БИЛЕТ № 15

1) Особенности задач, решаемых в распределенных ОС.

Конспект

Распределенная система подразумевает наличие ресурсов, которые для пользователей как один ресурс.
Характеристики распределенных систем: прозрачность, масштабируемость, открытость.
Подключение дополнительных устройств для сохранения характеристики системы.
Прозрачность доступа – обеспечение прозрачности разности представления данных.
Прозрачность местоположения – ресурс может находиться, где угодно.
Прозрачность переноса ресурса – изменение местоположения ресурса, а пользователь этого не знает.
Прозрачность репликации – скрыто, что пользуетесь копией, а не оригиналом.
Прозрачность параллельного доступа – система работает с разделяемыми данными.
Прозрачность отказа – скрыт отказ ресурса.
Прозрачность по сохранности данных – пользователь не знает, где находятся данные (в какой памяти).

Доп. Инфа. Свойства распределенных операционных систем:

- Отсутствие общей памяти приводит к тому, что нельзя определить общее состояние системы с помощью множества совместных переменных, а невозможность совместного обращения к памяти и различия в задержке передач сообщений приводят к тому, что при определении состояния какого-либо элемента системы из двух различных точек можно получить разные результаты, в зависимости от порядка предшествующих событий;

- Распределенная система распределяет выполняемые работы в узлах системы, исходя из соображений повышения пропускной способности всей системы;

- Распределенные системы имеют высокий уровень организации параллельных вычислений.

Два фактора увеличивают вероятность отказов отдельных элементов (по сравнению с централизованными системами):

1. Большое число элементов системы.

2. Надежность систем связи обычно меньше, чем надежность информационных систем.

Однако надежность всей системы в целом в распределенных операционных системах весьма высока за счет специфических свойств, позволяющих исключить или ослабить эффект отказа отдельных элементов системы (исправление вышедших из строя элементов, переконфигурация системы и т.д.).

2) Методы повышения эффективности организации вычислительного процесса ВС.

Приемы, методы повышения эффективности вычислительной машины(за счет организации выч. процесса):
- применение дополнительных уровней памяти(многоуровневая память).
- просмотр команд вперед(подкачка). Буфер – либо одна самая длинна, либо 6 коротких
- секционирование памяти
- конвейерный принцип
- многопрограмный режим работы

3) Методы доступа к файлам. Особенности решаемых задач.

Доступ к файлам

В старых операционных системах предоставлялся только один тип доступа к файлам — **последовательный доступ**. В этих системах процесс мог читать байты или записи файла только по порядку от начала к концу. Такой доступ к файлам появился, когда дисков еще не было и компьютеры оснащались магнитофонами. Поэтому даже в дисковых операционных системах при последовательном доступе к файлу имитировалось его чтение или запись на накопителе на магнитной ленте с возможностью многократной перемотки назад.

С появлением дисков стало возможным читать байты или записи файла в произвольном порядке или получать доступ к записям по ключу. Файлы, байты которых могут быть прочитаны в произвольном порядке, называются **файлами произвольного доступа**. Такие файлы используются многими приложениями.

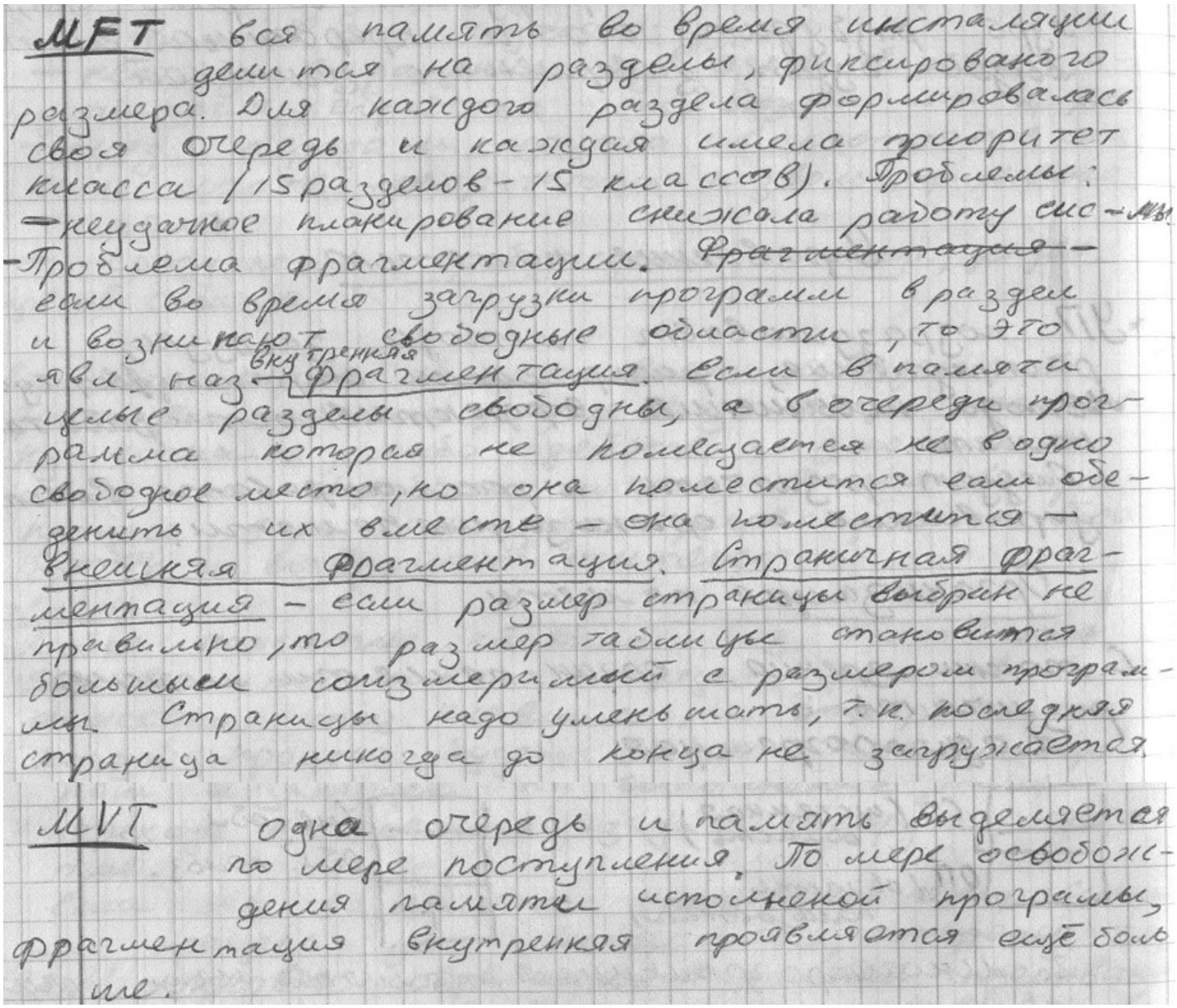
Файлы произвольного доступа очень важны для многих приложений, например для баз данных. Если клиент звонит в авиакомпанию с целью зарезервировать место на конкретный рейс, программа резервирования авиабилетов должна иметь возможность получить доступ к нужной записи, не читая все тысячи предшествующих записей, содержащих информацию о других рейсах.

Для указания места начала чтения используются два метода. В первом случае каждая операция read задает позицию в файле. При втором способе используется специальная операция поиска seek, устанавливающая текущую позицию. После выполнения операции seek файл может читаться последовательно с текущей позиции.

В некоторых старых операционных системах, использовавшихся на мэйнфреймах, способ доступа к файлу (последовательный или произвольный) указывался в момент создания файла. Это позволяло операционной системе применять различные методы для хранения файлов разных классов. В современных операционных системах такого различия не проводится. Все файлы автоматически являются файлами произвольного доступа.

4) Фрагментация, методы борьбы с ней.

Ответ:



Доп.инфа:

---

5) Виды прерываний, особенности их обработки.

Прерывание — это нарушение последовательности выполнения действий (команд), т.е. после текущего действия (команды) выполняется не следующее (команда), а некоторое другое действие.

Классы прерываний

По своему назначению, причине возникновения прерывания делятся на различные **классы**. Традиционно выделяют следующие классы:

1. *Прерывания от схем контроля машины*. Возникают при обнаружении сбоев в работе аппаратуры, например, при несовпадении четности в микросхемах памяти.
2. *Внешние прерывания*. Возбуждаются сигналами запросов на прерывание от различных внешних устройств: таймера, клавиатуры, другого процессора и пр.
3. *Прерывания по вводу/выводу*. Инициируются аппаратурой ввода/вывода при изменении состояния каналов ввода/вывода, а также при завершении операций ввода/вывода.
4. *Прерывания по обращению к супервизору*. Вызываются при выполнении процессором **команды обращения к супервизору** (вызов функции операционной системы). Обычно такая команда инициируется выполняемым процессом при необходимости получения дополнительных ресурсов либо при взаимодействии с устройствами ввода/вывода.
5. *Программные прерывания*. Возникают при выполнении **команды вызова прерывания** либо при обнаружении ошибки в выполняемой команде, например, при арифметическом переполнении.

В последнее время принято прерывания 4 и 5 классов объединять в один класс программных прерываний, причем, в зависимости от источника, вызвавшего прерывание, среди них выделяют такие подтипы:

- прерывание вызванное исполнением процессором *команды перехода к подпрограмме обработки прерывания*
- прерывания, возникающие в результате *исключительной* (аварийной) *ситуации* в процессоре (деление на "0", переполнение и т.д.).

В связи с многообразием различных ВС и их постоянным развитием меняется и организация системы прерываний. Так, с появлением виртуальной памяти, появился класс страничных прерываний, который можно отнести и к классу исключительных ситуаций в процессоре; в системах с кэш-памятью существуют прерывания подкачки страниц в кэш-память и т.д.

Во время выполнения обработчика одного из прерываний возможно поступление другого сигнала прерывания, поэтому система должна использовать определенную *дисциплину обслуживания заявок* на прерывания. Обычно различным классам либо отдельным прерываниям присваиваются различные *абсолютные приоритеты*, т.о. при поступлении запроса с высшим приоритетом текущий обработчик снимается, а его место занимает обработчик вновь поступившего прерывания. Количество уровней вложенности прерываний называют **глубиной системы прерываний**. Обработчики прерываний либо дисциплина обслуживания заявок на прерывание должны иметь также специальные средства для случая, когда при обработке прерывания возникает запрос на обработку прерывания от того же источника.

((((((((((((((((((((((((С ШПОР

В системе присутствует 2 приоритетных уровня:

- Приоритет сигналов
  Используется полноупорядоченная схема сигналов (контроллер, его приоритет определяется путём запуска команды смены прерывания) либо частичноупорядоченная (когда прерываний много – все они разбиваются на классы, а потом производиться сканирование в полносвязной системе).
- Приоритет прерывания программ
  Более важный уровень, на нём выбирается программа с наивысшим приоритетом.

Дополнительные классы прерываний:

- Прерывания по таймеру
- Прерывания по кэш-промаху
- Страничное прерывание
- Прерывание по нажатию клавиши

))

РЕАЛИЗАЦИЯ МЕХАНИЗМА ПРЕРЫВАНИЙ

В КОМПЬЮТЕРАХ ТИПА IBM PC

В компьютерах семейства IBM PC, построенных на основе микропроцессора Intel 80x86 и использующих операционную систему MS-DOS, механизм прерываний реализуется как при помощи аппаратных, так и при помощи программных средств.

Микропроцессор поддерживает обработку 256 различных прерываний, каждое из которых идентифицируется номером в диапазоне 00h-FFh. Сегментные адреса обработчиков прерываний, называемые **векторами прерываний**, содержатся в **таблице векторов прерываний**, которая располагается в начале оперативной памяти (по адресу 0000:0000). Каждый вектор прерывания имеет размер 4 байта, таким образом, таблица векторов занимает 1024 байта. Для определения адреса обработчика любого прерывания нужно номер прерывания умножить на 4.

В защищенном режиме старшие модели семейства процессоров вместо таблицы векторов используют **таблицу дескрипторов прерываний,** сходную по своему назначению с таблицей векторов.

Операционная система не контролирует содержимое таблицы векторов, хотя и имеются средства для чтения и изменения векторов. При помощи функций MS-DOS значение векторов можно изменить, поместив нужный адрес непосредственно в таблицу векторов прерываний. В этом случае контроль за содержимым векторов и их соответствием размещению обработчиков в памяти полностью ложится на программиста, а последствия в случае ошибки могут быть самыми непредсказуемыми.

Активизация обработчика прерывания может произойти в результате возникновения следующих ситуаций:

- возникновение сигнала внутреннего прерывания внутри микросхемы процессора (**внутренние** или **логические прерывания**)
- поступление в процессор сигнала запроса на прерывание от внешнего (по отношению к процессору) устройства (**аппаратные прерывания**)
- выполнение процессором команды INT вызова процедуры обработки прерывания (**программные прерывания**)

2.9.1. Внутренние прерывания

Для обслуживания внутренних прерываний микропроцессоры i8086 использовали 5 первых векторов прерываний (00h-04h). Прерывания с номерами 05h-31h фирма Intel зарезервировала для использования в дальнейших разработках, однако фирма IBM при создании IBM PC использовала эти вектора по своему усмотрению. В последующих реализациях процессоров эти же вектора были задействованы для обработки новых внутренних прерываний, в результате чего стали возможны конфликты. Поскольку все добавленные прерывания используются в защищенном режиме процессора, то для избежания конфликтов при переходе в защищенный режим контроллер прерываний перепрограммируется таким образом, что при поступлении сигнала на линию IRQ0 вызывается процедура обработки с номером, отличным от 08h (обычно 50h или 60h). Для последующих линий IRQ вызываются процедуры с последующими (относительно базового для IRQ0) номерами.

Сигналы внутренних прерываний возникают при нарушениях в работе самого микропроцессора либо при ошибках в выполняемых командах и формируются внутри схемы микропроцессора при возникновении одной из ситуаций, указанных в Табл. 2.2.

2.9.2. Аппаратные прерывания

Аппаратные прерывания инициируются различными устройствами компьютера, внешними по отношению к процессору (системный таймер, клавиатура, контроллер диска и пр.). Эти устройства формируют сигналы запросов на прерывания (IRQ), поступающие на **контроллер прерываний**.

. Немаскируемые прерывания

Существует одно аппаратное прерывание, отличное от всех остальных. Это **немаскируемое прерывание** *(Non-Maskable Interrupt, NMI).* Его отличие состоит в том, что хоть инициатором его и является внешнее устройство, сигнал запроса поступает в процессор, минуя контроллер прерываний. Поэтому такой запрос на прерывание невозможно замаскировать. Кроме этого, запрос по линии NMI вызывает немедленную реакцию процессора и имеет максимальный приоритет в системе, хотя обработчик NMI может быть прерван любым маскируемым прерыванием, если их разрешить командой STI. Это позволяет использовать данное прерывание в особо критических ситуациях: обычно оно используется при падении напряжения питания и при ошибках памяти, однако на некоторых моделях IBM — совместимых компьютеров NMI используется также для обслуживания сопроцессора, клавиатуры, каналов ввода/вывода, дисковых контроллеров, часов реального времени, таймеров, контроллеров прямого доступа к памяти и пр.

В процессорах i80286 и выше в случае, если во время обработки NMI возникнет повторный запрос на немаскируемое прерывание, он не прервет выполнение текущего обработчика, а запомнится и будет обработан после завершения текущего обработчика. Если повторных запросов во время выполнения обработчика будет несколько, то сохранится только первый, последующие будут проигнорированы. При возврате из обработчика командой IRET процессор не переходит на выполнение следующей команды, а пытается выполнить команду по тому же адресу, что привел к возникновению критической ситуации.

2.9.2.3. Программные прерывания

Программные прерывания инициируются специальной командой процессору INT n, где n указывает номер прерывания, обработчик которого необходимо вызвать. При этом в стек заносится содержимое регистров флагов, указателя команд и сегмента кода, а флаг разрешения прерывания сбрасывается (аналогично тому, как при обработке аппаратных прерываний).

Далее в процессор загружается новое PSW, значение которого содержится в векторе прерывания с номером n, и управление переходит к обработчику прерывания. Требования к обработчику программного прерывания такие же, как и к обработчику аппаратного прерывания, за исключением того, что команда завершения аппаратного прерывания (EOI) не требуется. Обработка прерывания завершается командой IRET, восстанавливающей из стека старое PSW.

Обработка программного прерывания является более мягким видом прерывания по отношению к прерываемой программе, нежели обработка аппаратного прерывания, т.к. аппаратное прерывание выполняется по требованию самой

прерываемой программы, и она ожидает от этого какого-либо результата. Аппаратное же прерывание получает управление безо всякого ведома выполняющейся программы и зачастую не оказывает на нее никакого влияния.