Конспект лекции 1

История ОС

ОПЕРАЦИОННАЯ СИСТЕМА (ОС) —

1) система оперативного получения, фиксации и обработки учетно-бухгалтерской информации о деятельности компании и ее подразделений, функционирующая как на документальной основе (получение и обработка документов), так и на основе сообщений, уведомлений, передаваемых через средства связи;

2) программное обеспечение, осуществляющее управление выполнением компьютерных программ, распределением памяти, организацией данных

Операционная система (ОС) управляет компьютером, запускает программы, обеспечивает защиту данных, выполняет различные сервисные функции по запросам пользователя и программ.

Каждая программа пользуется услугами ОС, а потому может работать только под управлением той ОС, которая обеспечивает для нее услуги. Таким образом, выбор ОС очень важен, так как он определяет, с какими программами Вы сможете работать на своем компьютере.

От выбора ОС зависит также производительность Вашей работы, стапень защиты данных, необходимые аппаратные средства и т. д. Однако, выбор операционной системы также зависит от технических характеристик (конфигурации) компьютера.

Чем более современнее операционная система, тем она не только предоставляет больше возможностей и более наглядна, но также тем больше она предъявляет требований к компьютеру (тактовая частота процессора, оперативная и дисковая память, наличие и разрядность дополнительных карт и устройств).

**История развития ЭВМ**

История компьютера тесным образом связана с попытками облегчить и автоматизировать большие объемы вычислений, около 500г. Н.э. появились счёты (абак) — устройство, состоящее из набора костяшек, нанизанных на стержни.

Все основные идеи, которые лежат в основе работы компьютеров, были изложены еще в 1833 году английским математиком Чарльзом Бэббиджем. Он разработал проект машины для выполнения научных и технических расчетов, где предугадал основные устройства современного компьютера, а также его задачи. Управление такой машиной должно было осуществляться программным путем. Для ввода и вывода данных Бэббидж предлагал использовать перфокарты — листы из плотной бумаги с информацией, наносимой с помощью отверстий. Идеи Бэббиджа стали реально воплощаться в жизнь в конце 19 века.

Дальнейшие развития науки и техники позволили в 1940-х годах построить первые вычислительные машины. Создателем первого действующего компьютера Z1 с программным управлением считают немецкого инженера Конрада Цузе.

В феврале 1944 года на одном из предприятий Ай-Би-Эм (IBM) была создана машина Mark 1. Это был монстр весом около 35 тонн. В Mark 1 использовались механические элементы для представления чисел и электромеханические — для управления работой машины.

Развитие ЭВМ делится на несколько периодов. Поколения ЭВМ каждого периода отличаются друг от друга элементной базой и математическим обеспечением.

**Первое поколение**

К первому поколению обычно относят машины, созданные на рубеже 50-х годов. В их схемах использовались электронные лампы. Эти компьютеры были огромными, неудобными и слишком дорогими машинами, которые могли приобрести только крупные корпорации и правительства. Лампы потребляли огромное количество электроэнергии и выделяли много тепла. Быстродействие порядка 10-20 тысяч операций в секунду. Для ввода-вывода использовались перфоленты, перфокарты, магнитные ленты и печатающие устройства.

**Второе поколение**

Второе поколение компьютерной техники — машины, сконструированные примерно в 1955—65 гг. Характеризуются использованием в них как электронных ламп, так и дискретных транзисторных логических элементов. Их оперативная память была построена на магнитных сердечниках. В это время стал расширяться диапазон применяемого оборудования ввода-вывода, появились высокопроизводительные устройства для работы с магнитными лентами, магнитные барабаны и первые магнитные диски.

Быстродействие — до сотен тысяч операций в секунду, ёмкость памяти — до нескольких десятков тысяч слов.

**Третье поколение**

Машины третьего поколения созданы примерно после 60-x годов. Поскольку процесс создания компьютерной техники шел непрерывно, и в нём участвовало множество людей из разных стран, имеющих дело с решением различных проблем, трудно и бесполезно пытаться установить, когда поколение начиналось и заканчивалось. Возможно, наиболее важным критерием различия машин второго и третьего поколений является критерий, основанный на понятии архитектуры.

Машины третьего поколения — это семейства машин с единой архитектурой, т.е. программно совместимых. В качестве элементной базы в них используются интегральные схемы, которые также называются микросхемами.

Машины третьего поколения имеют развитые операционные системы. Они обладают возможностями мультипрограммирования, т.е. одновременного выполнения нескольких программ. Многие задачи управления памятью, устройствами и ресурсами стала брать на себя операционная система или же непосредственно сама машина.

Примеры машин третьего поколения — семейства IBM—360, IBM—370, ЕС ЭВМ (Единая система ЭВМ), СМ ЭВМ (Семейство малых ЭВМ) и др.

Быстродействие машин внутри семейства изменяется от нескольких десятков тысяч до миллионов операций в секунду. Ёмкость оперативной памяти достигает нескольких сотен тысяч слов.

**Четвёртое поколение**

— это теперешнее поколение компьютерной техники, разработанное после 1970 года.

Наиболее важный в концептуальном отношении критерий, по которому эти компьютеры можно отделить от машин третьего поколения, состоит в том, что машины четвёртого поколения проектировались в расчете на эффективное использование современных высокоуровневых языков и упрощение процесса программирования для конечного пользователя.

В аппаратурном отношении для них характерно широкое использование интегральных схем в качестве элементной базы, а также наличие быстродействующих запоминающих устройств с произвольной выборкой ёмкостью в десятки мегабайт.

C точки зрения структуры машины этого поколения представляют собой многопроцессорные и многомашинные комплексы, работающие на общую память и общее поле внешних устройств. Быстродействие составляет до нескольких десятков миллионов операций в секунду, ёмкость оперативной памяти порядка 1 — 64 Мбайт.

Для них характерны:

1. Применение персональных компьютеров;
2. Телекоммуникационная обработка данных;
3. Компьютерные сети;
4. Широкое применение систем управления базами данных;
5. Элементы интеллектуального поведения систем обработки данных и устройств.

**История возникновения и стандарты UNIX**

В 1965 году Bell Telephone Laboratories (подразделение AT&T) совместно с peneral jlectric qompang и Массачусетским технологическим институтом (rIT) начали разрабатывать новую операционную систему, названную rULTIqS (rULTipleoed Information and qomputing Service). Перед участниками проекта стояла цель создания многозадачной операционной системы разделения времени, способной обеспечить работу нескольких сотен пользователей. От Bell Labs в проекте приняли участие два участника s Кен Томпсон (ten Tompson) и Дmннис Ритчи (Dennis uitchie) . vотя система rULTIqS так и не была завершена (в 1969 году Bell Labs вышла из проекта), она стала предтечей операционной системы, впоследствии получившей название Unio.

Однако Томпсон, Ритчи ряд других сотрудников продолжили работу над создание удобной системы программирования. Использую идеи и разработки, появившиеся в результате работы над rULTIqS, они создали в 1969 году небольшую операционную систему, включавшую в себя файловую систему, подсистему управления процессами и небольшой набор утилит. Система была написана на ассемблере и применялась на компьютере nDn-7. Эта операционная система получила название UNIX, созвучное rULTIqS и придуманное другим членом группы разработчиков, Брайаном Керниганом (Brian ternigan).

Хотя ранняя версия UNIX много обещала, она не смогла бы реализовать весь свой потенциал без применения в каком-либо реальном проекте. И такой проект нашелся. Когда в 1971 году патентному отделу Bell Labs понадобилась система обработки текста, в качестве операционной системы была выбрана UNIX. К тому времени она была перенесена на более мощный nDn-11, да и сама немного подрослаw 16К занимала собственно система, 8К отводилось прикладным программам, максимальный размер файла был установлен в 64К при 512К дискового пространства.

Вскоре после создания первых ассемблерных версий Томсон начал работать над компилятором для языка FxuTuAN, а в результате разработал язык B. Это был интерпретатор со всеми свойственными интерпретатору ограничениями, и Ритчи переработал его в другой язык, названный q, позволявший генерировать машинный код. В 1973 году ядро операционной системы было переписано на языке высокого уровня С, s неслыханный до mтого шаг, оказавший громадное влияние на популярность UNIX. Это означало, что теперь система UNIX может быть перенесена на другие аппаратные платформы за считанные месяцы и внесение изменений не представляло особых трудностей. Число работающих UNIX-систем в Bell Labs превысило 25, и для сопровождения UNIX была сформирована группа UNIX Sgstem proup (USp).

**Исследовательские версии(AT&T Bell Labs)**

В соответствии с федеральным законодательством США, AT&T не имела права коммерческого распространения UNIX и использовала ее для собственных нужд, но, начиная с 1974 года, операционная система стала передаваться университетам для образовательных целей.

Операционная система модернизировалась, каждая новая версия снабжалась соответствующей редакцией Руководства Программиста, откуда и сами версии получили название редакций (jdition). Всего с 1971 по 1989 год было выпущено 10 редакций. Ниже перечислены наиболее важные редакции.

**Редакция 1 (1971)**

Первая версия UNIX, написанная на ассемблере для nDn-11. Включала в себя язык B и много известных комманд и утилит, в том числе cat, chdir, chmod, cp, ed, find, mail, mkdir, mkfs, mount, mv, rm, rmdir, wc, who. В основном использовалась как инструментальное средство обработки текстов для патентного отдела Bell Labs.

**Редакция 3 (1973)**

В системе появилась команда сс, запускавшая компилятор языка С. Число установленных систем достигло 16.

**Редакция 4 (1973)**

Первая система, в которой ядро написано на языке высокого уровня С.

**Редакция 6 (1975)**

Первая версия UNIX, доступная за пределами Bell Labs. Система полностью переписана на языке С. С mтого времени начинается появление новых версий, разработанных не в Bell Labs и рост популярности UNIX. Эта версия системы была установлена в Калифорнийском университете в Беркли, и на ее основе вскоре была выпущена первая версия BSD (Berheleg Softkare Distributuion) UNIX.

**Редакция 7 (1979)**

Включала в себя командный интерпретатор Bourne Shell и компилятор С от Кернигана и Ритчи. Yдро системы было переписано для переносимости на другие платформы. Лицензия на mту версию была куплена фирмой ricrosoft, которая разработала на ее базе операционную систему XjNIX.

Популярность UNIX росла, и к 1977 году число работающих систем превысило 500. В mтом же году система впервые была портирована на компьютер, отличный от nDn.

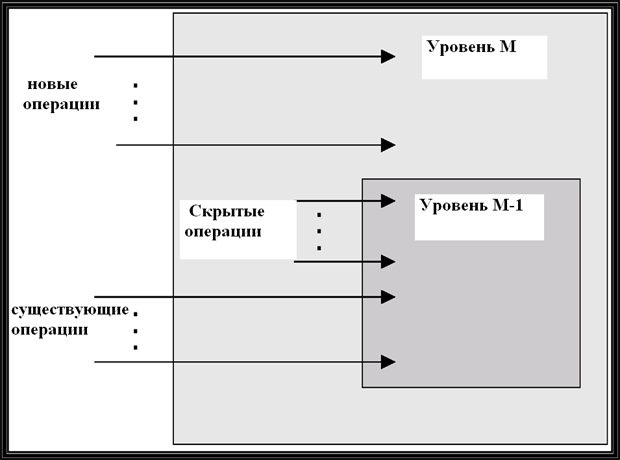
**Основные функции ОС**

Исполнение запросов программ (ввод и вывод данных, запуск и остановка других программ, выделение и освобождение дополнительной памяти и др.). Загрузка программ в оперативную память и их выполнение. Стандартизованный доступ к периферийным устройствам (устройства ввода-вывода).

Монолитное ядро ​​– это архитектура операционной системы, в которой вся операционная система работает в пространстве ядра. Это базовая операционная система, которая выполняет такие задачи, как управление файлами, управление устройствами, памятью и управление процессами. Все эти процессы напрямую контролируются в ядре. Это также управляет ресурсами между оборудованием и приложением наряду со службой пользователя и службами ядра, которые реализованы в одном и том же адресном пространстве. Это также увеличивает размер ядра и, следовательно, увеличивает размер операционной системы. Это ядро ​​было одной из наиболее используемых и предпочтительных архитектур ОС конца 1980-х годов, и многие операционные системы основывались на этом. Операционные системы, такие как Linux, Solaris, AIX, OpenVMS, HP-UX, DOS и т. Д., Используют архитектуру ОС с монолитным ядром.

Микроядро заведовало функциями взаимодействия процессов, управления виртуальной памятью, управления процессами и нитями (threads), управления процессорами (включая мультипроцессорные системы), а также управления вводом-выводом и обработки прерываний.

**Уровни абстракции ОС**

Проектирование и реализация операционных систем адекватно соответствуют принципу уровней абстракции, сформулированному Э. Дейкстрой и подробно рассмотренному в «История ОС. Отечественные ОС. Диалекты UNIX. Режимы пакетной обработки, мультипрограммирования, разделения времени .

Каждый новый уровень абстракции M определяет новые операции, в реализации которых используются операции предшествующего уровня M-1. Кроме того, уровень M-1 может иметь скрытые операции, используемые только на этом уровне и недоступные другим уровням.

На рис. 7.2 изображены уровни абстракции операционной системы OS/2 – второй ОС (вслед за MS DOS), разработанной фирмой Microsoft (в середине 1980-х гг.) для персональных компьютеров PS/2. OS/2 является значительно более развитой ОС, чем MS DOS, как видно из рисунка.



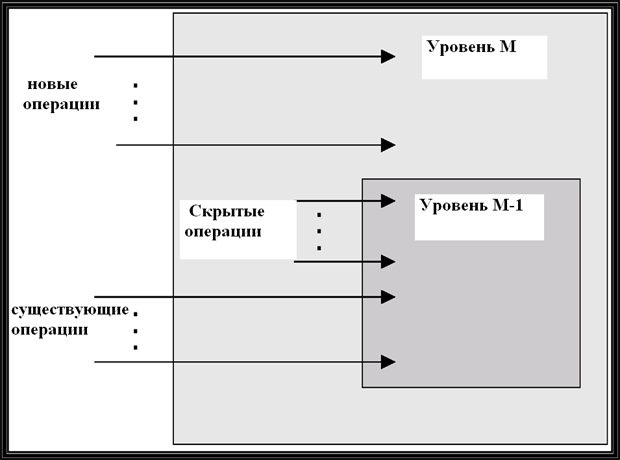
На верхнем уровне абстракции прикладные программы пользователей обращаются к интерфейсу прикладного программирования (application programming interface – API), который представляет собой набор библиотечных функций. API структурирован по подсистемам, причем набор подсистем расширяем. Нижние уровни абстракции реализуют ядро ОС, выполняющее функции управления памятью, диспетчеризации задач и управления устройствами. Наконец, на самом низком уровне абстракции реализованы драйверы устройств.

Уровень абстракции — один из способов сокрытия деталей реализации определенного набора функциональных возможностей. Применяется для управления сложностью проектируемой системы при декомпозиции, когда система представляется в виде иерархии уровней абстракции.

**Уровни абстракции ОС**

Проектирование и реализация операционных систем адекватно соответствуют принципу уровней абстракции, сформулированному Э. Дейкстрой и подробно рассмотренному в «История ОС. Отечественные ОС. Диалекты UNIX. Режимы пакетной обработки, мультипрограммирования, разделения времени «. На рис. 7.1 изображены уровни абстракции при разработке ОС.

Каждый новый уровень абстракции M определяет новые операции, в реализации которых используются операции предшествующего уровня M-1. Кроме того, уровень M-1 может иметь скрытые операции, используемые только на этом уровне и недоступные другим уровням.



На рис. 7.2 изображены уровни абстракции операционной системы OS/2 – второй ОС (вслед за MS DOS), разработанной фирмой Microsoft (в середине 1980-х гг.) для персональных компьютеров PS/2. OS/2 является значительно более развитой ОС, чем MS DOS, как видно из рисунка.

**Unix Socket**

**Unix domain socket, UDS) или IPC-сокет** (сокет межпроцессного взаимодействия) — конечная точка обмена данными, подобная Интернет-сокету, но не использующая сетевого протокола для взаимодействия (обмена данными). Используется в операционных системах, поддерживающих стандарт POSIX, для межпроцессного взаимодействия.

**Катало́г** — каталог, директория, справочник, папка — объект в файловой системе, упрощающий организацию файлов. Ранее часто использовалось слово директо́рия, которое представляет собой транслитерацию с английского (англ. directory). Типичная файловая система содержит большое количество файлов, и каталоги помогают упорядочить её путём их группировки. Каталог может быть реализован как специальный файл, где регистрируется информация о других файлах и каталогах на носителе информации.

**Символьное устройство — (character device)** — вид файла устройства в UNIX/Linux-системах, обеспечивающий интерфейс к устройству, реальному или виртуальному, с возможностью посимвольного обмена информацией.

В отличие от блочного устройства символьное устройство, как правило, не обладает возможностями произвольного доступа. В большинстве своём, чтение и запись данных в символьное устройство не буферизуется.

Типичные примеры символьных устройств: стриммер, модем, телетайп или терминал.

**Блочное устройство (block device)** — вид файла устройств в UNIX/Linux-системах, обеспечивающий интерфейс к устройству, реальному или виртуальному, в виде файла в файловой системе.

С блочным устройством обеспечивается обмен данными блоками данных. Как правило, это устройства произвольного доступа, то есть можно указать, из какого именно места должен быть прочитан или записан блок данных. Данные при чтении или записи на блочное устройство буферизуются.

Типичные примеры блочных устройств: жёсткий диск, CD-ROM, НГМД.

**В программировании именованный канал или именованный конвейер (англ. named pipe)** — один из методов межпроцессного взаимодействия, расширение понятия конвейера в Unix и подобных ОС. Именованный канал позволяет различным процессам обмениваться данными, даже если программы, выполняющиеся в этих процессах, изначально не были написаны для взаимодействия с другими программами. Это понятие также существует и в Microsoft Windows, хотя там его семантика существенно отличается. Традиционный канал — «безымянен», потому что существует анонимно и только во время выполнения процесса. Именованный канал — существует в системе и после завершения процесса. Он должен быть «отсоединён» или удалён, когда уже не используется. Процессы обычно подсоединяются к каналу для осуществления взаимодействия между ними.

**Символическая («мягкая») ссылка (также «симлинк», от англ. Symbolic link)** — специальный файл в файловой системе, в котором вместо пользовательских данных содержится путь к файлу, открываемому при обращении к данной ссылке (файлу).

Целью ссылки может быть любой объект: например другая ссылка, файл, каталог или даже несуществующий файл (в последнем случае при попытке открыть его должно выдаваться сообщение об отсутствии файла). Ссылка, указывающая на несуществующий файл, называется висячей или битой.

Символические ссылки используются для более удобной организации структуры файлов на компьютере, так как:

1)позволяют для одного файла или каталога иметь несколько имён и различных атрибутов;

2)свободны от некоторых ограничений, присущих жёстким ссылкам (последние действуют только в пределах одной файловой системы (одного раздела) и не могут ссылаться на каталоги).