

1) Классификация ОС и особенности их функционирования.

Операционная система — программная среда, обеспечивающая поддержку работы всех программ, их взаимодействие с аппаратными средствами вычислительной системы, а также предоставляющая пользователю возможности общего управления вычислительной системой и использования ресурсов ВС.

По функциям:

- Однопрограммные ОС

Система работает в *однопрограммном* режиме если в системе находится одна или несколько задач, но все ресурсы системы отданы одной задаче, и она не может быть прервана другой. Задача возвращает ресурсы только при нормальном или аварийном завершении.

- Многопрограммные ОС

Система работает в *многопрограммном* режиме, если в системе находится несколько задач на разной стадии выполнения, и каждая из них может быть прервана другой с последующим восстановлением.

Основной **проблемой** многопрограммной работы является **организация защиты программ от взаимного влияния** как на уровне оперативной памяти, так и на разных уровнях внешней памяти.

Многопрограммный режим в однопроцессорной системе — это организация вычислительного процесса с планированием во времени. При этом операционная система обеспечивает выполнение активизированных процессов на одном и том же оборудовании, разделяя (синхронизируя) их работу во времени. В функции такой операционной системы также входит защита влияния одного процесса на другой. Основные заботы о синхронизации взаимодействующих процессов возложены на программиста.

Многопрограммный режим в многопроцессорной системе — это планирование во времени и в пространстве. В этом режиме операционная система, выполняя одну из своих основных функций (обеспечение эффективности работы ресурсов), должна распределять активизированные процессы по процессорам (в пространстве) и синхронизировать их работу во времени. В том случае, если количество задач больше количества процессоров, задача планирования, диспетчеризации усложняется. В связи со сложностью решения задач распределения процессов по процессорам, их решение выполняется или до активизации процессов в многопроцессорной системе (статическое планирование), или решаются простыми способами, не дающими оптимального решения.

- Однопроцессорные ОС
- Многопроцессорные ОС
- Распределенные ОС
- Кластерные
- Офисные

Распределенные операционные системы представляют собой совокупность вычислительных средств и периферийных устройств, объединенных между собой с обеспечением возможности взаимного обмена информацией.

- Виртуальные ОС

Виртуальная операционная система характеризуется обеспечением многопрограммного режима работы, когда каждый пользователь может работать в собственной операционной среде.

По структуре:

1. **Монолитные.** При увеличении любой функции необходимо переписывание всей системы.

Таким образом, **монолитное ядро** — это такая схема *операционной системы*, при которой все ее компоненты являются составными частями одной программы, используют общие структуры данных и взаимодействуют друг с другом путем непосредственного вызова процедур. Для монолитной *операционной системы* ядро совпадает со всей системой.

Во многих *операционных системах* с **монолитным ядром** сборка ядра, то есть его компиляция, осуществляется отдельно для каждого компьютера, на который устанавливается *операционная система*. При этом можно выбрать список оборудования и программных протоколов, поддержка которых будет включена в ядро. Так как ядро является единой программой, перекомпиляция — это единственный способ добавить в него новые компоненты или исключить неиспользуемые. Следует отметить, что присутствие в ядре лишних компонентов крайне нежелательно, так как ядро всегда полностью располагается в оперативной памяти. Кроме того, исключение ненужных компонентов повышает надежность *операционной системы* в целом.

2. **Многоуровневые.** ОС с кольцевой структурой. Иерархические. Недостаток в том, что такое ядро тяжеловесное.
3. **Микроядерная архитектура.** Система функционально независима и универсальна. Имеется ядро и функциональные блоки. Ядро здесь компактное и только передает блокам информацию по синхронизации и отслеживает пересылки между процессами, дальше блоки работают самостоятельно.
4. **Система с экзоядром**
 Экзоядро — ядро операционной системы компьютеров, предоставляющее лишь функции для взаимодействия между процессами и безопасного выделения и освобождения ресурсов.

На нижнем уровне ядра работает программа, которая называется экзоядро. В ее задачу входит распределение ресурсов для виртуальных машин, а после этого — проверка их использования (то есть отслеживание попыток машин использовать чужой ресурс). Каждая виртуальная машина на уровне пользователя может работать с собственной операционной системой, с той разницей, что каждая машина ограничена набором ресурсов, которые она запросила и которые ей были предоставлены. **Преимущества схемы экзоядра:** позволяет обойтись без уровня отображения. При других методах работы каждая виртуальная машина считает, что она использует свой собственный диск с нумерацией блоков от 0 до максимума. Поэтому монитор виртуальной машины должен поддерживать таблицы преобразования адресов на диске (и всех других ресурсов). Необходимость преобразования отпадает при наличии экзоядра, которому нужно только хранить запись о том, какой виртуальной машине выделен данный ресурс. Такой подход имеет еще одно преимущество: он отделяет многозадачность (в экзоядре) от операционной системы пользователя (в пространстве пользователя) с меньшими затратами, так как для этого ему необходимо всего лишь не допускать вмешательства одной виртуальной машины в работу другой.

Особенности распределенных ОС(?) Распределенная sys – совокупность выч. узлов, связанных между собой каналами связи, с точки зрения пользователя представляют собой единое целое. Отсутствие общей памяти приводит к невозможности определения общего состояния с помощью множества совместных переменных, а невозможность совместного обращения к памяти и различие в задержках передач сообщений приводит к тому что при определении состояния какого либо элемента системы из двух различных точек можно получить разные результаты. Выполнение работы распределяется в узлах исходя из соображения пропускной способности всей системы. Распределенные системы имеют высокий уровень организации параллельных вычислений.

Основное отличие ОС UNIX (UNIX – Rulezzz!!! 🐜) ОС взаимодействует с аппаратурой непосредственно (программы-косвенно), обеспечивая обслуживание программ и их независимость от деталей аппаратной конфигурации.

Назначения частей ОС ориентированной на пользователя и на 'hardware'. Часть системы, ориентированной на «hardware» скрывает от пользователя те физические особенности системы, которые ему не нужны. (обеспечение взаимодействия устройств, процессов и т.д.). Часть ОС, ориентированной на пользователя, обеспечивает интерфейс с пользователем.

Почему используются многоуровневые системы программирования Для облегчения разработки компилятора (ставится несколько уровней трансляторов). Декомпозиция – разбиение сложной задачи на более мелкие. (Каждый класс задач на своем уровне связь вниз – выполнение ф-ции, вверх возврат результата) Стандартизация интерфейса – удобство собирать по кирпичикам

Что такое виртуальная ОС Так называемая ОС, которая позволяет многим пользователям работающим на одной и той же технической базе(одно ус-во) одновременно работать в различных операционных средах. Виртуальные ОС – разрешает нескольким useram работать на одном и том же dev на разных ОС.

Какая часть ОС обрабатывает сигналы прерываний Ядро.

Мультипроцессорные системы – совокупность процессоров в одном конструктиве, имеющих сильную связь по передаче данных (Общая память, единая ОС)

ДИСТРИБУТИВ (исходный вариант) ОС

В чем сложность для ОС организации многопрограммного режима работы

Организация защиты от взаимного влияния друг на друга на уровне оперативной и на уровне внешней памяти; разделение аппаратных и программных ресурсов; планирование (во времени, а в случае ПВС и в пространстве).

Дать определение – обрабатывающие программы ОС

Программы выполнения стандартных (в рамках ОС) функций, обработки исключительных ситуаций. Обработка – изменение ин-фы с которой работает прога в составе ОС (дрова, загрузчик)

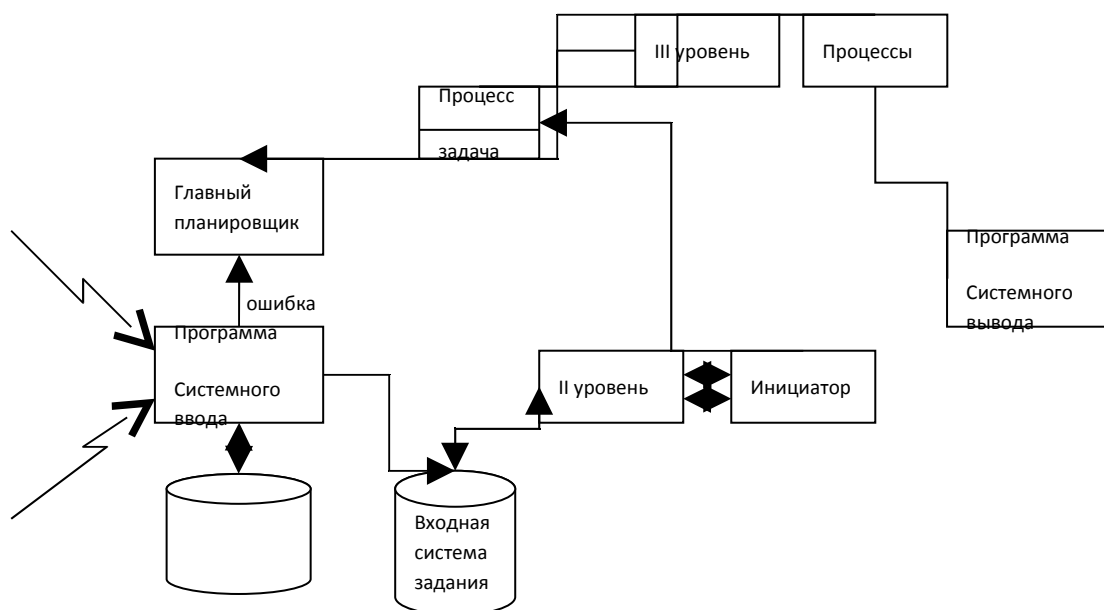
Дать определение – управляющие программы ОС

Программы постоянно находящиеся в памяти (резидентные) организующие корректное выполнение процессов и функционирование всех устройств системы при решении задач. Составляют ядро ОС. Управление : Заданиями – слежение за прохождением заданий от входа до выхода на всех этапах его выполнения. Задачами – (Процессами) – слежение за всеми задачами активизированными в системе и процессами их выполнения на ресурсах. Памятью – решение задач эффективного и/ mem (internal) в соответствии с ее организацией.(Защита – согласование ин-фы в кешах) Данными – эффективное размещение и и/ данных на внешних носителях (проблема эффективности и/ процессора) Внешними ус-вами ...

Что такое свопинг - способ реализации многопрограммного режима работы на однопроцессорной машине. Проблема настройки связана с перемещением программ в ОП, ЕСЛИ ОС работает со свопингом – нужна перенастройка адресных констант. (глобальная – перенастройка всех адресных констант, локальная – вычисление адреса той переменной, которая находится реально в ОП)

Что такое спулинг Режим буферизации для выравнивания скоростей при вводе и считывании информации из буфера. При работе программ системного ввода и/ режим спулинга – согласование скоростей на входе и выходе.

2) Управление заданиями. Состав. Функции. Взаимодействие с другими частями ОС.
Слежение за прохождением заданий от входа до выхода на всех этапах его выполнения



Программа системного ввода – планировщик 1 уровня

Командный интерпритатор берет входное задание

В том случае, если система считает, что есть ресурс для активизации задания, то активизируется планировщик 2-го уровня, то он выбирает задание с наивысшим приоритетом. В том случае, если система считает, что есть ресурс для активизации задания, то активизируется планировщик. Инициатор проверяет наличие ресурсов для выполнения задания. Если ресурсов нет, то задание сбрасывается. Если всё в порядке, то образуется задача или процесс. Как только сформирован TCB система должна его обязательно выполнить. При формировании PCB определяется программа подчиненная задаче, и данные, которые должна выполнить программа. Как только освобождается процессор, всплывает планировщик 3-го уровня, который обрабатывает TCB или PCB, ищет наиболее приоритетный процесс, который можно запустить

Система управления заданиями управляет прохождением заданий в ОС и выполняет функции:

1. Предоставление языковых средств управления работами в вычислительной системе.
2. Ввод и интерпретация заданий/команд.
3. Выделение и освобождение необходимых ресурсов.
4. Планирование заданий на выполнение.
5. Сбор и предоставление информации о состоянии заданий.

Существует три основных уровня планирования:

1. Планирование на верхнем уровне или планирование заданий.

На этом уровне осуществляется выбор заданий пользователем для выполнения и их запуск. Выбранные задания становятся готовыми процессами. Эту работу выполняет системный компонент - планировщик заданий.

2. Планирование на нижнем уровне или диспетчирование процессов.

Здесь осуществляется выбор готового процесса для выполнения, то есть предоставления ему ЦП. Выбранный процесс становится активным. Эту работу выполняет системный компонент - диспетчер.

3. Планирование на промежуточном уровне.

На данном уровне определяется, каким процессам будет разрешено состязаться за захват ЦП, то есть быть готовыми, и какие процессы будут кратковременно приостановлены (задержаны) для оптимизации загрузки системы. Промежуточное планирование управляет текущей производительностью вычислительной системы.

Эффективное планирование заданий и процессов является сложной проблемой, поскольку должно учитываться много противоречивых требований, таких как:

- справедливость;
- максимальная пропускная способность;
- приемлемое время ответа для максимального числа интерактивных пользователей;
- предсказуемость (задание должно выполняться примерно за одно время независимо от загрузки вычислительной системы);
- минимум накладных расходов на выполнение планирования;
- сбалансированность использования ресурсов;
- исключение бесконечного откладывания;
- учет приоритетов;
- отдавать предпочтение процессам, занимающие ключевые ресурсы;
- плавно деградировать при увеличении нагрузок.

Для того чтобы реализовать перечисленные требования, механизм планирования должен знать и учитывать следующие факторы:

- является ли процесс обменным (активно использующим операции ввода/вывода) или вычислительным (активно использующим процессор);
- является ли процесс пакетным или диалоговым;
- уровень реактивности интерактивного процесса;
- приоритетность процесса;
- частота прерываний по отсутствию нужной страницы;
- частота прерывания с низкого приоритета на высокий;
- длительность периода ожиданий ЦП процесса;
- суммарное использование времени ЦП и оценочное время, необходимое для завершения.

Для детального выделения действий, выполняемых при планировании, рассмотрим схему прохождения работ через ВС. При этом будем отслеживать прохождение задания с момента его ввода до полного завершения (вывода), фиксируя моменты изменения формы представления задания в ВС (Mi) и интервалы времени преобразования задания из одного состояния в другое (рис 4.2.)

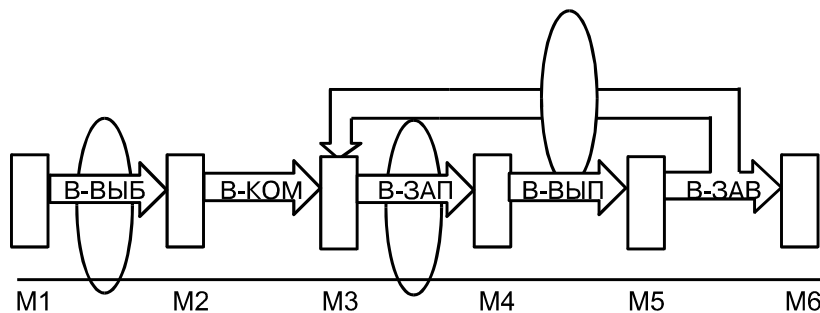


Рис. 4.2

Где:

- M1 — момент запроса: момент фиксации системой заявки/заявок на обслуживание;
- В-ВЫБ — время выборки: время предварительной обработки входного потока заявок;
- M2 — момент выборки: момент принятия решения об активизации заявок или отсрочке их выполнения;
- В-КОМ — время компиляции: время подготовки исходного описания задания-заявки и преобразования его в форму, необходимую для выполнения в параллельной системе;
- M3 — момент компоновки: момент, когда компиляция закончена, и входные задания переводятся в ранг "процессов", для которых ВС должна выделить ресурсы;
- В-ЗАП — время запуска: время, в течение которого выполняются действия, требуемые для запуска (инициирования) готовых процессов, — выделение ресурсов и перевод процессов в активное состояние;
- M4 — момент запуска: момент инициирования (активизации) задачи-процесса после распределения ресурсов (устройств, данных, памяти), который выполняется во время В-ЗАП.
- В-ВЫП — время выполнения: период времени, когда задания в системе активны, то есть выполняются.
- M5 — момент завершения: момент фиксации завершения задания (естественного или аварийного);
- В-ЗАВ — время завершения: время, в течение которого диспетчер ОС определяет, выполнено ли задание или его нужно продолжать выполнять. В последнем случае он ставит задание обратно в очередь на выполнение;
- M6 — момент выхода: момент, когда ОС считает задание полностью выполненным или устанавливает, что система не имеет возможности его дальнейшего выполнения, тогда оно удаляется из системы;
- Эллипс означает выполнение действий планировщика системы.
- заявок;
- требуется новый тип планировщика (назовем его ПТ — планировщик транспортный, рис. 4.17) для распараллеливания заданий, синхронизации процессов по данным — обеспечения поступления требуемых данных и поддержки связей между вычислительными узлами при реализации связи по данным;
- обработанные задания (или их распараллеленные модули) транслятором ОС и ПТ ставятся в новую очередь — буфер БУФ;
- к ПП добавляется функция *адаптирования*, при выполнении которой задания распределяются соответственно особенностям данной системы (например: специальные схемы для систем гиперкуб, транспьютерных систем или дополнительная схема для неоднородной среды).
- к ПН добавляется функция балансирования в случае реконфигурации (отказа некоторых элементов) системы.

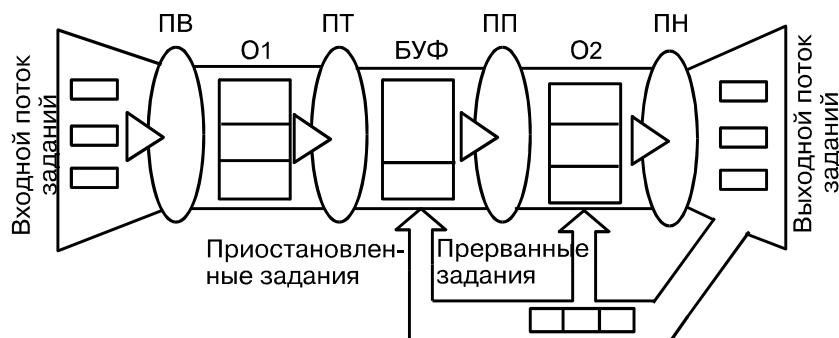


Рис. 4.17. Схема прохождения заданий через ВС

Полную систему планирования в этом случае можно представить в виде семиуровневой модели (табл. 4.1), где каждый уровень часто рассматривают как отдельную задачу. Порядок выполнения уровней может быть изменен в зависимости от особенности системы и целей задачи планирования.

Табл. 4.1

ПВ	1	Предварительное входное планирование исходного потока заявок, претендующих на захват ресурсов вычислительной системы	Задача Ввода
	2	Структурный анализ взаимосвязи входного потока заявок по ресурсам и определение общих ресурсов	Задача анализа
ПТ	3	Структурный анализ заявок и определение возможности распараллеливания каждой работы	Задача распараллеливания
ПП	4	Адаптирование распределения работ соответственно особенностям вычислительной системы	Задача адаптирования
	5	Составление расписания выполнения взаимосвязанных процедур: оптимизация плана по времени решения, количеству используемых ресурсов и количеству пересылок	Задача оптимизации
	6	Планирование потока процессов, претендующих на захват времени процессора/процессоров вычислительной системы	Задача распределения
ПН	7	Выделение времени процессоров ВС активизированным процессам, перераспределение (реконфигурация) работ в вычислительной среде (отказ оборудования)	Задача распределения и перераспределения

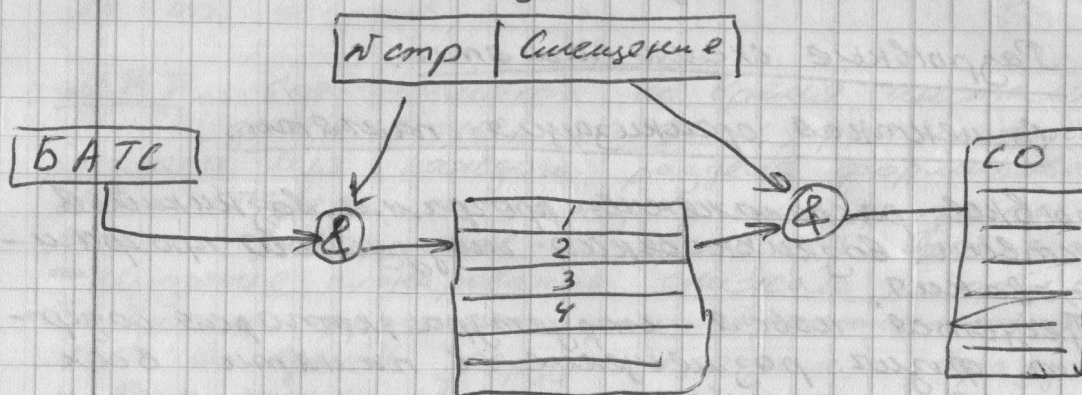
3) Страничная организация памяти. Виды, особенности.

Страничная организация памяти

Относится к разрывн. памяти. Разделить память на блоки фиксир. размера. Програм. может занимать столько блоков, сколько

ей надо.

Распределение (соответствие между страницами и блоками) находится в таблице страниц. Особенности: номер страницы кратен двум (или удваивается одно сложение при формировании исполнительного адреса.)



Чистая страничная организация. Все страницы прог. находятся в памяти.

Странич. организ. по запросу.

Таблица стр. на всю программу, но в строке есть бит наличия. Только если страничка прерывалась (0 → подгрузка → 1) Страница не выгрузилась.

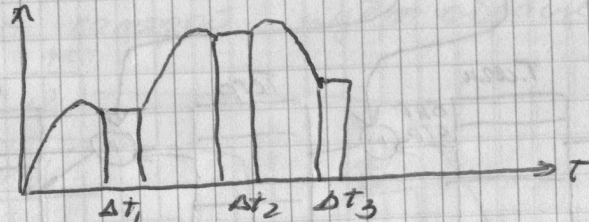
Виртуальная организация памяти.

Позразумевает работу сис-ми с программами, которые имеют объем, значит, превышает реальный размер памяти. Размер прог. ограничен только адресацией. Отсюда возник. проблема:

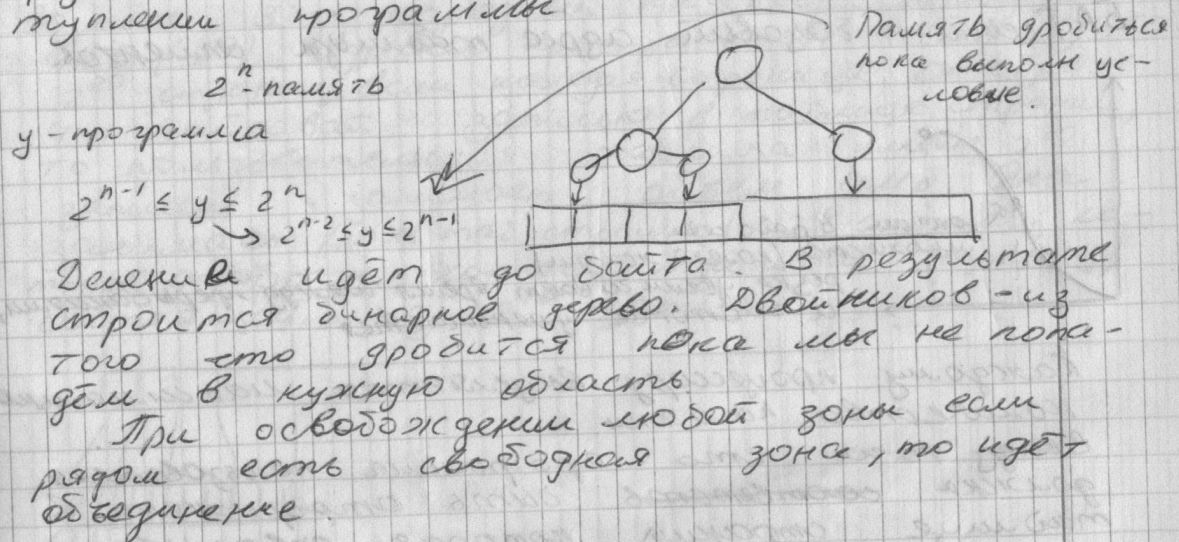
- 1. какая часть прог. должна наход. в памяти.
- 2. какую стр. одной и той же прог. выгрузить из памяти (заместить)

(1) - проблема размещения
(2) - проблема замещения.

Для проблемы размещения использ. теория
рабочих моментов. Если выполн. i -тая ко-
манда, то с большой вероятн. след. будет
выполн. $i+1$ команду. Если исполн. $i+1$,
то с большой вероятн. след. угадет вре-
мени будет выполн. следующим. Этот
блок команду (Принцип временной локальности)
В кэше находится активная рабочая
зона.



Алгоритм двоичного память изготавлю
представл как один большой раздел. Три пос-
тупления программы



Кроме страничной организации, которая соо-
рентир на повышение эффективн. рабо-
ты памяти. и сегментная организация
для эффективной работы пользователя.

4) Особенности управления файлами в ОС UNIX.

Ответ:

Доп.инфа:

Недостаток способа хранения расположения файла в системе UNIX.

Индексные Дескрипторы — это объект Unix, который ставится во взаимнооднозначное соответствие с содержимым файла. То есть для каждого ИД существует только одно содержимое и наоборот, за исключением лишь той ситуации, когда файл ассоциирован с каким-либо внешним устройством. Напомним содержимое ИД:

- поле, определяющее тип файла (каталоги и все остальные файлы);

- код привилегии/защиты;
- количество ссылок к данному ИД из всевозможных каталогов файловой системы;
- (нулевое значение означает свободу ИД)
- длина файла в байтах;
- даты и времена (время последней записи, дата создания и т.д.);
- поле адресации блоков файла.

Как видно — в ИД нет имени файла. Давайте посмотрим, как организована адресация блоков, в которых размещается файл.

В поле адресации находятся номера первых десяти блоков файла, то есть если файл небольшой, то вся информация о размещении данных файла находится непосредственно в ИД. Если файл превышает десять блоков, то начинает работать некая списочная структура, а именно, 11й элемент поля адресации содержит номер блока из пространства блоков файлов, в которых размещены 128 ссылок на блоки данного файла. В том случае, если файл еще больше — то используется 12й элемент поля адресации. Суть его в следующем — он содержит номер блока, в котором содержится 128 записей о номерах блоках, где каждый блок содержит 128 номеров блоков файловой системы. А если файл еще больше, то используется 13 элемент — где глубина вложенности списка увеличена еще на единицу.

Таким образом мы можем получить файл размером $(10+128+128^2+128^3)*512$.

Достоинства и недостатки файловой системы UNIX

Блок начальной загрузки	Суперблок	Индексные дескрипторы	Блоки файлов	Область сохранения
0				N-M+1

Суперблок файловой системы - содержит оперативную информацию о состоянии файловой системы, а также данные о параметрах настройки файловой системы. В частности суперблок имеет информацию о

- количестве индексных дескрипторов (ИД) в файловой системе;
 - размере файловой системы;
 - свободных блоках файлов;
 - свободных ИД;
 - еще ряд данных, которые мы не будем перечислять в силу уникальности их назначения
- Вот информация о суперблоке. Какие можно сделать выводы и замечания?

- суперблок всегда находится в ОЗУ; (Недостаток)
- все операции по освобождению блоков, занятию блоков файлов, по занятию и освобождению ИД происходят в ОЗУ (минимизация обменов с диском). Если же содержимое суперблока не записать на диск и выключить питание, то возникнут проблемы (несоответствие реального состояния файловой системы и содержимого суперблока). Но это уже требование к надежности аппаратуры системы.

Свойство файловой системы по оптимизации доступа, критерием которого является количество обменов, которые файловая система производит для своих нужд, не связанных с чтением или записью информации файлов.

(Преимущество)