файла (BAT)
 - выполнение ветровых команд DOS, находящихся в файле COMMAND.COM.
 - Загрузка и исполнение внешних команд DOS и прикладных программ в файлах COM и EXE. Процессор не отличает утилит от прикладных программ, поскольку по структуре они неразличимы.
 - Процессор инициализирует файл автозагрузки AUTOEXEC.BAT осуществляющий начальную настройку среды на запуск пользователя (если таковой есть).
- COMMAND.COM делится на 2 части: резидентную и нерезидентную. Последняя обрабатывает команды с клавиатуры или командных файлов, загружает внешние программы, назначает адрес и переход на исполнение. Эта часть загружается в старший адрес ОЗУ и может быть затерта прикладной программой, расположенной в адресов памяти.

Управление после прикладной программы передается в резидентную часть, которая пытается вернуть с диска свободный блок. Поэтому если свободен стартовый сектор - вступает загрузка внешней дискеты. На дискете это происходит незаметно. Параметры памяти при работе DOS можно представить так:

FFFF	
BIOS + ПЗУ BASIC	системная область ~ 384 К.
0000	
память дисплея и ЧВВ	
10000	
резидентная часть COMMAND.COM	~ 580 К
блок программы	
ОБЛАСТЬ ПРИКЛАДНЫХ ПРОГРАММ ПОЛЬЗОВАТЕЛЯ	область DOS
резидентная часть COMMAND.COM	
MSDOS.COM	
IBMBIO.COM	
переменные DOS	1024 К - вектора прерываний
переменные BIOS	
таблица векторов	

Утилиты DOS - это уже известные по CP-программистам утилиты: FORMAT.COM, CHKDSK.COM, PRINT.COM, EDIT.COM. Они не имеют отношения к системным программам. BASIC.COM, PASCAL.COM и другие языки программирования. T.O. DOS - структура системы и ее работа. Таблица векторов прерываний.

000000

Принципы построения многозадачных систем. Переключение задач. Разновидности многозадачности. Виртуальная память и эмуляция виртуальной машины. MS Windows, как пример ОС с параллельным выполнением процессов и вытесняющей многозадачностью.

**Принципы построения многозадачных систем.
Переключатель Задач**

Очень часто при работе в операционной системе DOS бывает очень удобно держать несколько задач одновременно в памяти и оперативно переключаться между ними. Например, иметь возможность нажатием одной клавиши переключаться, скажем, между средой разработки, текстовым редактором и программой-оболочкой. Такой процесс возможен, если вспомнить, что контекст любой задачи представляет собой содержимое регистров микропроцессора и оперативной памяти, используемой задачей. С прогрессом в разработке жестких дисков большого объема стало принципиально возможно оперативно сохранять контекст выполняемой задачи на "винчестер" и восстанавливать при необходимости. В дальнейшем этот механизм был интегрирован в структуру микропроцессора в качестве виртуальной памяти.

Впервые подобное решение было реализовано в операционных системах DR DOS 6.00 компании Digital Research и ее следующей версии, вышедшей под маркой Novell: Novell DOS 7.00.

Наряду с такими преимуществами, как защита от несанкционированного доступа к компьютеру, каталогам и файлам, утилиты настройки системных параметров, гипертекстовый справочник, в эти операционные системы включен Переключатель Задач.

В чем же его преимущество по сравнению с другими подобными системами?

- 1. Он позволяет переключать задачи непосредственно из командной строки DOS: {CTRL-1} — переключаемся на первую задачу; {CTRL-2} — на вторую и так далее.*
- 2. Список загруженных задач вызывается комбинацией клавиш {CTRL-ESC}. При этом для организации новой задачи достаточно нажать клавишу*

{INS}, а для удаления -{DEL}. При этом Переключатель Задач занимает всего 960 байт для каждой загруженной задачи и полностью использует преимущества расширенной памяти и, если она есть, переключает задачи мгновенно, если нет - за 4-5 секунд. При этом механизм устойчиво работает даже на XT.

- 3. Novell DOS 7 пошла еще дальше — Переключатель Задач может обеспечивать еще и Многозадачность — при этом несколько DOS-задач могут выполняться параллельно. Поскольку этот механизм встроен в ядро операционной системы все переключения выполняются быстро, четко и без сбоев.*

Обычно для работы хватает возможностей только Переключателя Задач. Многозадачность несколько снижает скорость работы программ, в то время как Переключатель нет, — в каждый момент времени выполняется только одна активная задача, а остальные ожидают своей очереди на диске или в расширенной памяти. При загруженном Переключателе Задач каждому приложению доступно 623 Kb оперативной памяти.

MS Windows, как пример ОС с параллельным выполнением процессов и вытесняющей многозадачностью.

Система Microsoft Windows (далее Windows) пользуется огромной популярностью и стала стандартом de facto для современных персональных компьютеров типа IBM PC. Это объясняется тем, что Windows предоставляет пользователю такие мощные средства, как единый графический интерфейс, многозадачность, обмен данными между приложениями, совместимость с ранее разработанным программным обеспечением, поддержку внешних устройств самых разнообразных типов и другие, отсутствующие в MS DOS средства.

С другой стороны, программист, создающий программу для Windows, может не заботиться о проработке вопросов совместимости с оборудованием различных типов, поскольку все эти проблемы решает за него Windows. Таким образом, программа, разработка которой выполнялась на компьютере, оборудованном адаптером VGA, будет точно так же работать и на компьютере с адаптером EGA или CGA. При этом в тексте её не будет ни одного упоминания типа адаптера дисплея.

Именно по этим причинам количество программного обеспечения для Windows увеличивается с каждым днем.

Основные отличия Windows от MS DOS

Основным отличием Windows от MS DOS является то, что Windows запускается на выполнение как обычная программа, а MS DOS — это, несмотря на все ее недостатки, операционная система, загружаемая в оперативную память при включении компьютера.

*С другой стороны, после того как Windows получает управление, она переключает на себя все функции DOS (за исключением файловой системы) по управлению аппаратными средствами. Поэтому Windows часто называют **операционной оболочкой** или **операционной средой**.*

В идеологии работы приложений Windows и MS DOS тоже имеются коренные различия. В MS DOS любая программа может работать не только с функциями операционной системы, но и напрямую обращаться к базовой системе ввода-вывода (BIOS) и даже непосредственно к ресурсам компьютера. Такая открытость MS DOS объясняется тем, что она создавалась для работы одного пользователя с одной программой. Бурное развитие персональных компьютеров и появление очень мощных приложений привели к возникновению противоречия между возможностями MS DOS и потребностями пользователей. Поэтому Windows создавалась для того, чтобы обеспечить единый подход к разработке программного обеспечения для различных аппаратных средств, единый пользовательский интерфейс, позволяющий работать на одном компьютере с несколькими задачами, сохранив при этом возможность применения всею программного обеспечения, созданного для MS DOS.

*Схема взаимодействия программы для Windows с ресурсами компьютера приведена на рис. 1. Как легко можно заметить, программа для Windows, в отличие от программы для **MS DOS**, непосредственно к ресурсам компьютера никогда не обращается. Вместо этого она вызывает системные функции Windows, которых насчитывается несколько сотен.*

*Все основные компоненты Windows сосредоточены в трех модулях **KERNEL.EXE**, **GDI.EXE** и **USER.EXE**.*

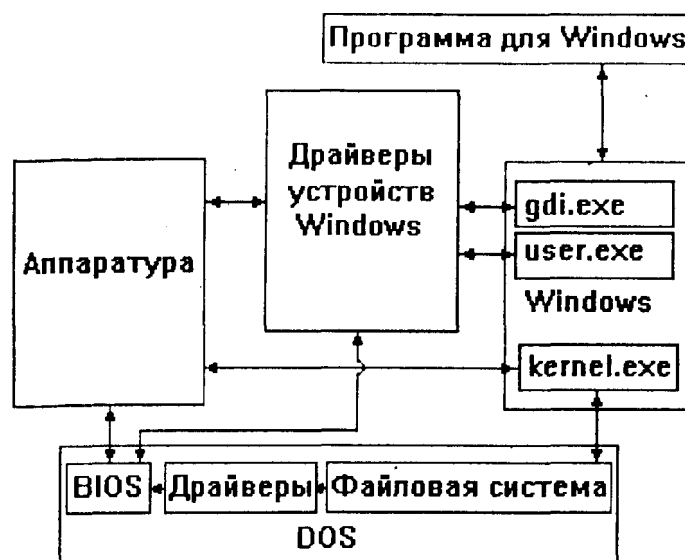


Рис.1. Взаимодействие Windows-программы с ресурсами компьютера.

Модуль *KERNEL.EXE* является ядром системы ("kernel" и означает "ядро") и отвечает за управление памятью, загрузку и выполнение программ и т. п. Модуль *GDI.EXE* (*Graphics Device Interface* — интерфейс с графическими устройствами), как видно из его названия, взаимодействует с различными графическими устройствами. Модуль *USER.EXE* отвечает за организацию взаимодействия пользователя с *Windows*-приложением через устройства ввода-вывода.

Функции, которые входят в состав этих модулей, составляют *Windows API* (*Applications Programming Interface* — интерфейс для создания приложений). При работе программы под *Windows* она должна обращаться к этим функциям для выполнения тех или иных действий.

Windows и объектно-ориентированное программирование

Среда *Windows* неразрывно связана с объектно-ориентированным подходом к программированию. Главным объектом *Windows*, что отражается даже в названии системы, является объект *Window* (окно). Любая программа *Windows* имеет хотя бы одно окно, называемое главным окном (*Win Main*), которое может владеть еще несколькими окнами.

С точки зрения пользователя, окно это участок экрана, в котором отображается работа текущей программы, а с точки зрения программиста, это объект, параметры и поведение которого определяются в соответствии с принципами объектно-ориентированного программирования.

*После того как создан экземпляр объекта типа окно, он получает доступ к **сообщениям системы**. Это значит, что все события, происходящие во время работы компьютера, — сигналы от таймера, нажатия клавиши мыши или клавиатуры, сообщения от различных устройств, а также от всех работающих в данный момент программ — обрабатываются Windows и помещаются в единомобразном виде в специальный блок, который и называется системной очередью сообщений. Модуль USER.EXE распределяет эти сообщения по очередям приложений.*

Программа для Windows периодически проверяет свою очередь сообщений, есть ли сообщения тех типов, на которые одно из его окон "умеет" реагировать.

Алгоритм работы программы для Windows должен полностью отражать идеологию объектно-ориентированного программирования, так как сообщения, которые, по сути, являются структурами данных, управляют взаимодействием объектов — окон приложения Windows.

Windows предоставляет большое количество разнообразных стандартных сообщений, которыми могут обмениваться программы. Кроме того, программист может создать свои сообщения, отличные от стандартных.

Таким образом, после того, как программа создает главное окно, она передает управление Windows и только периодически проверяет свою очередь сообщений. Если какое-нибудь из окон приложения может реагировать на очередное сообщение, то соответствующая функция окна вызывается через Windows функцией управления этим окном приложения.

Принцип разделения ресурсов компьютера при работе нескольких задач в Windows

MS DOS, как уже говорилось, изначально проектировалась как однозадачная система. Поэтому в ней практически нет средств, позволяющих одновременно выполнять несколько программ (кроме утилиты print). Поскольку Windows ориентирована на работу одновременно нескольких программ, то мо-

но полное владение ограниченными ресурсами, такими как устройства ввода-вывода, оперативная память и процессор, характерное для MS DOS, в данном случае недопустимо.

При создании программ для Windows нельзя напрямую обращаться к тем или иным устройствам. Это относится не только к дисплею или, скажем, к клавиатуре, но и к оперативной памяти, так как в любой момент времени диспетчер памяти Windows может вытеснить любой из блоков памяти, выделенных программе, на диск или переместить в другую область адресного пространства.

Поэтому если необходимо прикладной Windows-программе обратиться к какому-нибудь разделяемому ресурсу, нужно использовать функции Windows, а также специальную технику программирования для правильного взаимодействия создаваемой программы со средой Windows.

При соблюдении этих правил программа после инициализации будет находиться под управлением Windows, что позволяет использовать все функции среды по управлению задачами без их явного описания в самой программе.

Таким образом, разделение ресурсов, хотя и накладывает некоторые ограничения, предоставляет возможность программисту сосредоточиться на разработке собственно алгоритма программы, уделяя проработке интерфейса минимальное время.

Динамически подключаемые библиотеки (DLL) и принципы организации памяти Windows.

Если программа, предназначенная для работы в MS DOS, использует какие-либо библиотечные функции, то при компоновке выполняемого файла все эти функции помещаются в результирующий файл.

В Windows применяется принципиально иной метод, заключающийся в том, что в выполняемый модуль помещается не сама функция, а название библиотечного модуля и номер этой ФУНКЦИИ в данном модуле. Все вызовы этой функции в тексте программы преобразуются компилятором к виду дальних межсегментных вызовов (*far inter segment calls*) как будто бы данная функция находится в другом сегменте оперативной памяти.

При загрузке программы и обращении к какой-нибудь из библиотечных функций Windows подгружает эту библиотеку в память и подставляет нужный адрес в

команду вызова функции. Эта технология получила название **динамически подключаемых библиотек - DLL** (Dynamic Link Libraries). Например, при обращении в тексте программы к функции *Create Palette*, находящейся в модуле *GDI.EXE*, компоновщик поместит в выполняемый файл имя модуля *GDI.EXE* и число 360 — порядковый номер функции, которые не будут использоваться системой при вызове *Create Palette*.

Организация памяти в системе *Windows* включает глобальную область памяти, разделяемую всеми программами, и локальные динамические области, предназначенные для работы каждой отдельной программы *Windows*. В глобальную область памяти загружаются сегменты кода и данных программ. Эта область используется и в качестве динамической памяти. Сегменты, которые находятся в младших адресах глобальной области памяти, не могут перемещаться в другие части адресного пространства или вытесняться на диск. Такие сегменты называются фиксированными. В старших областях размещаются сегменты, которые могут и перемещаться, и вытесняться на диск. Сегменты, которые не могут вытесняться на диск, но допускают перемещение в памяти, размещаются между фиксированными и вытесняемыми сегментами.

При возникновении запроса *Windows* на размещение фиксированного сегмента система с младших адресов начинает поиск непрерывной свободной области памяти нужного размера. Если такая область не найдена, *Windows* выполняет перемещение сегментов в старшие адреса памяти. Когда и эта мера не позволяет выделить участок памяти нужного размера, *Windows* вытесняет на диск те сегменты, к которым было меньше всего обращений.

Для размещения в памяти перемещаемых, но невытесняемых сегментов *Windows* просматривает память, начинающуюся выше области фиксированных сегментов, перемещая или вытесняя сегменты на диск при необходимости. В области вытесняемых сегментов *Windows* всегда держит свободное непрерывное пространство, достаточное для размещения сегмента наибольшего размера. При работе с локальной памятью используются те же принципы и алгоритмы, что и при работе с глобальной.