Федеральное агентство связи

Ордена Трудового Красного Знамени федеральное государственное

бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра “Информатика”

Курсовая работа по дисциплине «Основы програмированния»

На тему: «Основы современных опирационных систем»

Выполнила:

Студентка 1 курса

Рагулина П.В.

Научный руководитель:

Доцент,

Кандидат педагогических наук,

Заместитель заведующего кафедрой, ТЫ У КОГО ЭТО В ГРУППЕ

СПИСАЛА??

ПРОСТО Я УЖЕ ЭТО ВИДЕЛ

ДАЛЬШЕ ТЫ НАЧИНАЕШЬ ИЗУЧАТЬ ГОСТ ПО НАСТОЯЩЕМУ И ЧИТАТЬ КОНСПЕКТ ЛЕКЦИЙ, НА КОТОРЫХ ДИКТОВАЛСЯ СООТВЕТСТВУЮЩИЙ МАТЕРИАЛ, ЧЕРТИТЬ РИСУНКИ И ТД, ЕСЛИ, КОНЕЧНО, ХОЧЕШЬ, ЧТОБЫ МЫ ПРОДВИНУЛИСЬ ДАЛЬШЕ, С ТОЧКИ ЗРЕНИЯ, ПРИЕМА КУРСОВОЙ, ПОТОМУ КАК ДАЛЕЕ СМОТРЕТЬ ПРИСЛАННОЕ НЕВОЗМОЖНО

Гуриков С.Р.

Москва 2020 г.

**Оглавление**

Глава 1.[Понятие операционной системы (ОС), цели ее работы. Классификация компьютерных систем. 7](#_Toc54892522)

* 1. [Классификация компьютерных систем 8](#_Toc54892523)
  2. [Особенности ОС для персональных компьютеров. 11](#_Toc54892524)
  3. [Распределенные компьютерные системы и особенности их ОС. 12](#_Toc54892525)
  4. [Обзор функций ОС: управление памятью, файлами, процессами, сетями, сервисы ОС, системные вызовы. 13](#_Toc54892526)
  5. [Управление файлами 14](#_Toc54892527)
  6. [Уровни абстракции ОС. Виртуальные машины. 16](#_Toc54892528)
  7. [Управление процессами. Планирование и диспетчеризация процессов. 21](#_Toc54892529)
  8. [Стратегии и критерии диспетчеризации процессов 24](#_Toc54892531)
  9. [Системы файлов 27](#_Toc54892532)

[Вывод 31](#_Toc54892533)

Техническое задание

1.1 Основание для разработки

Программный продукт разработан в соответствии с заданием, полученным от кафедры «Информатика» Московского технического университета связи и информатики, и утвержденное научным руководителем доцентом кафедры «Информатика» К.П.Н. Гуриковым С.Р. Дата утверждения: 2 октября 2020 года.

1.2 Назначение разработки.

Программный продукт предназначен для ознакомления с основными теоретическими положениями по теме курсовой работы «Основы современных операционных систем» и для проверки знаний пользователя по представленному материалу.

1.3 Требование к программному изделию

1.3.1 Требование к функциональным характеристикам

Разработанный программный продукт должен обеспечить выполнение следующих функций:

возможность ознакомления с теоретической частью по теме курсовой работы «Основы современных операционных систем»

* возможность вывода результатов исследования для пользователя.

1.3.2 Требование к надежности

Разработанное программное обеспечение должно иметь:

* устойчивую работу в соответствии с алгоритмом функционирования;
* выдавать сообщение об ошибках;
* поддерживать диалоговый режим в рамках предоставляемых пользователю решений;
* парольную защиту при запуске программы;
* защиту от несанкционированного копирования;

1.3.3 Требование к составу и параметрам технических средств

Минимальные и рекомендуемые системные требования для ПК:

* процессор с тактовой частотой не ниже 1,8 ГГц;
* объем оперативной памяти минимум 2 Гб;
* дискретный видеоадаптер;
* объем свободного дискового пространства;
* чтение компакт-диска.

1.3.4 Требование к информационной и программной совместимости

Программа должна легко устанавливаться, функционировать и корректно работать при наличии следующего программного обеспечения:

- Операционная система Windows 10;

- База данных Amazon RDS

1.3.5 Требование к транспортированию и хранению

Программа предоставляется на флеш-карте. Программная документация предоставляется в электронном и печатном виде.

* 1. Требование к программной документации

В ходе разработки программы должны быть подготовлены: текст и описание программы, методика испытаний, руководство пользователя.

* 1. Стадии и этапы разработки

Таблица 1- Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № этапа | Название этапа | Срок выполнения | Отчетность |
| 1 | Утверждение темы курсовой работы | 02.10.2020 | Выбрана тема |
| 2 | Написание Введения | 15.10.2020 | Определены актуальность, цели и задачи курсового проекта |
| 3 | Составление технического задания и написание теоретической части курсового проекта | 30.10.2020 | Подготовлены техническое задание и 1 глава |

**ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ**

**Актуальность темы.**  Операционная система является неотъемлемой частью ПК. Очевидно, что ознакомление с ПК необходимо начинать с ознакомления операционной системы, ведь без нее работа на ПК немыслима для большинства пользователей.

При включении компьютера операционная система загружается в память раньше остальных программ и затем служит платформой и средой для их работы. Без операционной системы невозможно представить работу с компьютером. Знание операционной системы необходимо для успешного пользования современными компьютерами. История развития информационных технологий характеризуется быстрым изменением концептуальных представлений, технических средств, методов и сфер их применения.

В современных реалиях весьма актуальным для большинства людей стало умение пользоваться промышленными информационными технологиями. Проникновение компьютеров во все сферы жизни общества убеждает в том, что культура общения с компьютером становится общей культурой человека. Также актуальность темы обусловлена потребностью улучшения операционных систем для повышения качества работы пользователя с ЭВМ, делая её, более простой, и освобождая его от обязанностей распределять ресурсы и управлять ими.

Как известно, процесс проникновения информационных технологий практически во все сферы человеческой деятельности продолжает развиваться и углубляться. Помимо уже привычных и широко распространенных персональных компьютеров становится все больше и встроенных средств вычислительной техники. С одной стороны, информационные технологии все усложняются. С другой стороны, упрощаются интерфейсы взаимодействия пользователей с компьютерами.

**Объектом исследования,** проведенного в рамках данной диссертации, является рассмотрение операционных систем.

**Предметом исследования,** является изучение основных особенностей каждой операционной системы.

**Цели работы и задачи исследования.** Цель курсовой работы – заключается в изучении наиболее распространенных операционных систем:

1. Изучить характеристику;
2. Рассмотреть основные понятия операционной системы и историю их создания;
3. Рассмотреть виды ОС, состав и основные функции;
4. Выявить особенности, достоинства и недостатки ОП;

**Методы исследования.** Для решения поставленных задач были использованы как экспериментальные, так и теоретические методы исследования. Теоретическую основу исследования составили труды в области современных операционных систем Сафонов, В.О. Рассмотрены методы управления оперативной и внешней памятью, виртуальная память, ее страничная и сегментная организация, методы управления процессами и потоками и их синхронизации, методы диспетчеризации процессов, методы распределения ресурсов и алгоритмы предотвращения и обнаружения тупиков, системы ввода-вывода, файловые системы, сети и сетевые протоколы, безопасность ОС и сетей.

**Глава 1. П****онятие операционной системы (ОС), цели ее работы. Классификация компьютерных систем.**

Операционная система( ОС, в англоязычном варианте-operating system ) – базовое системное , управляющее работой компьютера и являющееся посредником ( интерфейсом ) между аппаратурой( hardware), прикладным программным обеспечением (applicationsoftware) ипользователем компьютера ( user ). Фактически операционная система с точки зрения пользователя– это как бы продолжение аппаратуры, надстройка над ней, обеспечивающая более удобное, надежное и безопасное использование компьютеров и компьютерных сетей. Основные цели работы операционной системы следующие.

1. Обеспечение удобства, эффективности, надежности, безопасности выполнения пользовательских программ. Для пользователя самое главное – чтобы его программа работала, вела себя предсказуемо, выдавала необходимые ему правильные результаты, не давал асбоев, не подвергалась внешним атакам. Вычислительную среду длятакого выполнения программ и обеспечивает операционная система.
2. Обеспечение удобства, эффективности, надежности, безопасности использования компьютера. Операционная система обеспечивает максимальную полезность и эффективность использования компьютера и его ресурсов, обрабатывает прерывания, защищает компьютер от сбоев, отказов и хакерских атак. Эта деятельность ОС может быть не столь заметной для пользователя, но она осуществляется постоянно.
3. Обеспечение удобства, эффективности, надежности, безопасности использования сетевых, дисковых и других внешних устройств, подключенных к компьютеру. Особая функция операционной системы, без которой не возможно использовать компьютер, - это работа с внешними устройствами. Например, ОС обрабатывает любое обращение к жесткому диску, обеспечивая работу соответствующего драйвера (низкоуровневой программы для обмена информацией с диском) иконтроллера (специализированного процессора, выполняющего команды ввода-вывода с диском). Любая "флэшка", вставленная в USB-слот компьютера, распознается операционной системой, получает свое логическое имя (в системе Windows – в виде буквы, например, G) и становится частью файловой системы компьютера на все время, пока она не будет извлечена (демонтирована).
4. Подчеркнем особую важность среди функций современных ОС обеспечения безопасности, надежностии защиты данных. Следует учитывать, что компьютер и операционная система работают в сетевом окружении, в котором постоянно возможны и фактически происходят атаки хакеровиих программ, ставящие своей целью нарушение работы компьютера, "взлом" конфиденциальных данных пользователя, хранящихся на нем, похищение логинов, паролей, использование компьютера как "робота" для рассылки реклам или вирусов и др. В связи с этим в 2002 г. фирма Microsoft объявила инициативу по надежным и безопасным вычислениям (trustworthy computing initiative), целью которой является повышение надежности и безопасности всего программного обеспечения, прежде всего –операционных систем. В данном курсе мы будем подробно останавливаться на том, какие действия по обеспечению надежности, безопасности и защите данных предпринимают современные ОС.

**Классификация компьютерных систем**

Рассмотрим классификацию современных компьютерных систем, для которых разрабатываются и используются ОС –от **суперкомпьютеров** до **мобильных устройств**, -и суммируем требования к ОС для этих классов компьютеров.

**Суперкомпьютеры (super-computers)** –мощные многопроцессорные компьютеры, наиболее современные из которых имеют производительность до нескольких **petaflops** (1015 вещественных операций в секунду; аббревиатура **flops** расшифровываетсякак **floating point operations per second**). Суперкомпьютеры используются для вычислений, требующих больших вычислительных мощностей, сверхвысокой производительности и большого объема памяти. В реальной практике это прежде всего задачи моделирования –например, моделирования климата в регионе и прогнозирования на основе

построенной модели погоды в данном регионе на ближайшие дни. Особенностью суперкомпьютеров является их параллельная архитектура –как правило, все они являются многопроцессорными. Соответственно, ОС для суперкомпьютеров должны поддерживать распараллеливание решения задач и синхронизацию параллельных процессов, одновременно решающих подзадачи некоторой программы.

**Многоцелевые компьютеры, или компьютеры общего назначения (mainframes)** – традиционное историческое название для компьютеров, распространенных в 1950-х –1970-х гг.. Именно для mainframe-компьютеров создавались первые ОС. Типичные примеры таких компьютеров: IBM 360/370; из отечественных –М-220, БЭСМ-6. На таких компьютерах решались все необходимые задачи –от расчета зарплаты сотрудников в организации до расчета траекторий космических ракет. Подобный компьютер выглядел громоздко и мог занимать целый большой зал. Параметры ранних mainframe-компьютеров были весьма скромными: быстродействие -несколько тысяч операций в секунду, оперативная память –несколько тысяч ячеек (слов). Недостаточно удобным был пользовательский интерфейс (интерактивное взаимодействие с компьютерами было реализовано гораздопозже, в 1960-х гг.). Тем не менее, на таких компьютерах решались весьма серьезные задачи оборонного и космического назначения. С появлением персональных ипортативных компьютеров классические mainframe-компьютеры ушли в прошлое. Именно в операционных системах для mainframe-компьютеров были реализованы все основные методы и алгоритмы, которые впоследствии были использованы в ОС для персональных, карманных компьютеров и **мобильных устройств**.

**Кластеры компьютеров (computer clusters)** –группы компьютеров, физически расположенные рядом и соединенные друг с другом высокоскоростными шинами и линиями связи. Кластеры компьютеров используются для высокопроизводительных параллельных вычислений. Операционная система для кластеров должна, помимо общих возможностей, предоставлять средства для конфигурирования кластера, управления компьютерами (процессорами), входящими в него, распараллеливания решения задач между компьютерами кластера и мониторинга кластерной компьютерной системы. Примерами таких ОС являются ОС фирмы Microsoft –Windows 2003 for clusters; Windows 2008 High-Performance Computing (HPC).

**Настольные компьютеры (desktops)** –это наиболее распространенные в настоящее время компьютеры, которыми пользуются дома или на работе все люди. Такой компьютер размещается на рабочем столе и состоит из монитора, системного блока, клавиатуры и мыши. Все разнообразие современных операционных систем (Windows, Linux и др.) –к услугам пользователей настольных компьютеров. При необходимости на настольном компьютере можно установить две или более операционных системы, разделив его дисковую память на несколько разделов (partitions) и установив на каждый из них свою операционную систему, так что при включении компьютера пользователю предоставляется стартовое меню, из которого он выбирает нужную операционную систему для загрузки.

**Портативные компьютеры (laptops, notebooks** – –это миниатюрные компьютеры, по своим параметрам не уступающие настольным. На ноутбуках используются те же операционные системы, что и для настольных компьютеров (например, Windows или MacOS). Характерными чертами портативных компьютеров являются всевозможные встроенные порты и адаптеры для беспроводной связи: Wi-Fi ; Bluetooth. Внешние устройства подключаются к ноутбуку через порты USB. Другая характерная черта ноутбуков –это наличие кард-ридеров –портов для чтения всевозможных карт памяти, используемых в мобильных телефонах или цифровых фотокамерах; обеспечивается также интерфейс FireWire (официально –IEEE 1394) для подключения цифровой видеокамеры; таким образом, ноутбуки хорошо приспособлены для ввода, обработки и воспроизведения обработки мультимедийной информации. Один из критических параметров ноутбука –время работы его батарей без подзарядки;

**Карманные портативные компьютеры и органайзеры (КПК, handhelds, personal digital assistants –PDA)** –это устройство в виде миниатюрного компьютера, помещающегося на ладони или в кармане, но по своему быстродействию иногда не уступающего ноутбуку. Современные КПК имеют фактически те же порты и адаптеры, что и ноутбуки –Wi-Fi, Bluetooth, IrDA, USB. Операционные системы для КПК аналогичны ОС для ноутбуков, но все же учитывают более жесткие ограничения КПК по объему оперативной памяти. В настоящее время для КПК широко используется ОС Windows Mobile –аналог Windows для мобильных устройств. До недавнего времени была также широко распространена PalmOS для органайзеров типа PalmPilot фирмы 3COM. Разумеется, для КПК имеется аппаратура и программное обеспечение для подключения к ноутбуку или настольному компьютеру с целью синхронизации данных, что обеспечивает дополнительную надежность.

**Мобильные устройства (mobile intelligent devices –мобильные телефоны, коммуникаторы)** –это устройства, которыми каждый из нас пользуется постоянно для голосовой связи. С точки зрения автора, наиболее важные параметры мобильного устройства –это по-прежнему качество голосовой связи и время автономной работы батареи. Однако все большее значение приобретают встроенные в них цифровые фото-и видеокамеры. Операционные системы для мобильных устройств отличаются большей

компактностью. Эпоха доминирования на рынке мобильных телефонов операционных систем типа Symbian, видимо, заканчивается, и они уступают место более современным и обеспечивающим лучший пользовательский интерфейс ОС Google Android и Microsoft Windows Mobile. Для мобильных устройств, как и для КПК, весьма важная

характеристика ОС –это ее надежность, в частности, сохранность данных после переполнения памяти, возникающего, например, в результате приема большого числа SMS-сообщений, интенсивной фото-или видеосъемки. К сожалению, ОС Symbian, по опыту автора, ведет себя в этом отношении недостаточно надежно: при приеме "лишнего" SMS-сообщения, не помещающегося в память телефона, он блокируется с жалобным сообщением "Memory full", после чего приходится вручную отключать и включать телефон и таким образом перезагружать ОС. С этой точки зрения, ОС для мобильных устройств нуждаются в дальнейшем совершенствовании. Что касается прочего программного обеспечения для мобильных телефонов (игр, утилит, прикладных программ и др.), то при их разработке доминирует Java-технология.

**Носимые компьютеры (wearable computers)** –для повседневной жизни достаточно экзотические устройства, однако для специальных применений (например, встроенные в скафандр космонавта или в кардиостимулятор) они жизненно важны. Разумеется, их память и быстродействие значительно меньше, чем у настольных компьютеров, но критическим фактором является их сверхвысокая надежность, а для их операционных систем и прочего программного обеспечения –минимальное возможное время ответа (response time) –интервал, в течение которого система обрабатывает информацию от датчиков, от пользователя или из сети, превышение которого грозит катастрофическими последствиями. С этой точки зрения, ОС для носимых компьютеров можно отнести к системам реального времени.

**Распределенные системы (distributed systems)** –это системы, состоящие из нескольких компьютеров, объединенных в проводную или беспроводную сеть. Фактически, таковы ныне все компьютерные системы (вспомните девиз "Сеть –это компьютер"). Все операционные системы должны, таким образом поддерживать распределенный режим работы, средства сетевого взаимодействие, высокоскоростную надежную передачу информации через сеть.

**Системы реального времени (real-time systems)** –вычислительные системы, предназначенные для управления различными техническими, военными и другими объектами в режиме реального времени. Характеризуются основным требованием к аппаратуре и программному обеспечению, в том числе к операционной системе: недопустимость превышения времени ответа системы, т.е. ожидаемого времени выполнения типичной операции системы. Для ОС требования реального времени накладывают весьма жестких ограничения –например, в основном цикле работы системы недопустимы прерывания (так как они приводят к недопустимым временным затратам на их обработку).

**Особенности ОС для персональных компьютеров.**

Персональные компьютеры предназначены, как правило, для одного пользователя. Тем не менее, ОС для персональных компьютеров должна предусматривать режим мультипрограммирования (многозадачности), так как пользователям подчас удобнее выполнять несколько заданий параллельно –например, набирать некоторый текст в редакторе, принимать электронную почту и одновременно печатать на принтере какие-либо документы. Кроме того, при работе в локальной сети возможен удаленный вход на компьютер других пользователей. То есть, ОС для персональных компьютеров должна поддерживать также режим разделения времени. Персональные компьютеры имеют разнообразный набор устройств ввода-вывода, работу с которыми должна поддерживать операционная система с помощью драйверов –низкоуровневых системных программ для управления этими устройствами. Для пользователя удобнее всего, если все необходимые драйверы встроены в операционную систему. Однако ситуация осложняется тем что драйверы устройств разрабатывает обычно фирма-разработчик соответствующего устройства

-в англоязычной терминологии, Original Equipment Manufacturer (OEM), а не фирма-разработчик ОС. Поэтому при выпуске и установке на компьютер новой ОС могут возникнуть проблемы с драйверами –какое-либо устройство новая ОС "не понимает". На практике, должнопройти не менее двух-трех лет эксплуатации новой ОС, прежде чем для нее появятся драйверы для всех используемыхвнешних устройств, хотя в последнее время в этом отношении ситуация значительно улучшилась –новые ОС становятся все более "понятливыми" и имеют в своем составе огромные наборы драйверов.

**Распределенные компьютерные системы и особенности их ОС.**

В **распределенной системе (distributed system)** вычисления распределены между несколькими физическими процессорами (компьютерами),объединенными между собой в сеть.

**Слабо связанная система (loosely coupled system)**–распределенная компьютерная система, в которой каждый процессор имеет свою локальную память, а различные процессоры взаимодействуют между собой черезлинии связи –высокоскоростные шины, телефонные линии, беспроводную связь (Wi-Fi, EVDO, Wi-Max и др.).

**Преимущества распределенных систем:**

1.**Разделение (совместное использование) ресурсов**: в распределенной системе различные ресурсы могут храниться на разных компьютерах. Нет необходимости дублировать программы или данные, храня их копии на нескольких компьютерах.

2.**Совместная загрузка (load sharing)**: каждому компьютеру в распределенной системе может быть поручено определенное задание, которое он выполняет параллельно с выполнением другими компьютерами своих заданий.

3.**Надежность**: при отказе или сбое одного из компьютеров распределенной системы его задание может быть перераспределено другому компьютеру, чтобы сбой в минимальной степени повлиял или вовсе не повлиял на итоговый результат.

4**.Связь**: в распределенной системе все компьютеры связаны друг с другом, так что, например, при необходимости возможен удаленный вход с одного компьютера на другой с целью использования ресурсов более мощного компьютера. В распределенной системе компьютеры связаны в сетевую инфраструктуру, которая может быть:

1.локальнойсетью(local area network -LAN);

2.глобальной или региональной сетью (wide area network -WAN).

По своей организации распределенные системы могут быть клиент-серверными (client-server) или одноранговыми (peer-to-peer) системами. В клиент-серверной системе определенные компьютеры играют роль серверов, а остальные –роль клиентов, пользующихся их услугами.

**Обзор функций ОС: управление памятью, файлами, процессами, сетями, сервисы ОС, системные вызовы.**

Основную (оперативную) память компьютерной системы можно рассматривать как большой массив слов или байтов, каждый из которых имеет свой адрес. Память -это хранилище данных с быстрым доступом, совместно используемоепроцессором и устройствами ввода-вывода.Следует иметь в виду важную особенность основной памяти. В компьютерных архитектурах имеется два различных способа нумерации байтов в слове. По традиции будем представлять себе память как линейный массив, расположенный "слева направо", такой, что адреса слов, находящихся левее, меньше, чем адреса слов, находящихся правее. Каждое слово делится на байты, имеющие в слове свои номера –0, 1 и т.д. Если нумерация байтов в слове начинается слева, т.е. начиная со старших битов, то такую архитектуру принято называть big endian, если же справа, т.е. начиная с младших битов, то little endian. Например, при big endian –архитектуре 32-разрядного процессора байты двух соседних слов памяти нумеруются так: 0, 1, 2, 3, 0, 1, 2, 3. При little endian же архитектуре нумерация будет иной: 3, 2, 1, 0, 3, 2, 1, 0. При little endian –архитектуре приходится хранить и обрабатывать байтовые массивы и массивы слов отдельно, и нельзя изменять точку зрения на одну и ту же область памяти и рассматривать ее то как массив байтов, то как массив слов. При программировании на языках высокого уровня разработчику, как правило, не приходится учитывать это различие. Однако если при реализации распределения памяти требуется одну и ту же область памяти рассматривать то как массив слов, то как массив байтов, то для little endian –архитектур могут быть "сюрпризы", связанные с тем, что при записи в память как в массив слов байты как бы переставляются. Основная память –это неустойчивое (volatile) устройство памяти. Ее содержимое теряется при сбое системы или при выключении питания.

ОС отвечает за следующие действия, связанные с управлением памятью:

* **Отслеживание того, какие части памяти в данный момент используются и какими процессами.** Как правило, ОС организует для каждого процесса свою виртуальную память –расширение основной памяти путем хранения ее образа на диске и организации подкачки в основную память фрагментов (страниц или сегментов) виртуальной памяти процесса и ее откачки по мере необходимости.
* **Стратегия загрузки процессов в основную память, по мере ее освобождения.** При активизации процесса и его запуске или продолжении его выполнения процесс должен быть загружен в основную память, что и осуществляется операционной системой. При этом, возможно, какие-либо не активные в данный момент процессы приходится откачивать на диск.
* **Выделение и освобождение памяти по мере необходимости.** ОС обслуживаетзапросы вида "выделить область основной памяти длиной n байтов" и "освободить область памяти, начинающуюся с заданного адреса, длиной m байтов". Длина участков выделяемой и освобождаемой памяти может быть различной. ОС хранит списка занятой и свободной памяти. При интенсивном использовании памяти может возникнуть ее фрагментация –дробление на мелкие свободные части, вследствие того, что при запросах на выделение памяти длина найденного сегмента оказывается немного больше, чем требуется, и остаток сохраняется в списке свободной памяти как область небольшого размера. При исчерпании основной памяти ОС выполняет сборку мусора –поиск не используемых фрагментов, на которые потеряны ссылки, и уплотнение (компактировку) памяти –сдвиг всех используемых фрагментов по меньшим адресам, с корректировкой всех адресов.

**Управление файлами**

**Файл (file)** –совокупность логически взаимосвязанной информации, расположенная во внешней памяти. Как правило, файлы представляют программы (в видеисходного текста или в двоичной форме) или данные.

Другой термин, использованный для обозначения файлов фирмой IBM в ее операционной системе –IBM 360/370, -набор данных (data set).

ОС отвечает за следующие действия, связанные с управлением файлами.

**Создание и удаление файлов. Отображение файлов на внешнюю память.** ОС выделяет внешнюю память при создании нового файла. Файл в большинстве файловых систем состоит из заголовка и памяти. В заголовке хранятся атрибуты файла, например, его длина, тип, ссылка на элементы файла во внешней памяти. Память файла может быть организована по-разному –как список смежных областей (блоков или записей), одна смежная область, список индексных узлов, ссылающихся на блоки файла и т.д. Кроме создания и удаления файла, основные операции над ним –открытие и закрытие. Открытие файла –это считывание в основную память его заголовка и, возможно, одного или нескольких соседних блоков. Оно должно быть выполнено перед выполнениемопераций чтения из файла или записи в файл. Закрытие файла –это обратная операция: сброс всех копий блоков на внешнюю память и освобождение областей основной памяти, занятых открытым файлом. ОС закрывает файлы процесса при его завершении, если процесс несделал этого явно.

**Сброс**, или резервное копирование (backup) файлов на устойчивые носители (флэш-память, компакт-диск, ленточный стример и др.), с целью их последующего восстановления при сбое или приошибке пользователя. Значение резервного копирования для пользователей ОС трудно переоценить. Все наиболее важные документы, директории, файловые системы должны регулярно копироваться на внешнюю память (желательно делать не одну, а несколько копий на разные носители). Это должно стать непреложным правилом для каждого пользователя. Трудно даже вспомнить, сколько раз автору приходилось выслушивать сетования и жалобы студентов, аспирантов, сотрудников на то, что у них в самый ответственный момент "полетел винчестер", из-за чего они не могут показать свою программу или отчет. Рецепт очень простой: необходимо регулярно копировать важную информацию на устойчивые носители. Если Вы работаете в локальной сети фирмы, исследовательской лаборатории и т.д., то в ней должен быть системный администратор, который должен заботиться о регулярном резервном копировании всех важных файловых систем. Возможности ОС позволяют выполнять такое копирование автоматически, в определенное время, -например, ночью, когда в офисе никого нет, но компьютеры локальной сети работают.

**Управление вторичной памятью**

Поскольку размер основной памяти недостаточен для постоянного хранения всех программ и данных, в компьютернойсистеме должна быть предусмотрена вторичная (внешняя) память для откачки (back up, swapping) части содержимого основной памяти.

В большинстве компьютерныхсистем в качестве главной вторичной памяти для хранения программ и данных используются диски.

ОС отвечает за выполнение следующих действий, связанных с управлением дисками:

* **Управление свободной дисковой памятью;**
* **Выделение дисковой памяти;**
* **Диспетчеризация дисков (disk scheduling).**

При управлении вторичной памятью возникают проблемы, аналогичные проблемам распределения основной памяти. Всякая память, даже самая большая по объему, рано или поздно может исчерпаться, либо фрагментироваться на множество мелких областей свободной памяти.

**Управление сетевыми (распределенными) системами**. Как уже было сказано в более ранних лекциях, распределенная система –это совокупность процессоров, которые не используютобщую память или часы (такты процессора). Каждый процессор имеет собственную локальную память. Процессоры в такой системе соединены в сеть. Сетевое взаимодействие выполняется по определенному протоколу (интерфейсу, набору операций). Наиболее распространенный сетевой протокол –TCP/IP, основанный на IP-адресах машин (hosts)

**Система защиты (ptotection).** Термин защита (protection) используется для обозначения механизма управления доступом программ, процессов и пользователей к системным и пользовательским ресурсам.Механизм защиты в ОС должен обеспечивать следующие возможности: Различать авторизованный, или санкционированный (authorized) и несанкционированный (unauthorized) доступ. Под авторизацией понимается предоставление операционной системой пользователю или программе какого-либо определенного набора полномочий (permissions), например, возможности чтения или изменения файлов в файловой системе с общим доступом.

Описывать предназначенные для защиты элементы управления (конфигурации). Например, в UNIX используются специальные текстовые конфигурационные файлы для представления информации о файловых системах, к которым возможен сетевой доступ, с указанием списка машин (хостов), с которых возможен доступ, и набора действий, которые могут быть выполнены.

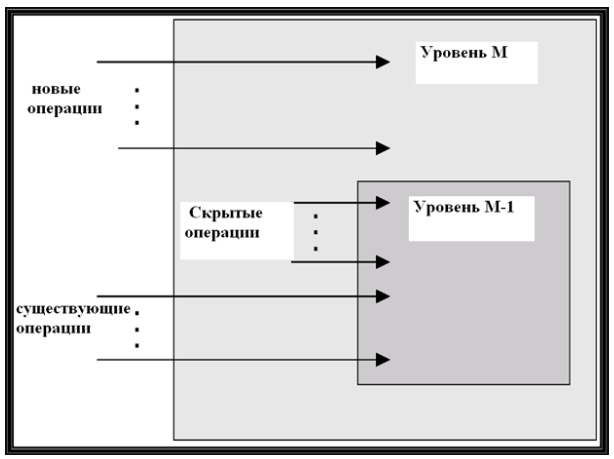
Обеспечивать средства выполнения необходимых для защиты действий (сигналы, исключения, блокировка и др.). Например, система защиты ОС должна фильтровать сетевые пакеты, получаемые извне локальной сети, выбирать и отсеивать "неблагонадежные" (получаемые с подозрительных IP-адресов), сообщать пользователю об обнаруженных и ликвидированных попытках сетевых атак с целью "взлома" Вашего компьютера (что и происходит на практике, например, при работе в Windows, когда Вы выходите с Интернет с Вашего компьютера). Если Вы нарушили

условия защиты (например, Ваша программа попыталась обратиться к файлу, работать с которым у Вас нет полномочий), ОС должна выдать понятное сообщение и прекратить работу программы. В современных системах это делается с помощью генерации исключений (exceptions), например, SecurityException.

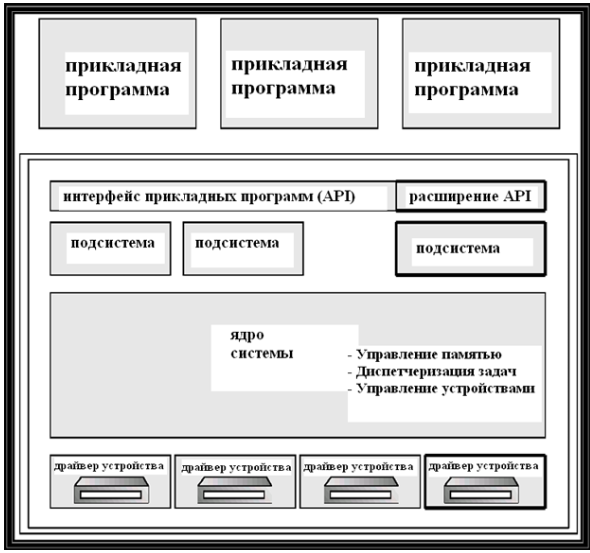
**Уровни абстракции ОС. Виртуальные машины.**

Проектирование и реализация операционных систем адекватно соответствуют

принципу уровней абстракции. На рис.1.1. изображены уровни абстракции при разработке ОС.Каждый новый уровень абстракции M определяет новые операции, в реализации которых используютсяоперации предшествующего уровня M-1. Кроме того, уровень M-1 может иметь скрытые операции, используемыетолько на этом уровне и недоступные другим уровням.

**Рис.1.1**.Уровни абстракции ОС.

На рис. 1.2. изображены уровни абстракции операционной системы OS/2 –второй ОС , 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.



**Рис. 1.2**.Уровни абстракции OS/2

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.

**Виртуальная машина Java (JVM)**

Подход, основанный на виртуализации, характерен не только для разработки операционных систем, но и для реализации современных платформ т языков

программирования. Причина в том, что реализаторы этих языков и платформ стремятся сделать их переносимыми с одной реальной аппаратнойплатформы на другую. Такой подход принят, как широко известно, при реализации Java, но авторы Java отнюдь не первыми предложили данную идею. Программы на Java компилируются в платформонезависимый байт-код (bytecode) –команды виртуальной Java-машины, построенные на основе постфиксной записи операндов. Байт-код исполняется виртуальной машиной Java (JVM).

JVM состоит из:

1. загрузчика классов (class loader), выполняющего загрузку классов в виртуальную машину во время выполнения программы; загрузчик классов может быть стандартным или может быть переопределен пользователем;
2. верификатора классов (class verifier), выполняющего при загрузке класса проверку корректности его байт-кода, контроль типов и другие необходимые проверки;
3. интерпретатора (runtime interpreter), выполняющего интерпретацию (эмуляцию) команд байт-кода –абстрактной машины Java;
4. Just-In-Time (JIT) –компилятора, выполняющего при первом вызове каждого метода его компиляцию в объектный код целевой платформы (native –код), что позволяет повысить суммарную производительность выполнения программ на Java.

Аналогичную архитектуру имеет виртуальная машина VES (Virtual Execution System) платформы Microsoft.NET, однако подход .NET более открытый –поддерживается многоязыковое программирование, и байт-код (в .NET называемый CIL –Common Intermediate Language) играет роль универсального промежуточного языка, в который компилируется исходный код на любом языке, например, на C# или Visual Basic. Подробное рассмотрение этих вопросов выходит за рамки данного курса. В качестве базового учебника по Java-технологии рекомендую книгу "Виртуальные файловые системы (VFS). Реализации файловых систем. Сетевая файловая система NFS".

Архитектура виртуальной машины Java изображена на рис. 1.3.



**Рис. 1.3**.Архитектура виртуальной машины Java (JVM).

**Реализация операционных систем**

Традиционно, начиная с 1950-х гг., ОС разрабатывались на низкоуровневых языках –ассемблерах. Система UNIX была первой ОС, разработанной почти полностью на языке высокого уровня –Си. Впоследствии язык Си стал одним из наиболее популярных языков системного программирования. Благодаря новаторскому подходу авторов UNIX, сложилась тенденция разработки ОС на языках высокого уровня.

Преимущества использования языков высокого уровня очевидны: код на языке высокого уровня

1. может быть разработан быстрее
2. более компактен
3. легче для понимания и отладки.

Кроме того, операционная система гораздо легче переносима на другие аппаратные платформы, если она разработана на языке высокого уровня.Гипотетический недостаток у языков высокого уровня, по сравнению с ассемблерами, только один –возможная неэффективность реализации высокоуровневых конструкций языка, по сравнению с "ручным" программированием на ассемблере. Однако данная проблема может быть решена путем разработки эффективного оптимизирующего компилятора с языка высокого уровня и не должна препятствовать его использованию.Близкими автору примерами разработки ОС на языках высокого уровня, помимо UNIX, является разработка на Си ОС Solaris фирмой Sun, а также разработка ОС "Эльбрус" на языке высокого уровня Эль-76 "Системы ввода-вывода ". Поставка ОС "Эльбрус" вместе с ее исходными кодами была весьма полезна для программистов и стимулировала целый ряд работ отраслевых организаций по разработке ОС реального времени на базе ОС "Эльбрус".

**Генерация операционной системы**

ОС проектируются с целью использования на любом компьютереиз некоторого класса. В поставку ОС входит очень большой набор типовых модулей для различных классов компьютеров(например, настольных и портативных). Поэтому для каждого компьютера система должна быть сконфигурирована при ее инсталляции –установке ОС на конкретный компьютер.Программа генерации ОС получает информацию о специфической конфигурации компьютерной системы. После генерации и инсталляции ОС система готова к работе.Следующий этап -загрузка (booting) –запуск компьютера посредством загрузки ядра ОС. При запуске компьютера первой запускается программа раскрутки (bootstrap program) –код, хранящийся в постоянной памяти (ROM) базовой системы ввода-вывода (BIOS) компьютера. Код программы раскрутки находит ядро ОС, загружает его в память и запускает

## **Управление процессами. Планирование и диспетчеризация процессов.**

**Понятие процесса**

**Процесс (process)** это программа пользователя при ее выполнении. При своей работе операционная системы исполняет множество классов программ: пакетные задания; пользовательские программы в режиме разделения времени; системные программы и процессы. Имеется несколько схожих терминов, характеризующих пользовательские программы: процесс (process), задание (job), задача (task)Однако не будем здесь преувеличивать различие между ними: для лучшего понимания специфики процессов и управления ими в ОС, мы можем считать приведенные термины синонимами, как и считается ио многих учебниках по ОС.

Важная особенность процесса: это единица вычислений, которая должна выполняться последовательно, т.е. каждый процесс имеет свой последовательный поток управления (control flow) –последовательность выполняемых процессом команд. Многие большие задачи успешно решаются путем параллельного выполнения процессов, но об этом речь пойдет немного позже.

Процесс при его создании и управлении им операционной системой включает следующую основную информацию:

* Счетчик команд (program counter -PC) –адрес текущей выполняемой команды процесса; обычно хранится в специальном системном регистре аппаратуры;
* Стек (stack) –резидентная область основной памяти, выделяемая операционной системой при создании процесса, в которой хранятся локальные данные процедур процесса, их параметры (аргументы) и связующая информация между ними, необходимая для организации вычислений. При запуске очередной процедуры в стеке отводится запись активации (activation record), называемая также стековым фреймом (stack frame) и областью локальных данных (local data area) для хранения локальных данных текущего поколения (запуска) процедуры. По окончании ее выполнения запись активации удаляется из стека;

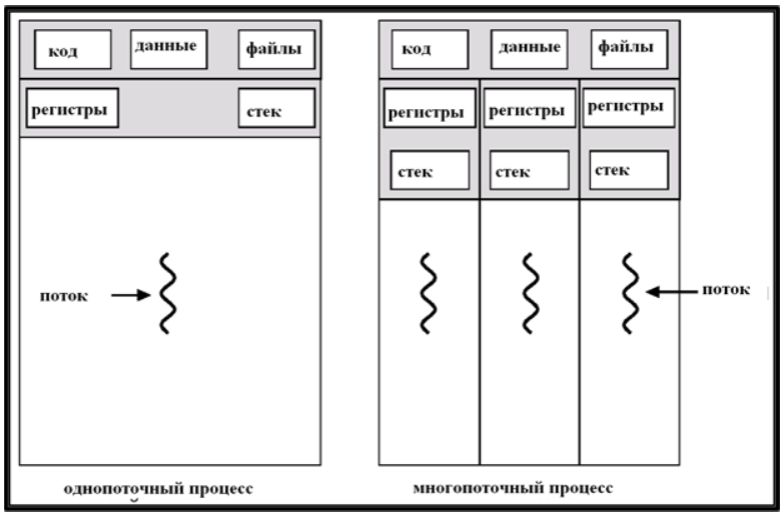
Секция данных (data section) –статическая (постоянно выделенная, неизменного размера) область основной памяти, выделяемая операционной системой процессу, в которой хранятся его глобальные переменные, массивы, структуры, объекты.

Исполняемый код (команды) процесса первоначально хранится во вторичной памяти (на диске) и загружается в основную память полностью или частично при обращении к нему.

## **Потоки (threads) и многопоточное выполнение программ (multi-threading)**

**Однопоточные и многопоточные процессы**

К сожалению, до сих пор мышление многих программистов при разработке программ остается чисто последовательным. Не учитываются широкие возможности параллелизма, в частности, многопоточности. Последовательный (однопоточный) процесс –это процесс, который имеет только один поток управления (control flow), характеризующийся изменением его счетчика команд. Поток (thread) –это запускаемый из некоторого процесса особого рода параллельный процесс, выполняемый в том же адресном пространстве, что и процесс-родитель. Схема организации однопоточного и многопоточного процессов изображена на рис. 2.1



**Рис. 2.1.**Однопоточный и многопоточный процессы.

Как видно из схемы, однопоточный процесс использует, как обычно, код, данные в основной памяти и файлы, с которыми он работает. Процесс также использует определенные значения регистров и стек, на котором исполняются его процедуры. Многопоточный процесс организован несколько сложнее. Он имеет несколько параллельных потоков, для каждого из которых ОС создает свой стек и хранит свои собственные значения регистров. Потоки работают в общей основной памяти и используютто же адресное пространство, чтои процесс-родитель, а также разделяют код процесса и файлы.

Многопоточность имеет большие преимущества:

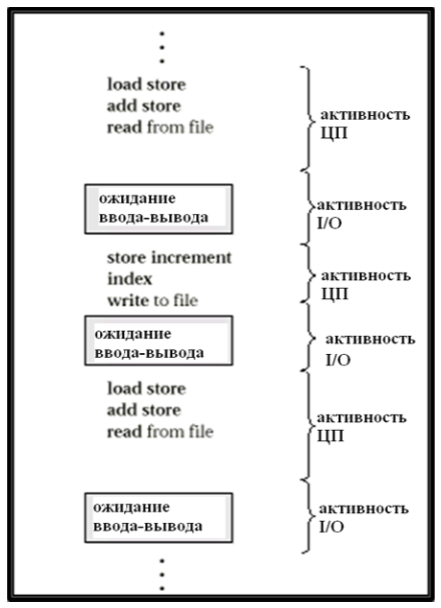
* **Увеличение скорости** (по сравнению с использованиемобычных процессов). Многопоточность основана на использовании облегченных процессов (lightweight processes), работающих в общем пространстве виртуальной памяти. Благодаря многопоточности, не возникает больше неэффективных ситуаций, типичных для классической системе UNIX, в которой каждая команда shell (даже команда вывода содержимого текущей директории ls исполнялась как отдельный процесс, причем в своем собственном адресном пространстве. В противоположность облегченным процессам, обычные процессы (имеющие собственное адресное пространство) часть называют тяжеловесными (heavyweight).
* **Использование общих ресурсов**. Потоки одного процесса используют общую память и файлы.
* **Экономия**. Благодаря многопоточности, достигается значительная экономия памяти, по причинам, объясненным выше. Также достигается и экономия времени, так как переключение контекста на облегченный процесс, для которого требуется только сменить стек и восстановить значения регистров, значительно быстрее, чем на обычный процесс.

**Использование мультипроцессорных архитектур.** Это особенно важно в настоящее время, в период широкого использования многоядерных гибридных и многопроцессорных систем. Именно многопоточность программ, основанная на многоядерности процессора, дает возможность, наконец, почувствовать реальные преимущества параллельного выполнения.

## **Стратегии и критерии диспетчеризации процессов**

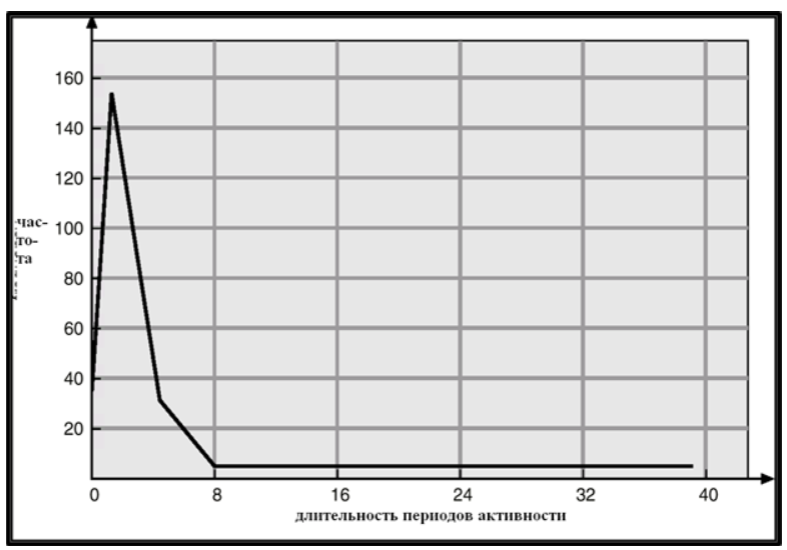
**Основные понятия диспетчеризации процессов**

Диспетчеризация процессора –распределение его времени между процессами в системе. Цель диспетчеризации –максимальная загрузка процессора, достигаемая с помощьюмультипрограммирования. Исполнение любого процесса можно рассматривать как цикл CPU / I-O –чередование периодов использованияпроцессора и ожидания ввода-вывода.Распределение периодов активности процессора (bursts) и ввода-вывода изображено на рис. 3.1



**Рис. 3.1**.Последовательность активных фаз процессора и фаз ввода-вывода.

На рис. 3.2 изображена примерная гистограмма периодов активности процессора, основанная на анализе реального поведения процессов в операционных системах.



**Рис. 3.2**. Гистограмма периодов активности процессора.

Из схемы видно, что чем короче период активности, тем выше частота таких периодов, и наоборот, т.е. частота периодов активности обратно пропорциональна их длительности.

**Планировщик процессора**

**Планировщик** –компонента ОС, которая выбирает один из нескольких процессов, загруженных в память и готовых к выполнению, и выделяет процессор для одного из них.

Решения по диспетчеризации могут быть приняты в случаях, если процесс:

1. Переключается из состояния выполнения в состояние ожидания.
2. Переключается из состояния выполнения в состояние готовности к выполнению.
3. Переключается из состояния ожидания в состояние готовности.
4. Завершается.

Диспетчеризация типов 1 и 4 обозначается термином **диспетчеризация без прерывания процесса (non-preemptive).**

Диспетчеризация типов 2 и 3 обозначается термином **диспетчеризация с прерыванием процесса (preemptive).**

**Собственно диспетчер процессора**

Диспетчер процессора –компонента ОС, предоставляющая процессор тому процессу, который был выбран планировщиком. Диспетчер выполняет последовательность действий:

* Переключает контекст
* Переключает процессор в пользовательский режим
* Выполняет переход по соответствующему адресу в пользовательскую программу для ее рестарта.

**Скрытая активность (латентность) диспетчера (dispatch latency)** –время, требуемое для диспетчера, чтобы остановить один процесс и стартовать другой. Разумеется, система должна стремиться минимизировать это время, однако набор критериев диспетчеризации более сложен.

**Управление памятью**

**Основные положения размещения процессов в памяти**

Любая программа, введенная в систему, должна быть размещена в памяти и оформлена в виде процесса для ее выполнения. Каждая программа при вводе в систему помещается во входную очередь –совокупность процессов на диске, ожидающих размещения в памяти для выполнения своих программ. До своего выполнения пользовательские программы проходят в системе несколько стадий.

**Связывание программ и данных с адресами в памяти**

Перед загрузкой данных или кода в память они должны быть в какой-либо момент связаны с определенными адресами в памяти. Связывание может выполняться на разных этапах:

* **Связывание во время компиляции (compile-time).** Если адрес в памяти априорно известен, компилятором может быть сгенерирован код с абсолютными адресами. При любом изменении размещения программы в памяти должна быть выполнена перекомпиляция. Данный подход более характерен для ранних компьютерных систем с небольшим объемом памяти, либо для обработки ивыполнения системных модулей –частей ядра ОС, для которых характерно использование резидентных абсолютных адресов. Для пользовательских программ такой подход неудобен, так как не обеспечивает достаточной гибкости, в частности, возможностибез изменений перезагрузить код в другую область памяти.
* **Связывание во время загрузки (load-time).** Загрузка программы в память –стадия ее обработки системой, предшествующая выполнению программы. Чтобы начальный адрес области памяти, куда загружается программа, можно было менять, и это не привело бы к необходимости изменения кода программы, применяется следующий метод. Генерируется перемещаемый код (relocatable code) –код, в котором адресация происходит относительно значения регистра перемещения (relocation register), и адрес в памяти равен суммезначения регистра перемещения и адреса, вычисляемого в команде. Таким образом, при необходимости загрузки кода на другое место в памяти требуется изменить только значение регистра перемещения. Подобный подход широко используетсядля программ, написанных на традиционных языках программирования.
* **Связывание во время исполнения (runtime), или динамическое (позднее) связывание.** Используется, если процесс во время выполнения может быть перемещен из одного сегмента памяти в другой. Для реализации связывания во время исполнения требуется аппаратная поддержка отображения адресов –например, регистры базы и границы. В большинстве систем для пользовательских программ используется, главным образом, именно связывание во время исполнения.

## **Системы файлов**

Файл –совокупность логически объединенных данных во внешней памяти. Управление файлами –одна из важных задач ОС, так как в виде файлов в системе хранится практически любая информация –программы и данные. В лекции рассмотрены следующиевопросы:

**Понятие файла**

Файл (file) –это смежная область логического адресного пространства. Как правило, файлы хранятся во внешней памяти.

Немного о терминологии. Слово файл уже несколько десятков лет используетсякак русское –один из многочисленных примеров программистских неологизмов. Первоначально, когда около 50 лет назад появился данный английский термин, в русскоязычной литературе специалисты пыталисьввести другую терминологию –слово file переводили как фонди даже тека (в смысле хранилище). Однако исторически сложилось иное решение–слово файл стало русским. В английском языке слово file имеет много других значений: например, подшитый в папку бумажный документ и даже стадо (например, слонов) –в последнем случае, как можно предположить, размер "файла" может быть очень велик. У всех в памяти название легендарного сериала "X files" (в вольном русском переводе –"Секретные материалы").

Фирма IBM в документации по своей системе IBM 360 в 1960-х гг. использовалаиной термин –набор данных (data set) –для обозначения этого же понятия, однако он не пережил операционную систему,в которой использовался.

Каждый файл имеет свой тип, определяющий, какая информация хранится в файле. Основные типы файлов –программа (код) или данные. Данные подразделяются на числовые, символьные (текстовые) и двоичные (произвольная информация).

**Системы ввода-вывода**

**Аппаратура ввода-вывода**

В настоящее время наблюдается все более и более активное развитие устройств ввода-вывода в компьютерныхсистемах. В значительной степени это объясняется, во-первых, необходимостью ввода, обработки и вывода мультимедийной информации (аудио, видео, цифровых фотографий, отсканированных образов и других изображений), во-вторых, постоянной потребностью в увеличении скорости и емкости устройств вследствие гигантского роста размеров обрабатываемой информации. Еще в 1980-х гг., например, нормой считалось использование гибких дискет (FDD) емкостью 1.44 мегабайта для резервного копирования. Сейчас устройствами FDD настольные и портативные компьютеры вообще не комплектуются, а, что касается резервного копирования, то и устройств емкостью 128 гигабайт может оказаться недостаточно для этой цели.

**Набор устройств включает, в частности:**

* клавиатуру и мышь;
* жесткие диски (HDD), включая внутренние и внешние (ZIV drives);
* flash-память;
* ленточные стримеры;
* компакт-диски BluRay, DVD, CD;
* твердотельные накопители на магнитных дисках (solid state disks –SSD);ZIP drives, JAZ drives –уже устаревающие, но еще используемые устройствадля резервного копирования со съемными носителями (их постепенно вытесняет флэш-память, физические размеры модулей которой гораздо меньше, а емкость –больше);
* магнито-оптические диски –ныне уже устаревшие устройства для резервного копирования, но долгое время использовавшиеся;
* устройства для мультимедийного ввода-вывода: порты и адаптеры IEEE 1394 (Fire-Wire) для подключения цифровых видеоустройств;
* порты и адаптеры High Definition Multimedia Interface (HDMI) для подключения видеоаппаратуры стандарта High Definition (HD);
* кард-ридеры для нескольких форматов (SmartMedia и др.) носителей, используемых в цифровых фотоаппаратах;
* мониторы, видеокарты (видеоадаптеры) и графические процессоры, в том числе –многоядерные;
* принтеры, сканеры.

**Системные механизмы Windows**

Базовая система файлов ОС Windows 2000, как и ОС Windows NT, -NTFS (NT File System).

Фундаментальная структура системы файлов Windows 2000 (NTFS) –**том (volume)**. Том создается утилитой администрирования диска. Структура тома основана на **логическом диске (partition).** Том может занимать часть диска, целый диск илираспределяться по нескольким дискам (рис. 28.1). Все метаданные, такие как информация о томе, хранятся в обычном файле.

NTFS используеткластеры как базовую единицу выделения дисковой памяти. Кластер –число секторов диска, размер которого –степень двойки. Поскольку размер кластера меньше, чем в FAT16, внутренняя фрагментация уменьшается.

NTFS использует **логические номера кластеров -logical cluster numbers** (LCN) в качестве дисковых адресов. Файл в NTFS –не просто байтовый поток, как в MS-DOS или в UNIX,но структурированный объект, состоящий из атрибутов. Каждый файл в NTFS описывается одной или несколькими записями в массиве, хранящемся в специальном файле, называемом **Master File Table (MFT).** Каждый файл в томе NTFS имеет уникальный идентификатор (ID), называемый ссылкой на файл -file reference. Это 64-битовое число, состоящее из 48-битового номера файла и 16-битового номера последовательности. Ссылка на файл может использоваться для выполнения внутренних проверок целостности.

Пространство имен NTFS организовано в иерархию директорий. Индексный корень (index root) содержит верхний уровень B+ -дерева.

Все изменения структур данных в файловой системе NTFS выполняются как **транзакции,** для которых используется журнал. Перед тем, как структура данных изменяется, транзакция заносит в журнал специальную запись, которая содержит информацию для повторного выполнения (redo) и отмены (undo) данного изменения. После изменения структуры данных в журнал заносится информация об успешном выполнении операции. В случае порчи информации файловая система может быть восстановлена до целостного состояния с использованием журнальных записей. Эта схема не гарантирует, что все данные пользовательского файла могут быть восстановлены в случае порчи информации, а гарантирует лишь, что все структуры данных о файлах в системе (метаданные) не повреждены и отражают какое-либо целостное состояние данных до порчи информации. Журнал хранится в третьем файле метаданных каждого тома.

**Безопасность** тома NTFS реализована на основе объектной модели Windows 2000. Каждый файловый объект имеет дескриптор безопасности, хранящийся в записи MFT. Данный атрибут содержит маркер доступа владельца файла, а также список управления доступом, устанавливающий права каждого пользователя для доступа к данному файлу.

FtDisk, дисковый драйвер Windows 2000, устойчивый к сбоям, обеспечивает несколько способов объединения нескольких SCSI-дисков в один логический том. Он логически конкатенирует диски, образуя один логический том (набор дисков тома –volume set). Драйвер выполняет обработку нескольких частей тома по принципу round-robin для формирования **"полосатого множества" (stripe set), или disk striping**) –рис. 28.2. Stripe set –набор от 2 до 32 дисков, логически объединенных в единый том. При записи данных в stripe set данные записываются порциями по 64 Кбайта (которые могут распределяться произвольным образом между дисками stripe set). Это позволяет сэкономить время в случае, если работа с дисками stripe set может выполняться параллельно. Такая схема не гарантирует сохранности данных.

Как вариант используется схема **stripe set with parity**, которая позволяет восстановить данные в случае сбоя.

Зеркальное отображение дисков (Disk mirroring) -это надежная схема, использующая **множество "зеркал" (mirror set)** две секции одного размера на разных частях диска с идентичным содержимым.

Для обработки запорченных дисковых секторов FtDisk использует аппаратный метод, называемый предохранением секторов (**sector sparing**), а NTFS использует программный метод, называемый повторным отображением кластеров (**cluster remapping**).

## **Вывод**

В данном курсе подробно рассмотрены как теоретические, так и практические основы современных операционных систем –от первых ОС до новейших ОС для облачных вычислений и мобильных устройств.

**Список используемых источников**

1. ГОСТ 19.201-78 ЕСПД Отчет о научно-исследовательской работе. Структура и правила оформления (с Поправками) URL: (http://docs.cntd.ru/document/gost-19-201-78)(дата обращения 29.10.2020)
2. Сафонов В.О. Основы современных операционных систем [Электронный ресурс]: учебное пособие/ Сафонов В.О.— Электрон. текстовые данные.— Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.— 826 c.— ЭБС «IPRbooks»URL: http://www.iprbookshop.ru/94855.html (дата обращения: 29.10.2020).