1. Планирование процессов. Состояния процесса. Диспетчер процессов. Стратегия выбора процесса для активизации.

Процесс - выполняемый программный сегмент вместе с ресурсами, необходимыми для его выполнения (планирования).

Обычно для управления процессами, процессом выделяется центральный процессор.

Планирование – распределение ресурсов центрального процессора между конкурирующими процессами, путем передачи им управления согласно некоторой стратегии планирования.

Процесс может находиться в ряде состояний, и каждому состоянию в ОС соответствует примитив, обрабатывающий это состояние.

Процесс создаётся, когда начинается выполнение задания пользователя и уничтожается, когда задание заканчивается.

В однопроцессорной системе в памяти в каждый момент времени может находиться только один процесс. В системах коллективного пользования все процессы активизируются по очереди, получая квант времени ЦП. В многопроцессорных системах одновременно может существовать несколько процессов.

Процесс активен, если он использует центральный процессор для выполнения команд.

Процесс блокирован, если он ждёт наступления некоторого события.

Процесс находится в состоянии готовности, если он неактивен и неблокирован, и если ему выделены необходимые ресурсы.

В любой ОС существуют очереди:

- активных процессов;
- очередь процессов в состоянии готовности;
- очередь блокированных процессов.

Выбор процесса и передача ему управления называется диспетчеризацией. В случае блокировки процесса активизируется новый процесс. Когда ожидаемое событие для блокируемого процесса наступило, он переводится в состояние готовности и ждёт завершения второго процесса.

Любой процесс характеризуется блоком состояния процесса (PSW):

- Имя процесса
- Дескриптор процесса
- Информация о системных ресурсах
- Номер в очереди
- Номер дескриптора процесса
- Состояние процессора

Процессы обмениваются между собой сигналами.

Сигнал – это объект, данные в котором представляют собой следующую структуру:

1. Номер процесса;
2. Номер дескриптора процесса;

3….n) информация о состоянии передаваемым сигналом (номера семафоров, их активность, номера очередей)

Управляет процессами (нитями) – диспетчер процессов.

Алгоритм работы диспетчера:

PROCEDURE DISPACH:

<обновить PSW активного процесса, если он есть>;

<выбрать следующий готовый процесс для передачи ему управления>.

Выбор процесса осуществляется в соответствии с некоторой стратегией LIFO, FIFO.

```
if <готовый процесс найден>
        then
      begin
         <пометить выбранный процесс как активный>;
         <выделить квант времени, установив привилегированной командой таймер>;
         <передать управление выбранному процессу>
      end
   else
      <перевести центральный процессор привилегированной командой в состояние простоя>;
```

Стратегия выбора процесса для активизации:

1. круговая стратегия (в мультипрограммных системах) – в этом случае все процессы считаются равноценными, и диспетчер циклически просматривает PSW всех процессов и выделяет им по очереди квант времени.

2. Выбор процессов по приоритету – приоритеты могут назначаться пользователем, либо их устанавливает ОС в целях повышения производительности всей системы.

2. Управление памятью. Архитектура, подкачка страниц, кэш файловой системы. Достижение оптимальной производительности.

Оперативная память компьютера является наиболее интенсивно потребляемым и дефицитным ресурсом операционной системы. От эффективности ее использования в наибольшей степени зависит производительность всего комплекса.

Оперативная память - особый ресурс, распределяемым не изолированно, а согласованно с разделением времени ЦП, при этом стремятся обеспечить потребности максимального числа параллельных процессов и тем самым повысить пропускную способность системы.

Один из методов эффективного управления памятью - разделение всей доступной оперативной памяти на страницы фиксированного размера. Все страницы последовательно нумеруются, так что номер однозначно определяет диапазон адресов в памяти. Таким образом, любой адрес можно представить в виде суммы номера страницы и смещения внутри страницы.

Страницы, принадлежащие одному процессу, чередуются со свободными или принадлежащими другому процессу, располагаться в памяти в произвольном порядке, так что начало процесса располагается в страницах с номерами большими, чем его конец. Вместе с тем, в хвосте последней страницы может оказаться некоторый диапазон не используемых, пропадающих адресов. Что бы уменьшить потери памяти, часто используют страницы небольшого размера от 512 байт до 4 Кбайт.

Архитектура памяти

ОС Win 2k стремится выделить каждому приложению до 4 ГБ памяти.

Диспетчер виртуальной памяти (VMM) предоставляет приложениям диапазоны виртуальных адресов для доступа к информации, которые ограничиваются в физической ОП.

На самом деле VMM отображает виртуальные адреса в физическое адресное пространство. Последнее не может вместить все порции, выделенные каждому процессу. Отсюда VMM постоянно перебрасывает информацию от жесткого диска в ОП и обратно – подкачка страниц.

Подкачка страниц

VMM вытесняет из ОП программный код и данные, которые процесс не использует активно в данный момент. Подкачка страниц означает, что информация перемещается в файл подкачки. Информация вытесняется из ОП и попадает либо в кеш ФС, либо в файл подкачки.

Кэш — промежуточный буфер с быстрым доступом, содержащий информацию, которая может быть запрошена с наибольшей вероятностью. Доступ к данным в кэше идёт быстрее, чем выборка исходных данных из оперативной (ОЗУ) и быстрее внешней (жёсткий диск или твердотельный накопитель) памяти, за счёт чего уменьшается среднее время доступа и увеличивается общая производительность компьютерной системы. Прямой доступ к данным, хранящимся в кэше, программным путем невозможен.

Кеш ФС – отводится для информации, только что отправленной на диск, а так же для информации, которая будет скоро запрошена.

Когда VMM хочет извлечь информацию, которой нет в ОП, он генерирует событие: «сбой доступа к странице». Это приводит к тому, что информация извлекается из файла подкачки pagefile.sys с жесткого диска. Однако считывание занимает много времени.

Кеш ФС

Win 2k регулирует размер кеша ФС в динамическом режиме. В целом размер кеша не должен превышать размера, при котором на долю системы остается менее 4 ГБ свободной памяти. Это не позволит ей успешно обслуживать большое приложение. Размер кеша вручную изменить нельзя.

3. Управление ресурсами в ОС. Операции ввода и вывода ресурсов. Алгоритм операций put/get.

Ресурсы системы состоят из повторно-используемых аппаратных и программных средств, таких как:

- основная и внешняя память;
- блокировки (прерывания), связанные с критическими областями;
- потребляемые ресурсы (межпроцессорные сообщения и управляющие сигналы, формируемые прерываниями);
- время ЦП

Критической областью в одной или нескольких программах - группа команд или часть команды, для выполнения которых требуется использовать некоторый разделяемый ресурс. Отсутствие его - может привести к блокировке программы или её зависанию. Чтобы этого не происходило, надо иметь средства управления критическими состояниями.

Состояние и стратегия выделения ресурса определяется дескриптором конкретного ресурса. Дескриптор в ОС имеет поля:

1) указатель на список свободных устройств или сообщений;
2) указатель на список заблокированных процессов;
3) указатель на программу-распределитель данного ресурса;

Заблокированные процессы организуют ссылки на дескрипторы ресурсов через дескрипторы процессов и в каждой очереди списков дескрипторов существуют программы вставки и удаления элементов.

Для работы с ресурсами используются операции: get«ресурс» и put«ресурс».

Процесс, требующий ресурс R формирует вызов или запрос: *Call R Get (m)* (m – список параметров), R – Область памяти, в которой содержится сведения о ресурсе.

Запрашивающий ресурс процесс устанавливается в заблокированное состояние до тех пор пока не получит затребованный ресурс.

Процедура *put* либо возвращает ресурс, полученный предварительно, либо формирует потребляемый ресурс (сообщение). Операции *put* и *get* выполняются планировщиком задач как единое целое.

Алгоритм операций *put/get.*

```
Procedure get (m);
/* разрешить процедуре get быть текущим процессом ЦП – СР */
/* блокировать процесс СР и вставить данные m в блокированный процесс СР */
CP.Status = ‘ blocked’;
CP.BlockData = m;
call waitList_Insert(CP.List)
/* связать СР со списком ожидания ресурсов */
/* вызвать распределитель ресурсов Allocator(K, PD)
```

Allocator – это логическая функция, осуществляющая поиск среди имеющихся устройств (ресурсов), способных удовлетворить запросу; PD(i) – указывает на процессы, запросы которых уже выполнены. K – текущий номер запроса на ресурс */

```
if Allocator(K,PD) then
begin
  <выполнить процедуру get>
  /* повторить текущий процесс СР */
  CP.Status=’running’;
  /*включить текущ. процесс СР в начало списка в соответствии с его приоритетом*/
  call ReadyList_Insert2(CP.List)
end
else
  /*процедура get не выполняется. Найти другой процесс*/
call Schedular; /* необходимо обратиться к планировщику */
end.{get}
```

procedure put(m) вызывает распределитель Allocator(K, PD). Затем циклически от 1 до K переводит все процессы в состояние готовности, вставляя их в конец списка, в соответствии с их приоритетом, а затем вызывает планировщик, который вызывает готовый процесс с наивысшим приоритетом. Все потребляемые ресурсы обычно связаны с направленными сообщениями, включающими в себя операции get и put.

4. Основные понятия и концепция виртуальной памяти. Алгоритм программы страничных прерываний (программный).

Виртуальная память (ВП) — технология управления памятью ЭВМ, разработанная для многозадачных операционных систем. При использовании данной технологии для каждой программы используются независимые схемы адресации памяти, отображающиеся тем или иным способом на физические адреса в памяти ЭВМ.

Физическая память – это среда хранения из элементов, адресуемых в соответствии с физическими возможностями памяти и принятым способом адресации.

Множество адресов, упорядоченных по некоторому признаку, называют адресным пространством.

Механизм физической адресации элементов памяти, есть основа более сложного механизма доступа, реализуемого программно-аппаратными методами. Основная задача диспетчера виртуальной памяти заключается в отображении линейного пространства адресов виртуальной памяти на часть адресного пространства физической памяти.

Построение механизма виртуальной памяти основано на решении четырёх задач:

Задача размещения: выбор страницы (сегмента) в ОЗУ, куда будут отображаться страницы (сегменты) виртуальной памяти. (необходимо произвести преобразование виртуальных адресов в физические, и наоборот).

Задача преобразования: преобразует адрес виртуальный в адрес оперативный и наоборот.

Задача перемещения: в архивной среде выбирается такая информация, которая принадлежит отображаемым виртуальным страницам, и она передаётся в страницы (сегменты) ОЗУ, найденные в результате решения задачи размещения.

Задача замещения: заключается в выборе кандидата на перераспределение.

ОС содержит 2 таблицы, описывающие состояние страниц и сегментов:

1) РМТ – это карта памяти определяет положение сегмента в ОП;

2) Таблица страничных кадров (ТСК) – следит за состоянием страниц.

Когда возникает прерывание, таблица рассматривается в поисках свободного страничного кадра. Он найден - требуемая страница может быть загружена немедленно. Иначе, страница, находящаяся в памяти, должна быть вытолкнута, чтобы освободить место для той страницы, которую надо загрузить. Все операции делает программа страничных прерываний, при этом она осуществляет динамичное преобразование адресов и обрабатывает страничные прерывания.

Часть алгоритма этой программы реализуется аппаратным способом, а часть – программным.

Программный алгоритм страничных прерываний:

```
Procedure PAGEFAULT {реализована как часть ОС}
  <сохранить состояние процесса из рабочей области прерываний>;
  <пометить этот процесс как блокированный (blocked)>;
  if <имеется свободный страничный кадр> then
    begin
      <выбрать свободный страничный кадр>;
      <пометить выбранный кадр в таблице страничных кадров как 'занятый'>;
      <разрешить все прерывания>
    end
  else
    begin
       <выбрать страницу для выталкивания>;
       <пометить выбранный кадр в таблице страничных кадров как 'занятый'>;
      <разрешить все прерывания>;
      if <выбранная страница была модифицирована> then
        begin
          <обновить РМТ и таблицу страничных кадров>;
          <выдать запрос на ввод-вывод>;
          <ждать завершения операции записи>;
```

```
      end
    end;
  <выдать запрос на ввод-вывод для чтения страниц в выбранный страничный кадр>;
  <ожидать завершения операции чтения>;
  <обновить PMT и таблицу страничных кадров>;
  <восстановить состояние пользовательского процесса, пометив процесс, как “ready”>
end.
```

5. Механизм управления и средства взаимодействия параллельных процессов. Типичные задачи синхронизации.

Управление параллельно-развивающимися процессами определяется характером взаимодействия между ними. Это значит, что существуют группы задач по управлению этими процессами. Для управления параллельными процессами ОС определяем процессы в виде синхронизирующих правил, зависящих от вида отношений между ними:

1) Отношение предшествования (порождающие и порожденные процессы);

2) Отношение приоритетности;

3) Отношение взаимного исключения.

Реализация синхронизирующих правил осуществляется с помощью механизмов синхронизации, которых к данному времени разработано много и реализация которых обычно имеет программно-аппаратную форму. Эти механизмы выполняют двоякую роль:

1) Они обеспечивают способ упорядочивания развития процессов;

2) Они обеспечивают взаимодействие между процессами.

Типичные задачи синхронизации:

1) Задача взаимного исключения; Нужно согласовать работу N>2 параллельных процессов при использовании некоторого критического ресурса при следующих требованиях:

одновременно внутри критической области должно находиться не более одного процесса;

критические области не имеют приоритетов в отношении друг к другу;

остановка процесса вне его критической области не должна влиять на дальнейшую работу по использованию критического ресурса;

решения обхождения равно приоритетных процессов в их критической области не откладывается на неопределенное время;

относительные скорости процессов неизвестны и произвольны;

процесс, использующий критический ресурс, выходит из критической области за конечное время.

Критическая область в программе – группа команд для выполнения которых требуется разделяемый ресурс.

2) Производитель-потребитель; Простейший случай взаимодействия двух процессов с жестко закреплёнными функциями: один процесс вырабатывает сообщение (производитель), а другой их обрабатывает (потребитель).

Взаимодействие этих процессов производится через некоторую обобщенную область оперативной памяти, которая по смыслу и представляет критический ресурс.

3) Читатели-писатели; Возникает при построении файловых систем. Критический ресурс – некоторая область памяти, и с этой областью памяти работают процессы двух типов: читатели и писатели. Запись информации в эту область должна производиться на основе взаимоисключений (записывает только один процесс).

4) «Обедающие философы» - задача распределения пересекающихся ресурсов. Пусть x, y, z – параллельные процессы.

P1 – устройство ввода;

P2 – печатающее устройство;

P3 – накопитель на магнитном диске;

Для активного состояния процесса X требуется P1 и P2 , для У - P2 и P3 , для Z - P1 и P3. Скорости развития процессов – произвольная.

Возникают две ситуации:

1) «тупиковая», когда ни один из процессов не может получить ресурса;

2) «голодание», когда один процесс блокируется на неопределённое время.

6. Классификация программного обеспечения ЭВМ. Свойства ПО. Функции операционных систем.

ППО (прикладное программное обеспечение) – это программы, комплексы программ и пакеты программ, которые предназначены для решения задач из данной предметной области. (MathCad , Microsoft Office , Бухгалтерия 1С и т. д.)

СПО (системное программное обеспечение) - это программы, комплексы программ и пакеты программ, которые предназначены для обеспечения эффективной организации вычислительного процесса на вычислительной системе.

САП (Системы автоматизированного программирования).

Текстовые редакторы: а) для составления программ;
б) для записи и редактирования пакетных файлов.

Ассемблер - это язык символического кодирования, предназначенный для записи алгоритма в виде псевдокоманд, имитирующий команды вычислительной машины.

Препроцессор – это программа, позволяющая использовать в программном коде директивы, позволяющие включать файлы, обобщенные имена констант, условные конструкции и т. д.

Транслятор – переводит исходный модуль, написанный на языке высокого уровня в объектный код.

Компиляторы – это трансляторы, в которых сначала осуществляется полный перевод программы, а затем ее выполнение.

Интерпретаторы – это трансляторы, в которых фазы перевода и выполнения меняются (повторяются).

Загрузчик – это программа, которая позволяет разместить другую программу в определенное место памяти (назначить адреса).

Редакторы связи (связывающие загрузчики) – это загрузчики, которые позволяют не только размещать программу в памяти, но и компоновать ее из отдельно-транслируемых модулей.

Отладчики – это программы, позволяющие отыскивать ошибки времени исполнения. Синтаксические ошибки отлавливаются транслятором, а ошибки времени выполнения отлавливаются ОС.

Свойства ПО.

1) - Машинно-зависимые свойства определяются процентом операторов программ, написанных в коде данной машины.

- Машинно-независимые свойства определяются процентом операторов программ, записанных на языке высокого уровня.

2) Переносимость.

Если СПО можно ставить на вычислительные системы различной архитектуры то оно переносимо.

3) Вариабельность.

Свойство ПО подвергаться модификации.

Функции ОС:

загрузка и передача управления первой команде выполняемой программы;

выделяет необходимые ресурсы для выполнения данной программы или распределяет их между несколькими программами;

обнаруживает сбои или ошибки в ходе вычислительного процесса;

предоставляет пользователю средства для управления ходом вычислительного процесса, средства для настройки ОС, средства, информирующие пользователя о ходе вычислительного процесса (интерфейс пользователя).

7. Системное обеспечение управления вычислительного процесса. Определение ОС.

Операционная система – это комплекс программ, предназначенных для эффективной организации вычислительного процесса, управления ходом этого процесса и создания удобного интерфейса для пользователя.

Функции ОС:

загрузка и передача управления первой команде выполняемой программы;

выделяет необходимые ресурсы для выполнения данной программы или распределяет их между несколькими программами;

обнаруживает сбои или ошибки в ходе вычислительного процесса;

предоставляет пользователю средства для управления ходом вычислительного процесса, средства для настройки ОС, средства, информирующие пользователя о ходе вычислительного процесса (интерфейс пользователя);

Типы ОС:

1) однопрограммные ОС (в памяти только одна программа, и все ресурсу принадлежат ей(MS – DOS));

2) многопрограммные ОС (одновременно в памяти выполняется несколько задач). Различают истинную мультизадачность (в этом случае есть несколько процессоров в системе, каждый из которых выполняет свою задачу (ЕС1096)) и кажущуюся мультизадачность (процессор ОС переключается на выполнение первой, второй, третьей и т. д. программ).

3) системы коллективного пользования (допускают одновременную работу нескольких пользователей под управлением одной ОС, выделяя каждому пользователю фиксированные ресурсы и определенное количество времени) – (UNIX , LINUX);

4) системы реального времени (в этих системах время реакции системы на любое событие в управляемом объекте не превышает времени завершения этого события(RT/11 , RSX));

5) Сетевые системы.(MS Windows NT)

Структура ОС и основные понятия ОС

Любая ОС состоит из двух основных частей: ядра и нерезидентной части, которая содержит процедуры, нечасто используемые при работе ОС. Ядро содержит загрузчик ОС и набор процедур, управляющих теми или иными функциями ОС, а именно:

подсистему управления процессов (диспетчер).

подсистему управления основной памяти.

подсистему управления виртуальной памяти (если она есть).

Кроме того ОС содержит совокупность резидентных драйверов: драйвер диска, мыши, клавиатуры и др.

Также содержит командные процессоры-интерпретаторы входного языка ОС. Командный процессор служит для обработки команд ОС и выдачи сообщений.

В состав ядра входят подсистемы управления ввода – вывода (BIOS для DOS).

Для ОС отводится фиксированная область памяти, защищенная от несанкционированного доступа.

Примитив - процедура, реализующая ту или иную элементарную функцию ОС (создание буферов ввода-вывода, просмотр буфера ввода и т. д.).

В состав системных библиотек обычно входят команды ОС: внутренние и внешние.

В любой ОС в системной области выделяется «среда округления». Она играет роль почтового ящика для обмена информацией между программами.

Любая ОС имеет два файла:

файл конфигурации системы (определяет конкретную на данный момент конфигурацию ОС и ее параметры).

Файл автозапуска (предназначен для формирования удобной для пользователя операционной среды).

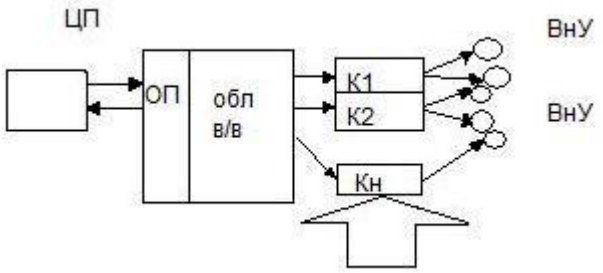
8. Обслуживание ввода/вывода. Каналы ввода/вывода. Типы каналов. Диспетчер заданий.

Обслуживание ввода-вывода - это группа процессов, которые инициируются программами или самой ОС, и которые связаны с передачей информации между оперативной памятью и дисками, оперативной памятью и магнитными лентами, между магнитными лентами и магнитными дисками.

Обмен информацией осуществляется побайтно или блоками с помощью специально-выделенных каналов.

Системы ввода-вывода включают как правило группу буферов ввода-вывода, каналов ввода-вывода.

Канал – это специализированная на вводе-выводе машина. Канал работает на специальной канальной программе, в которой используются привилегированные команды(внутренние команды ЦП).



Каналы делятся на три типа:

А) мультиплексные (навешано много устройств ввода-вывода, действующих медленно: поэтому канальная программа должна переключаться от одного канала к другому(принтеры));

Б) селекторные (одно устройство быстрое, с большим объёмом (винчестер));

В) DMA (прямой доступ к памяти) – канал, выполняющий канальную программу без вмешательства центрального процессора. По окончанию программы канал генерирует прерывания по вводу-выводу, которые могут работать параллельно.

В мультипрограммной системе одновременно могут существовать процессы пользователя Ui ,которые обрабатываются диспетчером заданий(SuperVisor). Диспетчер задания имеет основную часть – загрузчик заданий, который обеспечивает загрузку заданий с дискового накопителя в оперативную память. Управление загрузчиком и процессами пользователя осуществляет диспетчер заданий или диспетчер программ, который:

1) определяет, какое задание надо выбрать для загрузки в оперативную память;

2) формирует процесс Ui пользователя, а именно его слово состояния PSW;

3) завершает выполнение очередного задания.

В любой ОС есть 2 важных процесса: спулер ввода и спулер вывода.

Спулер – процесс ввода-вывода в режиме реального времени.

Диспетчер спулинга – это программа, которая формирует Pool(пул) буфер, т.е. структуру, соответствующую каждому буферу (занят – незанят). Из этих структур формируются очереди пустых буферов, готовых к вводу. Диспетчер спулинга обеспечивает одновременную работу всех буферов и непосредственно взаимодействует с управляющими программами устройств ввода и печатающих устройств.

9 Понятие файла.Хар-ка файлов. Назна-е и ф-ции ФС.

Файл – именованная область внешней памяти, в которую можно записать, и из которой можно считывать данные.

Файлы хранятся в памяти, не зависящей от энергопитания (магнитные носители).

Основные цели использования файла:

1) Долговременное и надежное хранение информации. Долговременность достигается за счет использования энергонезависимых устройств. Надежность определяется средствами защиты доступа к файлам и общей организации программного кода ОС, при котором сбои аппаратуры чаще всего не разрушают информацию, хранящуюся в файле.
2) Совместное использование информации.

Файлы обеспечивают естественный и легкий способ разделения информации между пользователями за счет наличия понятного человеку символьного имени, постоянства хранимой информации и расположения файла

Пользователь должен иметь удобные средства работы с файлами (справочники, каталоги и др.).

Файл может быть создан одним пользователем, а использоваться другим, при этом можно определить права доступа к файлу другими пользователями. Эти действия реализуются ФС ОС.

Характеристики файла

В понятие файла входят: имя, данные и атрибуты.

Атрибуты – информация, описывающая свойства файла.

Примеры возможных атрибутов: тип файла, владелец файла, информация о разрешенных операциях, время создания…

Признаки: системный, архивный, скрытый, только для чтения.

Набор атрибутов определяется спецификой ФС. Пользователь может получить доступ к атрибутам, используя средства ФС. Обычно разрешается читать значения всех атрибутов, но менять некоторые.

Значения атрибутов могут содержаться в каталогах(MS DOS). Другой вариант – размещение атрибутов в специальных таблицах, а в каталогах только ссылки на них(Unix). В этой ФС структура каталогов – простая запись о каждом файле. Она содержит короткое символьное имя файла и указатель на индекс дескриптора файла.

В обоих вариантах каталоги обеспечивают связь между именами файлов и самими файлами. Второй подход(Unix) наиболее гибкий. Файл может быть включен сразу в несколько каталогов под разными именами, но ссылка на дескриптор будет одна и та же.

Назначение и функции ФС

Главной функцией обеспечиваемой СУФ является:

1) Создание файла – определить имя и выделить место, которое ему необходимо;
2) Уничтожение файла – освободить имя и занимаемое файлом пространство.

Удалить файл логически – имя исчезнет из каталога, но содержимое останется.

Удалить файл физически – полное удаление;

3) Открытие файла – объявить системе о намерении использовать этот файл с определенными функциями и доступом;
4) Закрытие открытого файла – запретить всякий последующий доступ к нему;
5) Другие – разные функции проверки и модификации (чтение/запись i-той записи).

Эти функции являются базовыми операциями во всех СУФ. В ОС СУФ играет главную роль, потому что она должна управлять большей частью информации, принадлежащей как пользователям, так и самой ОС. Часто СУФ тесно связана с системой В/В.

10 Тупики.Усл-я возник-я Т.Стратегия Хаверда.Предотвращение Т.Алг Банкира и его недост.

В мультипрограммной системе процессы находятся в состоянии тупика, если они ожидают некоторое событие, которое никогда не произойдет. Системная тупиковая ситуация или зависание системы – это ситуация, когда один или более процессов оказываются в состоянии тупика. В мультипрограммной ЭВМ одной из главных функций ОС является распределение ресурсов. Когда ресурс распределяется между многими пользователями, каждому из которых предоставляется право исключительного управления своими ресурсами. Возможно появление тупиков, из-за которых процессы некоторых пользователей никогда не могут завершиться.

условия возникновения

Процессы требуют предоставления им права монопольного управления ресурсами – условие взаимоисключения.

Процессы, удерживающие за собой ресурсы, выделяемые им, ожидают выделения дополнительных ресурсов – условие ожидания ресурсов.

Ресурсы нельзя отобрать у процессов, удерживающих их, пока эти ресурсы не будут использованы для завершения работы – условие неперераспределяемости.

Существует кольцевая цепь процессов, в которой каждый процесс удерживает один или более ресурсов, необходимых следующему процессу – условие кругового ожидания.

Стратегия Хавендера

Хавендер показал, что возникновение тупиков невозможно, если нарушено 1 из условий возникновения тупиков.

Стратегия:

Каждый процесс должен запрашивать все необходимые ему ресурсы сразу, причем он не может начать выполнение до тех пор, пока все они не будут ему предоставлены.

Если процесс, удерживающий определенные ресурсы, получит отказ на запрос на дополнительный ресурс, то этот процесс должен освободить свои первоначальные ресурсы и при необходимости запросить их снова вместе с дополнительными.

Если процессу выделяются ресурсы данного типа, то в дальнейшем он может запросить только ресурсы более далеких по порядку типов.

Предотвращение тупиков и алгоритм банкира

Если необходимые условия возникновения тупиков не нарушены, то все равно можно избежать тупиковой ситуации, если рационально распределены ресурсы, придерживаясь определенных правил. Наиболее известным является алгоритм банкира (придумал Дейкстра), который имитирует действия банкира, который из определенных источников капитала дает ссуды и принимает платежи. Текущее состояние ВМ называется надежным, если ОС может обеспечить всем текущим пользователям завершение их задач в течение конечного времени. Ненадежное состояние – когда со временем может привести к тупику.

Распределение ресурсов согласно алгоритму банкира

Условия взаимоисключения, ожидания дополнительных ресурсов и неперераспределенности выполняются. Процессы могут претендовать на монопольное использование ресурсов, которые требуются. Процессам реально разрешается удержание ресурсов при запросе дополнительных ресурсов, причем ресурсы нельзя отбирать у процессов, которым они выделены. Система может удовлетворить/отклонить запрос. Если запрос отклоняется, пользователь удерживает за собой ресурсы, ждет некоторое время, пока запрос не будет удовлетворен. Система удовлетворяет только те запросы, при которых ее состояние остается надежным, а запрос, приводящий к ненадежному состоянию, откладывается до определенного момента, когда его можно выполнить.

Недостатки алгоритма банкира

Требуется, чтобы:

1. число работающих пользователей оставалось постоянным, что нереально.
2. распределитель ресурсов гарантированно должен удовлетворить все запросы за некоторый конечный период времени (нужно больше гарантий).
3. задания и процессы гарантированно должны возвращать ресурсы в течение конечного времени.
4. пользователи заранее указывали свои максимальные потребности в ресурсах

11 Классификация ОС

ОС могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами), особенностями использованных методов проектирования, типами аппаратных платформ, областями использования и многими другими свойствами.

1. Особенности алгоритмов управления ресурсами

От эффективности алгоритмов управления локальными ресурсами компьютера во многом зависит эффективность всей сетевой ОС в целом.Характеризуя сетевую ОС, часто приводят важнейшие особенности реализации функций ОС по управлению процессорами, памятью, внешними устройствами автономного компьютера.В зависимости от особенностей использованного алгоритма управления процессором, ОС делят на многозадачные и однозадачные, многопользовательские и однопользовательские,на многопроцессорные и однопроцессорные.
Поддержка многозадачности. По числу одновременно выполняемых задач ОС делят на: однозадачные (например, MS-DOS, MSX) и многозадачные (ОС ЕС, OS/2, UNIX, Windows 95).

Однозадачные ОС в основном выполняют функцию предоставления пользователю виртуальной машины, делая более простым и удобным процесс взаимодействия пользователя с компьютером. Однозадачные ОС включают средства управления периферийными устройствами, средства управления файлами, средства общения с пользователем.

Поддержка многопользовательского режима. По числу одновременно работающих пользователей ОС делятся на: 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.

Многопроцессорные ОС могут классифицироваться по способу организации вычислительного процесса в системе с многопроцессорной архитектурой: асимметричные ОС и симметричные ОС. Асимметричная ОС целиком выполняется только на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам. Симметричная ОС полностью децентрализована и использует весь пул процессоров, разделяя их между системными и прикладными задачами.
Специфика ОС проявляется и в том, каким образом она реализует сетевые функции: распознавание и перенаправление в сеть запросов к удаленным ресурсам, передача сообщений по сети, выполнение удаленных запросов. При реализации сетевых функций возникает комплекс задач, связанных с распределенным характером хранения и обработки данных в сети: ведение справочной информации о всех доступных в сети ресурсах и серверах, адресация взаимодействующих процессов, обеспечение прозрачности доступа, тиражирование данных, согласование копий, поддержка безопасности данных.

Особенности аппаратных платформ

Сетевая ОС имеет в своем составе средства передачи сообщений между компьютерами по линиям связи, которые совершенно не нужны в автономной ОС.

Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:
системы пакетной обработки (например, ОС ЕС), системы разделения времени (UNIX, VMS), системы реального времени (QNX, RT/11).

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.
Системы разделения времени призваны исправить основной недостаток систем пакетной обработки - изоляцию пользователя-программиста от процесса выполнения его задач
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
Некоторые операционные системы могут совмещать в себе свойства систем разных типов, например, часть задач может выполняться в режиме пакетной обработки, а часть - в режиме реального времени или в режиме разделения времени. В таких случаях режим пакетной обработки часто называют фоновым режимом.

12 **Современные файловые системы: FAT32, NTFS, HPFS, Ext2/ext3 и др.**

FAT32

Важная характеристика FAT – использование имен файлов в формате 8.3. К стандартной FAT добавились еще 2 разновидности от MS: VFAT и FAT32.

Сейчас FAT32 поддерживается Win NT/2k/Me и Linux. Имеются СУФ FAT32.

ФС VFAT появилась в Win 3.11 и была предназначена для выполнения файлового в/в в защищенном режиме. В Win 95 в VFAT добавилась поддержка длинных имен файлов. VFAT сохранила совместимость с FAT.

Недостатки FAT и VFAT: большие потери на кластеризацию при больших размерах логического диска. Это привело к разработке новой ФС с использованием FAT – FAT32.

FAT32 является полностью совместимой 32 разрядной ФС и содержит многочисленные усовершенствования и дополнения по сравнению с предыдущими реализациями FAT.

Отличие FAT32 в том, что она использует кластеры меньшего размера по сравнению с предыдущими версиями => по сравнению с FAT16 экономится дисковое пространство (10-15%).

Для длинных имен файлов используется несколько элементов каталога => появление длинных имен приводит к уменьшению количества файлов в каталоге корня. MS рекомендует избегать длинных имен файлов в корневых каталогах FAT.

Файловая система HPFS

Появилась в OS/2, разработана IBM и MS.

Архитектура HPFS использует преимущества многозадачного режима и обеспечивает надежную и эффективную работу на дисках большого объема.

HPFS – первая ФС для ПК, в которой реализована поддержка длинных имен. Она поддерживает атрибуты как у FAT. Размещение файлов на диске с помощью HPFS увеличит производительность и надежность системы в целом.

Способы достижения:

1) Размещение каталогов в середине дискового пространства;
2) Исп-е бинарных сбалансированных деревьев для ускорения поиска информации о файле;
3) Рассредоточение информации о местоположении записей файлов по всему диску(записи каждого файла размещаются в смежных секторах и близко от данных их местоположения.)

В HPFS диск разбит на полосы => избегается дополнительное движение головок.

В HPFS дисковое пространство выделяется не кластерами, а блоками. Размещение файлов в небольших блоках позволяет более эффективно использовать дисковое пространство, т.к. потери свободного места – 256 Байт на файл. Чем больше размер кластера, тем больше места на диске расходуется зря, но FAT занимает меньше места.

В HPFS структура каталога – сбалансированное дерево с записями, расположенными в алфавитном порядке.Система HPFS использует механизм HotFix (аварийное замещение). Полоса в центре диска используется для хранения каталогов (Directory Band) = 8 МБ.

NTFS:

- работа с дисками большого объема происходит намного эффективнее;
- имеются средства ограничения доступа к файлам;
- введены механизмы, повышающие надежность ФС;
- снято ограничение на максимальное количество кластеров.

Дополнительные возможности:

- усовершенствованная отказоустойчивость;- эмуляция других ФС;- мощная модель безопасности;- параллельная обработка потоков данных;- создание файловых атрибутов, определяемых пользователем.

Поддержка системы POSIX.

Это международный стандарт на машинно-независимый (переносимый) интерфейс компьютерной среды. Поддержка POSIX введена из-за того, что все закупаемые Америкой системы должны соответствовать этому стандарту.

Гибкость – модель распределения дискового пространства отличается гибкостью:

1) размер кластера меняется от 512 байт до 64 КБ;
2) поддерживаются длинные имена файлов, набор символов Unicode и альтернативные имена формата 8.3 для FAT;
3) хорошо работает с томами 300-400 МБ и выше
4) количество файлов в каталогах неограниченно, т.к. в основу структуры каталогов заложены эффективные структуры данных – бинарные деревья.

Обладает средствами самовосстановления.
NT поддерживает механизм проверки целостности системы. NT обеспечивает безопасность на уровне файлов: права доступа к томам, каталогам и файлам зависят от учетной записи пользователя и тех групп, к которым он принадлежит. Каждый раз при обращении пользователя к объекту его права проверяются по списку разрешений данного объекта.
ext2— файловая система ядра Linux. По скорости и производительности работы она может служить эталоном в тестах производительности файловых систем.
Главный недостаток ext2 (и одна из причин демонстрации столь высокой производительности) заключается в том, что она не является журналируемой файловой системой.
Файловая система ext2 по-прежнему используется на флеш-картах и твердотельных накопителях (SSD), так как отсутствие журналирования является преимуществом при работе с накопителями, имеющими ограничение на количество циклов записи.
Атрибуты: 1) тип и права доступа к файлу,2)владелец, группа доступа,3)информация о разрешённых операциях, 4)время создания, дата последнего доступа, дата последнего изменения и время последнего удаления,5)текущий размер файла, и др.
Атрибуты файлов хранятся не в каталогах, как это сделано в ряде простых файловых систем, а в специальных таблицах. В результате каталог имеет очень простую структуру, состоящую всего из двух частей: номера индексного дескриптора и имени файла.
ext3— журналируемая файловая система, используемая в операционных системах на ядре Linux, является файловой системой по умолчанию во многих дистрибутивах. Основана на ФС ext2.Основное отличие от ext2 состоит в том, что ext3 журналируема, то есть в ней предусмотрена запись некоторых данных, позволяющих восстановить файловую систему при сбоях в работе компьютера.

13 Структура ОС и осн. понятия ОС. Реализ. Ос на примере МС ДОС.Прерывание в ОС

любая ОС состоит из двух основных частей: ядра и нерезидентной части, которая содержит процедуры, нечасто используемые при работе ОС. Ядро содержит загрузчик ОС и набор процедур, управляющих теми или иными функциями ОС, а именно:

- подсистему управления процессов (диспетчер).
- подсистему управления основной памяти.
- подсистему управления виртуальной памяти (если она есть).

ОС содержит совокупность резидентных драйверов: драйвер диска, мыши, клавиатуры и др. и командные процессоры-интерпретаторы входного языка ОС, который служит для обработки команд ОС и выдачи сообщений.

Кроме того, в состав ядра входят подсистемы управления ввода – вывода (BIOS для DOS).

Для ОС отводится фиксированная область памяти, защищенная от несанкционированного доступа. В этой области располагается ядро и системная область памяти, где располагаются таблицы ОС, структуры данных ОС, и, возможно, системные библиотеки, которые содержат макроопределения примитивов ОС.

Примитивом называется процедура, реализующая ту или иную элементарную функцию ОС (создание буферов ввода-вывода, просмотр буфера ввода и т. д.).

В состав системных библиотек обычно входят команды ОС: внутренние и внешние.

Любая ОС имеет два файла:1) файл конфигурации системы (config.sys для DOS). Он определяет конкретную на данный момент конфигурацию ОС и ее параметры.2) Файл автозапуска (autoexec.bat для DOS). Он предназначен для формирования удобной для пользователя операционной среды.

Пример: MS –DOS v. 6.22.

1) BIOS – (базовая система ввода – вывода)

Микросхема, которая содержит все подпрограммы, соответствующие примитивам ОС. Обычно 64 килобайта.

команда int <номер прерывания>.

BIOS управляется специальными векторами прерываний, которые представляют собой ячейки из четырех байт, расположенных в начале оперативной памяти.

Вектора прерываний содержат адрес соответствующей программы обработки прерываний, которые заставляют ядро ОС выполнить ту/иную функцию.

2) _IO.SYS – (система ввода – вывода);

3) _DOS.SYS – содержит основные подпрограммы управления; в частности, примитивы, записанные в этом файле, аналогичны процедурам BIOS;

4) COMMAND.COM – командный процессор;

5) CONFIG.SYS – файл конфигурации;

6) AUTOEXEC.BAT

Прерывание – это событие внутри системы, которое связано с приостановкой работы центрального процессора, с запоминанием его состояния, с передачей управления программе обработки данного прерывания и возврату к прежнему состоянию процесса.

4 Вида прерываний

1) SVS(SuperViSor) – они связаны с переходом всей вычислительной системы в состояние «система» и вызовом супервизора или диспетчера процессов. В состоянии «система» центральный процессор в состоянии выполнения привилегированной команды.

Состав: менеджеры

- ОП;
- Виртуальной памяти;
- Ресурсов;
- Вспомогательной внешней памяти

2) Программные прерывания (break'и деление на ноль, выполнение несуществующей команды и др.).

3) Прерывания по таймеру - привилегированная команда, приостанавливающая работу центрального процессора, вводит некое число в регистр таймера, и он сбрасывает по единице до нуля. Это дает возможность выделять кванты времени для программ пользователя.

4) Прерывания ввода–вывода – инициируются каналами или устройствами ввода-вывода.

Все прерывания характеризуются приоритетом:

Системные прерывания – самые важные;

Прерывания ПО и ввода–вывода – самые неважные.
В системной области ОС для каждого типа прерываний существует своя рабочая область – область прерываний, где хранится информация о них, а состояние центрального процессора записывает в слово состояния PSW
Слово состояния представляет собой структуру, которая содержит следующие поля:
- Режим (0 – пользователь или 1 – супервизор)
- Активность (0 – центральный процессор активен; 1 – выполняются привилегированные команды или ничего не выполняет)
- Процесс; идентификатор данного процесса PID.
- Условие; код условия данного процесса.
- Маска; используется для контроля за разрешением прерывания. Она устанавливается ОС в соответствии с классом прерывания так, чтобы все прерывания равного или более низкого класса были запрещены, а с более высоким приоритетом были разрешены.
Все прерывания в ОС образуют очереди, которые обрабатываются в ОС по мере завершения тех или иных прерываний.
Прерывания могут быть вложенными.
В ОС обработкой любого прерывания занимаются обработчики прерываний.
Прерывания используют для реализации таких функций ОС, как планирование процессов, функций ввода-вывода, функций распределения памяти.

14 Графы распределения ресурсов при решении задач обнаружения тупиков. Редукция графов

Графы распределения ресурсов

При рассмотрении задачи обнаружения тупиков применяется распространенная операция, согласно которой распределение ресурсов и запросы изображаются в виде направленного графа.

- процессы

- классы идентичных ресурсов

- идентичные ресурсы каждого класса

Пример:

а) Процесс P запрашивает ресурс типа R1. Стрелка только до большого кружка показывает, что в текущий момент запрос от процесса находится в состоянии рассмотрения.

б) P2 выдается только один из идентичных ресурсов.

в) Ситуация приближена к тупику, т.к. P3 требует выданный P4 ресурс.

г) Представлен пример кругового ожидания.

Графы запросов и распределения ресурсов меняются по мере того, как процессы запрашивают ресурсы, получают их, а затем возвращают их ОС.

Редукция графов распределения ресурсов

Один из способов обнаружения тупиков – приведение (редукция) графа. Это позволяет определить процессы, которые могут завершиться и процессы, которые будут оставаться в тупиковой ситуации. Если запросы ресурсов для некоторого процесса могут быть удовлетворены, то граф можно редуцировать на этот процесс. Такая редукция эквивалентна изображению графа в том виде, который он будет иметь, если процесс завершится и возвратит ресурсы системе. Редукция графа на конкретный процесс изображается исключением стрелок, т.е. текущего запроса на выделение ресурса.

Вывод: если граф можно редуцировать на все процессы, то это значит, что тупиковой ситуации нет.

15 ОС Linux. Осн-е команды работы с файлами (cat, echo, Ls, cp,rm, more)

1. Показ содержания файлов.

Показать содержание файла:

cat < имя файла >.

Показать содержания файлов:

cat < имена файлов через пробел >.

Добавить содержание файла1 в конец файла2(символ >>):

cat >> < файл2 > < файл1 >.

Заменить содержимое файла1 содержанием файла2(символ >):

cat > < файл1 > < файл2 >.

2.Команда echo.

Отображает на мониторе введённые данные.

echo " выражение ".

3.Создание перечней файлов (каталогов).

Команда ls.

Создание перечней файлов по маске: ls < маска >.

Используются два символа : * ?

Пример: ls *i* -показ файлов, в наименовании которой присутствует буква i .

В синтаксисе этой команды можно присутствовать < путь > ,а также специальные символы:

.. -- заменяет название предыдущего каталога (две точки).

. -- заменяет название текущего каталога (одна точка).

ls -l < маска > - вывод в длинном полном формате.

ls -l /usr/lib | more - постраничный вывод списка файлов в полном формате.

4.Переименование файлов.

Команда mv (сокр. от англ. to move).

Переименовать файл1 в файл2: mv < файл1 > < файл2 >.

5.Копирование файлов.

Команда cp (сокр. от англ. to copy).

Скопировать файл1 в файл2: cp < файл1 > < файл2 >.

Если файл2 существует, то при переименовании и копировании его содержание удаляется.

6. Удаление файлов.

Команда rm (сокр. от англ. to remove).

rm [-i] < имена файлов через пробел >.

При этом выдаётся запрос на подтверждение удаления.

Ответ:Y or N.

7.Создание файлов.

Создать файл1 и записать в него выражение:

echo > <файл1> " выражение".

cat –u > <имя файла>

vi <имя файла>

8. Постраничный показ файлов.

Команда more.

more < файл >

Если текст не помещается на экране полностью, то он показывается постранично , используются следующие команды:

h -Краткий текст подсказки;

пробел -Следующая страница текста ;

Enter -Следующая строка текста ;

Q -Закончить работу команды;

b -На одну страницу назад;

11.Поиск файлов.Команда find просматривает каждый из каталогов, перечисленных в списке поиска, отыскивая файлы, удовлетворяющие логическому выражению.

find <список поиска> <выражение>

16 ОС Linux.Права доступа. Владельцы, группы, прочие. Установка и изменение прав доступа

Правильная настройка прав доступа позволяет повысить надёжность системы, защитив от изменения или удаления важные системные файлы.

У любого файла в системе есть владелец — один из пользователей. Однако каждый файл одновременно принадлежит и некоторой группе пользователей системы. Каждый пользователь может входить в любое количество групп, и в каждую группу может входить любое количество пользователей из числа определённых в системе.

Когда в системе создаётся новый пользователь, он добавляется по крайней мере в одну группу. В системе Linux при создании новой учётной записи создаётся специальная группа, имя которой совпадает с именем нового пользователя, и пользователь включается в эту группу. В дальнейшем администратор может добавить пользователя к другим группам.

Механизм групп может применяться для организации совместного доступа нескольких пользователей к определённым ресурсам. Например, на сервере организации для каждого проекта может быть создана отдельная группа, в которую войдут учётные записи (имена пользователей) сотрудников, работающих над этим проектом. При этом файлы, относящиеся к проекту, могут принадлежать этой группе и быть доступными для её членов.

Права доступа определяются по отношению к трём типам действий: чтение, запись и исполнение. Эти права доступа могут быть предоставлены трём классам пользователей: владельцу файла (пользователю), группе, которой принадлежит файл, а также всем остальным пользователям, не входящим в эту группу. Право на чтение даёт пользователю возможность читать содержимое файла или, если такой доступ разрешён к папке, просматривать её содержимое. Право на запись даёт пользователю возможность записывать или изменять файл, а право на запись для папки — возможность создавать новые файлы или удалять файлы из этой папки. Наконец, право на исполнение позволяет пользователю запускать файл как программу или сценарий командной оболочки (разумеется, это действие имеет смысл лишь в том случае, если файл является программой или сценарием). Для папок право на исполнение имеет особый смысл — оно позволяет сделать данную папку текущей, т. е. “перейти” в неё, например, командой cd.

Чтобы получить информацию о правах доступа к файлам и папкам в консоли, используйте команду ls с ключом -l. При этом будет выведена подробная информация о файлах и папках, в которой будут, среди прочего, отражены права доступа

chmod - изменение прав доступа к файлу или папке

chown - изменение владельца файла

chgroup - изменение группы, которой принадлежит файл

umask - определение прав доступа по умолчанию для файлов, создаваемых пользователем

Изменение прав доступа

Команда chmod используется для установки (изменения) прав доступа файла. Только владелец файла может менять права доступа к нему.

Синтаксис команды имеет вид:

chmod {a,u,g,o}{+,-}{r,w,x} <имя файла>

Кратко, вы выбираете из all (все), user (пользователь), group (группа) или other (другие). Далее указываете, либо вы добавляете права (+), либо лишаете прав (-). И наконец, вы указываете один или несколько режимов: read, write или execute. Несколько примеров допустимых команд:

chmod a+r user.txt

Дает всем пользователям право читать файл user.txt.

chmod +r user.txt

То же самое, что и ранее (a - по умолчанию).

chmod og-x user.txt

Лишает права на выполнение всех, кроме владельца.

chmod u+rwx user.txt

Разрешает владельцу все (read, write и execute).

chmod o-rwx user.txt

Запрещает все (read, write и execute) пользователям категории другие (other).

17 ОС Linux. Файловые систмы. Создания ФС и их монтирование.

ФС – логическая структура данных, определяющая способ хранения файлов и их заголовков. Состоит из 2х частей: Данные файлов и заголовки. В Linux для работы с файловыми системами, находящимися на разных носителях информации, необходимо их сначала объединить в одну файловую систему. Процесс объединения файловых систем называется монтированием.

Принцип монтирования:

Существуют различные файловые системы:

XENIX

MINIX. Служила первоосновой для ext и ext2, так как первые версии Linux были разработаны на базе MINIX.

ext (Extended, расширенная).

ext2 (Second Extended, вторая расширенная). Используется в большинстве современных дистрибутивах Linux.

FAT (Используется MS-DOS и MS Windows),NTFS (Поставляется вместе с Windows NT),ISO9660 (Используется для CD-ROM).NFS (Сетевая файловая система).

Создание файловой системы

Команда mkfs создает новую файловую систему. Она расположена в каталоге /etc и имеет три параметра: mkfs < -t тип > <имя> <размер>.

Первый параметр – тип создаваемой файловой системы

Примеры файловых систем, поддерживаемые Linux:

minix – Файловая система Minix.

ext2 – Наиболее часто используемая файловая система в Linux.

msdos – MS-DOS.

Второй параметр является именем специального файла и указывает устройство, на котором создается файловая система.

Третий параметр – размер пространства файловой системы в блоках.

Чтобы использовать mkfs, надо иметь право на запись в файлы устройств, на которые монтируется файловая система. Пример: mkfs -t minix dev/fd0 1440

Монтирование файловой системы

Любая новая файловая система интегрируется в существующую систему каталогов. Для этого служит пустой каталог (точка монтирования – mount point), на место которого монтируется новая файловая система. Любой последующий доступ к каталогу будет доступом к только что смонтированной файловой системе.

Команда mount.

mount <опции> < -t тип > <устройство> <каталог>.

<устройство> - имя спецфайла для монтируемого устройства, содержащего подключаемую файловую систему.

<каталог> - точка монтирования.

< -t тип > - устанавливаемая файловая система принадлежит указанному типу.

С т.з. пользователя существует 1 ФС, реально состоит из любого количества примонтированных к каталогам ФС. Системы монтируются в каталог /mnt/.

Примеры возможных опций:

-r данные можно только считывать (read only).

-w данные можно считывать, изменять и удалять (read/write).

Чтобы выяснить, какие файловые системы смонтированы в данный момент, надо подать команду mount без параметров .

Пример монтажа вновь созданной файловой системы на гибком диске под каталогом, созданным командой mkdir в корне корневой файловой системы:

```
$ cd /
$ mkdir floppy
$ mount –t minix -rw /dev/fd0  /floppy
```

Демонтирование файловой системы

Команда unmount. unmount <опции> < -t тип > <устройство или каталог>.

-a - демонтирует все имеющиеся файловые системы.

< -t тип > - демонтируемая файловая система принадлежит указанному типу.

Пример:

unmount -a .

18. Ос Linux Интерпретатор команд Shell. Ввод и вывод команд. Стандартный и нестандартный ввод и вывод. Конвейер команд.

ОС Linux

История Linux

Linux построен на основных идеях Unix, но Linux – это отдельная ОС. Она создана людьми всего мира с помощью Интернета. Идея зародилась в 1990 г. В Хельсинки. Шведский студент Linus Torvalds написал диплом по ядру Unix-подобной системы. Оно было настолько удачно, что у него зародилась идея написания бесплатной Unix-подобной системы.

Дистрибутивы Linux

Существуют готовые комплексные продукты, содержащие ОС Linux и различные комплекты программного обеспечения. Эти продукты ориентированы на разных потребителей.

Наиболее распространенные дистрибутивы:

- RedHat – самый популярный; превратился в коммерческую структуру.
- SuSe – похож на RedHat, но больше пакетов. Сложна в настройке.
- Mandrake – переделка RedHat под Pentium с KDE.
- Slackware – упрощена установка: вместо командной строки графический интерфейс. Это традиционный дистрибутив Linux.

Инсталляция Linux

Загрузиться с CD и ответить на вопросы, после чего инсталлятор скопирует указанные пакеты.

Интерфейс

1981 – 1-й компьютер с графическим интерфейсом пользователя.

1984 – 1-й интерфейс GUI (Graphic User Interface) для Macintosh. Это помогало лучше адаптироваться к новой ОС.

В ней содержаться протоколы для отображения графических объектов и работы с сетью.

Диспетчер окон

1996 – KDE проект. Основан на использовании графических библиотек. Достоинства: все приложения в одном стиле, одинаковые меню, простота использования.

Gnome – поддерживает GPL-стандарт (General Public License). Предназначен для использования ПО, сделанного по правилам GNU.

Свойство рабочего стола: поддержка региональных настроек(для популяризации)

Командный интерпретатор Shell

Интерпретатор команд shell (оболочка) обеспечивает интерфейс пользователь-ядро при введении пользователем команд с терминала. Интерпретирует также команды, вводимые из командного файла.

Большинство современных shell обладают рядом дополнительных возможностей: удобное редактирование командной строки, история вводимых команд и т.д. В Linux известно большое количество shell’ов.

Самыми популярными являются:

bash - Bourne Again Shell,
csh - C Shell,
ksh - Korn Shell.

Аналогом shell в операционной системе MS-DOS является COMMAND.COM.

Оболочка Bourne Again Shell запускается командой bash, C Shell командой csh, Korn Shell командой ksh.

Когда пользователь входит в систему, указывая свой пароль, автоматически запускается shell.

Большинство оболочек закрываются путем ввода спец. команд *logout, exit, bye* или *quit* (одна из них действует всегда). При выполнении команды система возвращается к исходному состоянию, т.е. выдается запрос имени пользователя и пароля.

Ввод и вывод команд

Использование команд, введенных ранее

Командная оболочка сохраняет команды, которые вы набирали ранее, в так называемой *истории команд* (command history), в файле с названием .bash_history, который расположен в домашнем каталоге. Для извлечения команд из *истории команд*, следует использовать клавиши ↑(вверх) и ↓(вниз).

Командная оболочка bash может сохранить до 1000 команд.

Стандартные ввод и вывод

Каждая команда получает в свое распоряжение от shell три канала:

Стандартный ввод *(stdin*). По этому каналу вводятся данные в программу.

Стандартный вывод *(stdout).* По этому каналу выводятся данные.

Стандартный вывод ошибок *(stderr).* По этому каналу программы выдают информацию и наличие ошибок.

Обычно все эти каналы связаны с монитором и клавиатурой.

Изменение направления ввода и вывода данных

Согласно принципам работы Linux, стандартный ввод можно переназначить. Например: *prog* < *file*. Символ < говорит, что стандартный ввод для программы *prog* переназначается, и данные будут браться не с клавиатуры терминала, а из файла *file.*

Переназначение стандартного вывода: *prog* > *file* .Т. е. выводимые данные перенаправляются в файл *file*. При этом исходное содержание файла *file* заменяется.

Конвейеры команд

Стандартный вывод одной команды может быть стандартным вводом следующей команды. Обозначается знаком | (вертикальная черта). Последовательность команд, соединённых таким образом, называется конвейером.

19. ОС Linux. Стандартная структура каталогов. Учетные записи пользователей.

В последнее время оживленно обсуждается введение Linux-стандартов, которые позволят упростить использование Linux и разработку программного обеспечения. Но пока до этого не дошло, разберем наиболее распространенный вариант.

В файловой системе Linux, независимо от сложности физической конфигурации, есть одна общая точка — точка монтирования файловой системы. Она представляет собой корневой каталог и обозначается символом «/». К этой точке монтируются все остальные каталоги, независимо от типа содержащихся в них файлов и распределения по физическим устройствам (винчестерам, разделам).

ТОЧКА МОНТИРОВАНИЯ

Точка монтирования (mount point) — это каталог, в котором будет представлена файловая система после получения пользователем разрешения на доступ к ней.

Таким образом, все ваши разделы на всех винчестерах, даже содержащие, например, операционную систему MS DOS/Windows, будут представлены в файловой системе Linux как катало!" с подкаталогами. Причем корневой каталог чужой файловой системы может быть смонтирован в любой точке (каталоге) файловой системы Linux.

Разбиение файловой системы Linux на разделы — не только принятая практика, по и важный способ защиты информации. Блоки информации разделяются не только логически, но и физически. Например, наиболее ценные данные располагаются не только в отдельном каталоге, по и в отдельном разделе на винчестере пли даже па другом носителе информации (другом винчестере).

Добавляет проблем и то, что при установке дистрибутива каждый пользователь может по-своему распределить файловую систему по физическим разделам и дискам, а в дальнейшем переименовать каталоги, создать новые точки монтирования и многое-многое другое. Фактически, нет одинаковых по структуре файловых систем — у каждого пользователя она своя.

Несмотря на такое разнообразие структур файловых систем Linux, при изучении этой операционной системы можно ориентироваться на следующую структуру, которую можно признать типичной;:

/ — корневой каталог, от которого идет отсчет пути ко всем остальным файлам и каталогам.

/bin -- основные системные программы (файлы в двоичном коде).

/boot — файлы ядра, которые используются при загрузке операционной системы.

/dev — специальные файлы — файлы устройств, предназначенные для взаимодействия с системными и физическими устройствами.

/etc — конфигурационные файлы системы и подкаталоги с конфигурационными файлами прикладных программ.

/home — каталог, в котором создаются домашние каталоги всех обычцых пользователей.

/lib — библиотеки общих кодов для программ.

/lost+found - каталог предназначен для сохранения информации об удаленных файлах.

/mnt — здесь размещаются каталоги (точки монтирования) внешних файловых систем, таких как гибкий диск, CD-ROM (как бы «стандартизованные» точки монтирования, но не во всех случаях).

/root — домашний каталог администратора (супергюльзователя, root).

/sbin — двоичные файлы программ, применяемых в основном для администрирования системы.

/usr — каталоги, файлы прикладных программ и пакетов. Этот каталог содержит разветвленную систему подкаталогов и наибольший объем файлов.

/usr/man и /usr/doc — документация по системе. /usr/Xll — файлы X Window.

/var — рабочие и журнальные файлы, которые имеют тенденцию менять свой размер в течение времени (аналог — временные файлы в Windows).

Учётные записи

Linux — система многопользовательская, а потому пользователь — ключевое понятие для организации всей системы доступа в Linux. Когда пользователь регистрируется в системе (проходит процедуру авторизации, например, вводя системное имя и пароль), он идентифицируется с учётной записью, в которой система хранит информацию о каждом пользователе: его системное имя и некоторые другие сведения, необходимые для работы с ним. Именно с учётными записями, а не с самими пользователями, и работает система. Ниже приведён список этих сведений.

Системное имя (user name)

Это то имя, которое вводит пользователь в ответ на приглашение login:. Оно может содержать только латинские буквы и знак “_”. Это имя используется также в качестве имени учётной записи.

Идентификатор пользователя (UID)

Linux связывает системное имя с идентификатором пользователя в системе — UID (User ID). UID — это положительное целое число, по которому система и отслеживает пользователей1. Обычно это число выбирается автоматически при регистрации учётной записи, однако оно не может быть совершенно произвольным. В Linux есть некоторые соглашения относительно того, каким типам пользователей могут быть выданы идентификаторы из того или иного диапазона. В частности, UID от “0” до “100” зарезервированы для псевдопользователей2.

Идентификатор группы (GID)

Кроме идентификационного номера пользователя с учётной записью связан идентификатор группы. Группы пользователей применяются для организации доступа нескольких пользователей к некоторым ресурсам. У группы, так же, как и у пользователя, есть имя и идентификационный номер — GID (Group ID). В Linux каждый пользователь должен принадлежать как минимум к одной группе — группе по умолчанию. При создании учётной записи пользователя обычно создаётся и группа, имя которой совпадает с системным именем3, именно эта группа будет использоваться как группа по умолчанию для этого пользователя. Пользователь может входить

более чем в одну группу, но в учётной записи указывается только номер группы по умолчанию. Группы позволяют регулировать доступ нескольких пользователей к различным ресурсам.

Полное имя (full name)

Помимо системного имени в учётной записи содержится и полное имя (имя и фамилия) использующего данную учётную запись человека. Конечно, пользователь может указать что угодно в качестве своего имени и фамилии. Полное имя необходимо не столько системе, сколько людям — чтобы иметь возможность определить, кому принадлежит учётная запись.

Домашний каталог (home directory)

Файлы всех пользователей в Linux хранятся раздельно, у каждого пользователя есть собственный домашний каталог, в котором он может хранить свои данные. Доступ других пользователей к домашнему каталогу пользователя может быть ограничен. Информация о домашнем каталоге обязательно должна присутствовать в учётной записи, потому что именно с него начинает работу пользователь, зарегистрировавшийся в системе.

Начальная оболочка (login shell)

Важнейший способ взаимодействовать с системой Linux — командная строка, которая позволяет пользователю вести «диалог» с системой: передавать ей команды и получать её ответы. Для этой цели служит специальная программа — командная оболочка (или интерпретатор командной строки), по-английски — shell. Начальная оболочка (login shell) запускается при входе пользователя в систему в текстовом режиме (например, на виртуальной консоли). Поскольку в Linux доступно несколько разных командных оболочек, в учётной записи указано, какую из командных оболочек нужно запустить для данного пользователя. Если специально не указывать начальную оболочку при создании учётной записи, она будет назначена по умолчанию, вероятнее всего это будет bash.

Все перечисленные данные об учётных записях хранятся в файле /etc/passwd. Сведения о конкретной учётной записи пользователя можно получить с помощью утилиты getent

20 ОС Linux. Работа с файлами каталогами. Основные команды работы с каталогами (mkdir,ls,mv,rmdir,cd).

Команда mkdir

Команда mkdir позволяет создать подкаталог в текущем каталоге. В качестве аргумента этой команде надо дать имя создаваемого каталога. Во вновь созданном каталоге автоматически создаются две записи: . (ссылка на этот самый каталог) и .. (ссылка на родительский каталог). Чтобы создать подкаталог, вы должны иметь в текущем каталоге право записи. Можно создать подкаталог не в текущем, а в каком-то другом каталоге, но тогда необходимо указать путь к создаваемому каталогу:

[user]$ mkdir /home/kos/book/glava5/part1
Команда mkdir может использоваться со следующими опциями:
-m mode — задает режим доступа для нового каталога (например, -m 755);
-p — создавать указанные промежуточные каталоги (если они не существуют).

Команда mv

Если вам необходимо не скопировать, а переместить файл из одного каталога в другой, вы можете воспользоваться командой mv. Синтаксис этой команды аналогичен синтаксису команды cp. Более того, она сначала копирует файл (или каталог), а только потом удаляет исходный файл (каталог). И опции у нее такие же, как у cp.
Команда mv может использоваться не только для перемещения, но и для переименования файлов и каталогов (т. е. перемещения их внутри одного каталога). Для этого надо просто задать в качестве аргументов старое и новое имя файла:
[user]$ mv oldname newname
Но учтите, что команда mv не позволяет переименовать сразу несколько файлов (используя шаблон имени), так что команда mv *.xxx *.yyy не будет работать.
При использовании команды mv, также как и при использовании cp, не забывайте применять опцию -i для того, чтобы получить предупреждение, когда файл будет перезаписываться.

Команды rm и rmdir

Для удаления ненужных файлов и каталогов в Linux служат команды rm (удаляет файлы) и rmdir (удаляет пустой каталог) . Для того, чтобы воспользовался этими командами, вы должны иметь право записи в каталоге, в котором расположены удаляемые файлы или каталоги. При этом полномочия на изменение самих файлов не обязательны. Если хотите перед удалением файла получить дополнительный запрос на подтверждение операции, используйте опцию -i.
Если вы попытаетесь использовать команду rm (без всяких опций) для удаления каталога, то будет выдано сообщение, что это каталог, и удаления не произойдет. Для удаления каталога надо удалить в нем все файлы, после чего удалить сам каталог с помощью команды rmdir. Однако можно удалить и непустой каталог со всеми входящими в него подкаталогами и файлами, если использовать команду rm с опцией -r.
Если вы дадите команду rm *, то удалите все файлы в текущем каталоге. Подкаталоги при этом не удалятся. Для удаления как файлов, так и подкаталогов текущего каталога надо тоже воспользоваться опцией -r. Однако всегда помните, что в Linux нет команды восстановления файлов после их удаления (даже если вы спохватились сразу же после ошибочного удаления файла или каталога)!

ls

Это очень часто используемая команда. Она перенаправляет содержимое каталога в стандартный поток вывода. Если выполнить эту команду не указывая целевой каталог, то выведется содержимое текущего каталога.

```
adam@laptop:~/Documents/polishlinux.org$ pwd
 /home/adam/Documents/polishlinux.org/
 adam@laptop:~/Documents/polishlinux.org$ls
 example.txt all_about_console.txt
 adam@laptop:~/Documents/polishlinux.org$ls /var/
 backups cache crash games lib local lock log mail opt run spool
 tmp
```

Если после команды ls поставить ключ -l, то выведется более подробная информация о содержимом каталога.

```
adam@laptop:~/Documents/polishlinux.org/examples$ ls -l
 total 0
 -rw-r–r– 1 adam adam 0 2007-05-30 11:31 example.txt
 -rw-r–r– 1 adam adam 0 2007-05-30 11:31 all_about_console.txt
```

В первой строке показывается общее количество блоков на диске занимаемое файлами каталога. Остальные строчки описывают фалы в следующем формате (по столбцам):

* -rw-r–r– - уровни доступа к файлам и каталогам
 * количество жестких ссылок на файл
 * пользователь и группа владельцев файла
 * размер файла
 * время последнего изменения файла
 * имя файла или каталога

Команда ls с ключом -a выводит так же и скрытые файлы. Имена скрытых файлов начинаются с точки.

```
adam@laptop:~/Documents/polishlinux.org/examples$ ls -a
 . .. .hidden_file example.txt all_about_console.txt
```

По мимо этого, доступна возможность сортировки файлов командой ls со следующими ключами:

* strong>-t - по времени (по убыванию)
 * -S - по размеру (по убыванию)
 * -r - обратный порядок (-tr)

Если запустить с ключом -R, то выведется все содержимое каталога и его подкаталогов (рекурсивный обход):

```
adam@laptop:~/Documents/polishlinux.org/examples$ ls -R
 .:
 directory1 directory2 example.txt all_about_console.txt
```

./directory1:
file1 file2
./directory2:
file3 file4

2. cd

С помощью команды cd (change directory) мы можем перемещаться по дереву каталогов. Для этой команды необходим один параметр - это целевой каталог (относительно текущего или относительно корня).

adam@laptop:~$cd /usr/bin
adam@laptop:/usr/bin$pwd
/usr/bin

Если вместо пути поставить символ тильда (~), то вас перекинут в домашнюю папку.

adam@laptop:/usr/bin$cd ~
adam@laptop:~$pwd
/home/adam

Если ввести ключ в виде ~имя_пользователя, то нас перекинут в домашний каталог этого пользователя.

adam@laptop:~$cd ~zoidberg
adam@laptop:/home/zoidberg$pwd
/home/zoidberg

Любой каталог к UNIX всегда содержит каталог “.” и “..”. Одна точка “ссылка” (не жесткая, и не символическая) на текущий каталог, а две точки - это “ссылка” на родительский каталог.

adam@laptop:~$cd ..
adam@laptop:/home$pwd
/home

Как уже говорилось выше, можно использовать как относительный путь, так и абсолютный. Абсолютный всегда начинается с корня и дальше по иерархии каталогов (/home/adam), а относительный - это относительно текущего каталога, начинается сразу с его имени.

21 ОС Linux. Установка прав доступа символьным и числовым способом.

Права доступа

Для файлов и каталогов в linux существуют 3 уровня прав доступа, соответствующими 3-м категориям: владелец, группа, прочие.

Каждый уровень имеет свои права: чтение, запись, выполнение.

Чтобы реализовать концепцию прав доступа, надо каждому файлу сопоставить владельца.

Изменение прав доступа

`$chmod`–команда для изменения прав доступа к файлу.Для надо указать 3 парам.:

1) изменяемый уровень (владелец/группа/другие);
2) изменяемые права доступа;
3) об'екты, к которым применяются эти изменения.

опция	права
u	пользователь(владелец файла)
g	группа
o	прочие
a	все
право	описания
r	чтение
w	запись
x	выполнение
t	стики бит (Sticky bit) используется вместе с другими битами для установки прав доступа к каталогу. Если он установлен, то в этом каталоге только владелец файла может его удалить (используется для /tmp)

Уровни и права доступа используются в комбинации с + или – чтобы установить или сбросить их:

Между уровнем и режимом доступа используются символы:

«+» - добавляет право доступа, не оказывая влияния на другие права;

«–» - удаляет право доступа;

«=» - делает указанный режим доступа единственным.

Для изменения режима доступа к файлу вместо символьных параметров могут использоваться числа:

3^{x} и 4^{x} разрядные числа в восьмеричной системе.

Последние 3 разряда относятся к правам для u g o. Если $1^{й}$ разряд=0, то он не пишется.

$1^{й}$ разряд имеет следующие параметры:

0 – нет дополнительных прав доступа;

1 – устанавливает Sticky bit, вместо x ставят t в символьных параметрах:rwt

Изменение пользователя или группы, которые владеют файлом

Ключ к концепции прав доступа – у файла есть как владелец, так и группа. С помощью утилиты `$ chown` можно изменить владельца,

`$ chgrp` – изменить группу

Формат команд: $ <команда> <имя нового владельца/группы> <имя файла>

Изменять владельца может только суперпользователь (администратор). Примеры команд:

```
$ chown natalie testfile
$ ls –s
-rwxr--r-- 1 natalie users <дата> testfile
…
$ chgrp eng testfile
$ ls –s
-rwxr--r-- 1 natalie eng <дата> testfile
```

22. ОС Windows. Области применения. Версии. Перспективы развития.

Microsoft Windows (/ˈwɪndoʊz/) (англ. windows — о́кна) — семейство проприетарных операционных систем корпорации Майкрософт (Microsoft), ориентированных на применение графического интерфейса при управлении. Изначально были всего лишь графическими надстройками для MS-DOS.

В настоящее время под управлением операционных систем семейства Windows, по данным ресурса Netmarketshare (Net Applications) на 2009 год, работает около 89 % персональных компьютеров[1].

Версии:

Windows NT 3.1 (1993)
Windows NT 3.5 (1994)
Windows NT 3.51 (1995)
Windows NT 4.0 (1996)
Windows 2000 (2000) — Windows NT 5.0
Windows XP (2001) — Windows NT 5.1
Windows XP 64-bit Edition (2006) — Windows NT 5.2
Windows Server 2003 (2003) — Windows NT 5.2
Windows Vista (2006) — Windows NT 6.0
Windows Home Server (2007) — Windows NT 5.2
Windows Server 2008 (2008) — Windows NT 6.0
Windows Small Business Server (2008) — Windows NT 6.0
Windows 7 — Windows NT 6.1 (2009)
Windows Server 2008 R2 — Windows NT 6.1 (2009)
Windows Home Server 2011 — Windows NT 6.1 (2011)
Windows 8 — Windows NT 6.2 (2012)

23. ОС Windows Mobile для карманных ПК и смартфонов (платформы, версии, возможности).

Windows Mobile (также известна как Windows Phone для ветки 6.5.x) — мобильная операционная система, разработанная Microsoft для собственных аппаратных платформ Pocket PC (коммуникатор) и Smartphone. В настоящее время переживает постепенный отказ от поддержки и разработки.

Handheld PC 1.0

Handheld PC 2.0

Pocket PC 2000

Pocket PC 2002

Windows Mobile 2003

Windows Mobile 5.0

Windows Mobile 6

Windows Mobile 6.1

Windows Mobile 6.1.4

Windows Mobile 6.5

Windows Mobile 6.5.1

Windows Mobile 6.5.3

Windows Mobile 7

Pocket PC 2000 (Windows CE 3.0 Pocket PC Edition)

Pocket PC 2000 (первая версия Windows Mobile), кодовое имя Rapier. Выпущена 19 апреля 2000 года на основе Windows CE 3.0. Это был дебют, впоследствии операционную систему назвали Windows Mobile. Также она призвана быть преемником операционной системы Palm-Size ПК. Сохранена обратная совместимость с приложениями Palm-Size PC. Pocket PC 2000 предназначен в основном для КПК, однако для нескольких устройств Palm-Size PC есть возможность обновления. Кроме того, для Pocket PC 2000 были выпущены несколько телефонов, однако Smartphone Edition еще не была создана. Поддерживалось только разрешение 240×320 (QVGA). Также поддерживались съемные карты памяти CompactFlash и Multimedia Card. В то время для устройств Pocket PC не была определена конкретная архитектура процессора. В результате Pocket PC 2000 была выпущена на нескольких архитектурах: SH-3, MIPS и ARM.

Первоначально Pocket PC была похожа на операционные системы Windows 98, Windows Me и Windows 2000.

Особенности / встроенные приложения для Pocket PC 2000, включали следующее:

Pocket Office

Pocket Word

Pocket Excel

Pocket Outlook

Pocket Internet Explorer

Windows Media Player

Microsoft Reader

Microsoft Money

Заметки «от руки»

Поддержка распознавания символов

Инфракрасный (ИК) порт

Pocket PC 2002

Pocket PC 2002 (кодовое имя Merlin), основанная на Windows CE 3.0, это первый выпуск под названием Pocket PC. Нацелена на бесклавиатурные Pocket PC устройства с разрешением QVGA (320×240). Впервые появляется поддержка устройств без сенсорного экрана, что рождает появление первых устройств типа «Смартфон» на Windows Mobile.

Windows Mobile 2003 (Pocket PC 2003)

Вторая версия (кодовое имя Ozone), основанная на Windows CE 4.20, была выпущена 23 июня 2003 года и шла в трёх редакциях. Первые две из них довольно похожи: Pocket PC 2003 for Pocket PC и Pocket PC 2003 Phone Edition.

Данная версия работает значительно быстрее предыдущих.

Третья редакция называлась Pocket PC 2003 for Smartphone. По сути, это несколько иная (в отличие от Pocket PC) платформа, требующая специальной доработки приложений под неё. Наиболее существенные отличия в том, что экран устройства нечувствителен к нажатию, разрешение экрана меньше, клавиатура — стандартная для мобильных телефонов.

Windows Mobile 2003 SE (Pocket PC 2003 Second Edition)

Pocket PC 2003 Second Edition или Pocket PC 2003SE выпущена 24 марта 2004 года и впервые предложена на устройстве Dell Axim x30. Список отличий от предыдущих версий включает:

Возможность изменения ориентации экрана с вертикальной на горизонтальную.

Помимо разрешения QVGA (320×240) также поддерживается разрешение VGA (640×480) и квадратные разрешения (240×240 и 480×480), что хорошо подходит для устройств, имеющих встроенную клавиатуру.

Поддержка протокола WPA (Wi-Fi Protected Access)

Windows Mobile 2003SE основана на Windows CE 4.21.

Windows Mobile 5.0

Windows Mobile 5.0 (кодовое имя Magneto) — выпущена 9 мая 2005 года. Основана на Windows CE 5.0 и использует .NET Compact Framework 1.0 SP2.

Новые возможности

Новая версия Office — Office Mobile, в которую включен PowerPoint Mobile. Новая версия Office Mobile поддерживает оригинальные документы Microsoft Office и не требует их предварительной конвертации.

Windows Media Player 10 Mobile

Photo Caller ID

Улучшена поддержка Bluetooth

Поддержка клавиатуры QWERTY включена по умолчанию

ActiveSync 4.0

С AKU 3.2 поддержка .NET Compact Framework 2.0.

Интерфейс работы с GPS

Поддержка Persistent Storage («постоянного хранилища»), в котором хранятся все данные, копируясь по мере необходимости в ОЗУ. Обычно Persistent Storage реализовано в виде флэш-памяти.

Windows Mobile 5.0 была впервые представлена на конференции Mobile and Embedded Developers Conference 2005 в Лас-Вегасе 9-12 мая 2005 года.

Windows Mobile 6

Основана на Windows CE 5.2, кодовое название — Crossbow. Представлена на выставке 3GSM. Уже вышла, доступна в трёх редакциях:

Windows Mobile 6 Classic — для КПК

Windows Mobile 6 Professional — для коммуникаторов

Windows Mobile 6 Standard — для смартфонов

Новые возможности

Обновлённый внешний вид

Интеграция с Windows Live сервисами

Windows Mobile 6.1

(AKU — дополнительный пакет обновлений на версию WM 6)

Представлена 1 апреля 2008 года на выставке CTIA Wireless 2008.

Новые возможности:

- Карусельное меню Windows Mobile 6.1 (AKU)

Увеличена скорость работы интерфейса

SMS-сообщения отображаются в виде чата

В Internet Explorer добавлены функции «Масштаб» и «Обзор страницы»

Добавлен полноценный менеджер задач

Новый элемент меню «Managed Programs»

Улучшена работа по Wi-Fi и Bluetooth

Добавлена поддержка Bluetooth 2.1

Переработан алгоритм синхронизации с ПК

Добавлена поддержка MS Exchange Server 2007

Улучшен органайзер

Версия ядра не поменялась и осталась такой же — Windows CE 5.2

java midp2.1

Windows Mobile 6.5

Релиз состоялся 6 октября 2009 года.

Альтернативное название устройств под управлением Windows Mobile 6.5 — Windows Phone.

Выпуск этой версии неоднократно откладывался. Первым устройством должно было стать Sony Ericsson XPERIA X1.

Представляет собой усовершенствованную версию Windows Mobile 6.1 с рядом новых возможностей и обновлённым интерфейсом направленным на пользование устройством с помощью пальцев.

новый экран ожидания, с быстрым доступом к основным функциям устройства.

обновлён внешний вид меню.

новая версия мобильного браузера Internet Explorer Mobile 6.

Windows Mobile 6.5.3

Отличительными особенностями от Windows Mobile 6.5 являются:

Кнопка Пуск, увеличенная и смещённая в нижний бар;

Увеличенный нижний бар и уменьшенный верхний;

Новые виды меню стандартных программ.

Появился мультитач

Windows Phone 7

Windows Phone 7 (кодовое название «Photon») — операционная система Windows Mobile, разработанная Microsoft, основанная на Windows Embedded CE 7.0. Планируется к выпуску в 2010 году. Сначала она была назначена к выпуску в 2009 году, но через некоторые задержки Microsoft выпустила версию 6.5 в качестве временной версии между 6 и 7.

На Mobile World Congress в Барселоне, Microsoft рассказал подробности о Windows Phone 7 Series, которая оснащена интеграцией с Xbox Live и Zune. Интерфейс «Metro» полностью пересмотрен и визуально похож на интерфейс Zune HD. Microsoft заявили, что они предъявляют «жесткие, но справедливые» аппаратные требования к производителям, с Windows Phone 7 Series, для функциональности нужны три кнопки (кнопка «Назад», кнопка «Пуск» и кнопка Поиск) и FM-радио возможности. Microsoft переделала home screen, используются «плитки», которые прокручиваются по вертикали и могут быть настроены как быстрый запуск, ссылки на контакты или управления, содержатся виджеты. Windows Phone 7 Series будет иметь более дружественный пользовательский интерфейс с технологией Multi-Touch. Будет интегрированный стиль интерфейса Zune, Xbox Live и программы социальных сетей. Ранее сотрудники компании Microsoft заявляли, что ОС смартфонов, работающих под управлением Windows Mobile 6.x можно будет обновить до Windows Mobile 7. Но в действительности это оказалось не так, устройства, работающие под младшими версиями ОС, нельзя будет обновить до более новой версии ОС.

Демонстрационное видео работы WP7 доступно здесь. ОС запущена на прототипе смартфона от Samsung. Кодовое имя «Taylor» по всей видимости, переделанный i8910HD (ОС Symbian). Интерфейс двухмерный и абсолютно отличается от интерфейса WM6 - WM6.5, для переходов используются различные эффекты. Поддерживается мультисенсорность. Основная проблема WP7 - отсутствие многозадачности для пользовательских приложений при том что она поддерживается для базовых приложений. Стандарт разрешения дисплея для новой ОС 480x800. В ОС присутствует заявленная поддержка Zune и Xbox Live. Грэг Салливан (Greg Sullivan), старший менеджер по продукции Microsoft, назвал компании, которые первыми выпустят смартфоны для WP7: Dell, ASUS, LG, HTC и Samsung в конце года ближе к Рождеству. Microsoft обещает, что проблем с совместимостью приложений и сервисов не будет. Но это заявление не вызывает доверия так как WP7.x не совместима с WM6.x даже на уровне базового разрешения дисплея.

24. Операционная система Windows 7. Версии. Особенности. Перспективы.

Windows 7 — операционная система семейства Windows NT, следующая за Windows Vista. В линейке Windows NT система носит номер версии 6.1

Операционная система поступила в продажу 22 октября 2009 года,[2] меньше, чем через три года после выпуска предыдущей операционной системы, Windows Vista. Партнёрам и клиентам, обладающим лицензией Volume Licensing, доступ к RTM был предоставлен 24 июля 2009 года. В интернете оригинальные установочные образы финальной версии системы были доступны с 21 июля 2009 года.

Начальная Starter Обычно предустановлена на нетбуках

Домашняя базовая Home Basic

Домашняя расширенная Home Premium

Профессиональная Professional

Корпоративная Enterprise Для продажи частным лицам

МаксимальнаяUltimate

Начальная редакция (Windows 7 Starter) распространяется исключительно в версии OEM и не включает в себя предустановленной функциональной части для проигрывания H.264, AAC, MPEG-2. Домашняя базовая — предназначена исключительно для выпуска в развивающихся странах, в ней нет интерфейса Windows Aero с функциями Peek, Shake, и некоторых других функций. Также в ней есть те же ограничения на просмотр, что и в начальной редакции. В профессиональной, корпоративной и максимальной версиях существует поддержка XP Mode.

Кроме этого, согласно требованиям Еврокомиссии, для продаж в Европе планировалась специальная версия всех редакций с индексом «E» («Windows 7 E») — без предустановленного браузера Internet Explorer, однако она снята с продажи. Одной из причин снятия с продаж является недовольство партнёров и OEM-производителей отсутствием браузера в изначальной комплектации.[12] Всем, кто выполнил пред-заказ версии с индексом «E», было предложено заказать обновление или заказать полную версию с 1 сентября 2009 года.[13]

Тем не менее, в течение 5 лет планируется заменить встроенный Internet Explorer опциональным. В настоящее время данная схема согласована с Еврокомиссией.[14]

Все редакции, за исключением Начальной, существуют как в 32-битной, так и в 64-битной версиях.[15]

Максимальный объём оперативной памяти для 32-битных версий ограничен 4 Гб, Начальная редакция поддерживает до 2 Гб. Однако на практике пользователю часто доступен меньший объём, из-за аппаратных ограничений часто невозможно реализовать функцию режима PAE.

Поддержка более крупных объёмов памяти доступна только для 64-битных версий. Они поддерживают до 8 Гб («Домашняя базовая»), до 16 Гб («Домашняя расширенная»), все старшие версии могут адресовать до 192 Гб оперативной памяти.

Бесплатная 90-дневная версия Windows 7 Корпоративная доступна для ИТ-специалистов, желающих познакомиться с Windows 7 в рамках организации.[23]

25. ОС Ubuntu 7/8. Характеристики. Версии. Развитие (EdUbuntu, KUbuntu).

Ubuntu (рус. Убу́нту; зулу ubuntu — человечность[1]) — операционная система, использующая ядро Linux и основанная на Debian. Основным разработчиком и спонсором является компания Canonical. В настоящее время проект активно развивается и поддерживается свободным сообществом [2].

Ubuntu удерживает примерную долю мирового рынка в более, чем 20 миллионов пользователей,[3] что делает его самым популярным дистрибутивом Linux для десктопов[4]. Он является 4-ым в списке самых популярных ОС для веб-серверов[5] и его популярность быстро растёт.