Операционные системы

Курс лекций для гр. 4057/2

Лекция №12

Вопросы к лекции 11

1. В каких ситуациях в параллельных алгоритмах удобно использовать барьеры?
2. Дайте неформальное обоснование полезных свойств маркерного алгоритма (взаимоисключение, отсутствие взаимоблокировок и голодания) и оцените, сколько всего сообщений требуется послать в системе из N узлов, чтобы процесс мог войти в свой критический участок.
3. То же для алгоритма «маркерное кольцо»
4. Как модифицировать этот алгоритм для случая ненадежной связи? - Связь с любым узлом может в любой момент прекратиться из-за отказа узла или линии связи.

Маркерный алгоритм

1. Маркер - массив M, где M[k] – временная метка: когда маркер последний раз находился в k-ом процессе
2. У каждого процесса – массив запросов R, элемент R[j] – временная метка сообщения - запроса от j-го процесса
3. Инициализация: маркер присваивается произвольному процессу
4. Перед входом в критический участок, если у процесса нет маркера, он рассылает всем другим запрос с временной меткой и ждет прихода маркера
5. После выхода из критического участка процесс Pj просматривает массив запросов в следующем порядке индексов: j+1, j+2, …, 1, 2, …, j -1, пока не найдет первый элемент R[k] такой, что R[k] >M[k], т.е. процесс Pk сделал запрос после последнего пребывания в нем маркера. Этому процессу Pk и посылается маркер.

Содержание

Основные понятия

Надежность (dependability) - это способность системы к долговременному правильному функционированию

Это комплексное свойство с двумя составляющими:

– Безотказность (reliability) – долговременное соответствие требованиям (спецификации)

– Отказоустойчивость (fault tolerance) – способность продолжать нормальное функционирование после отказов программ или аппаратуры

Отказ (failure) – нарушение работоспособности изделия и его соответствия требованиям технической документации

Программный отказ – это проявление ошибки в программе;

это неспособность функциональной единицы системы, зависящей от программы, выполнять требуемую функцию в заданных пределах (стандарт IEEE/ANSI).

Причины отказов

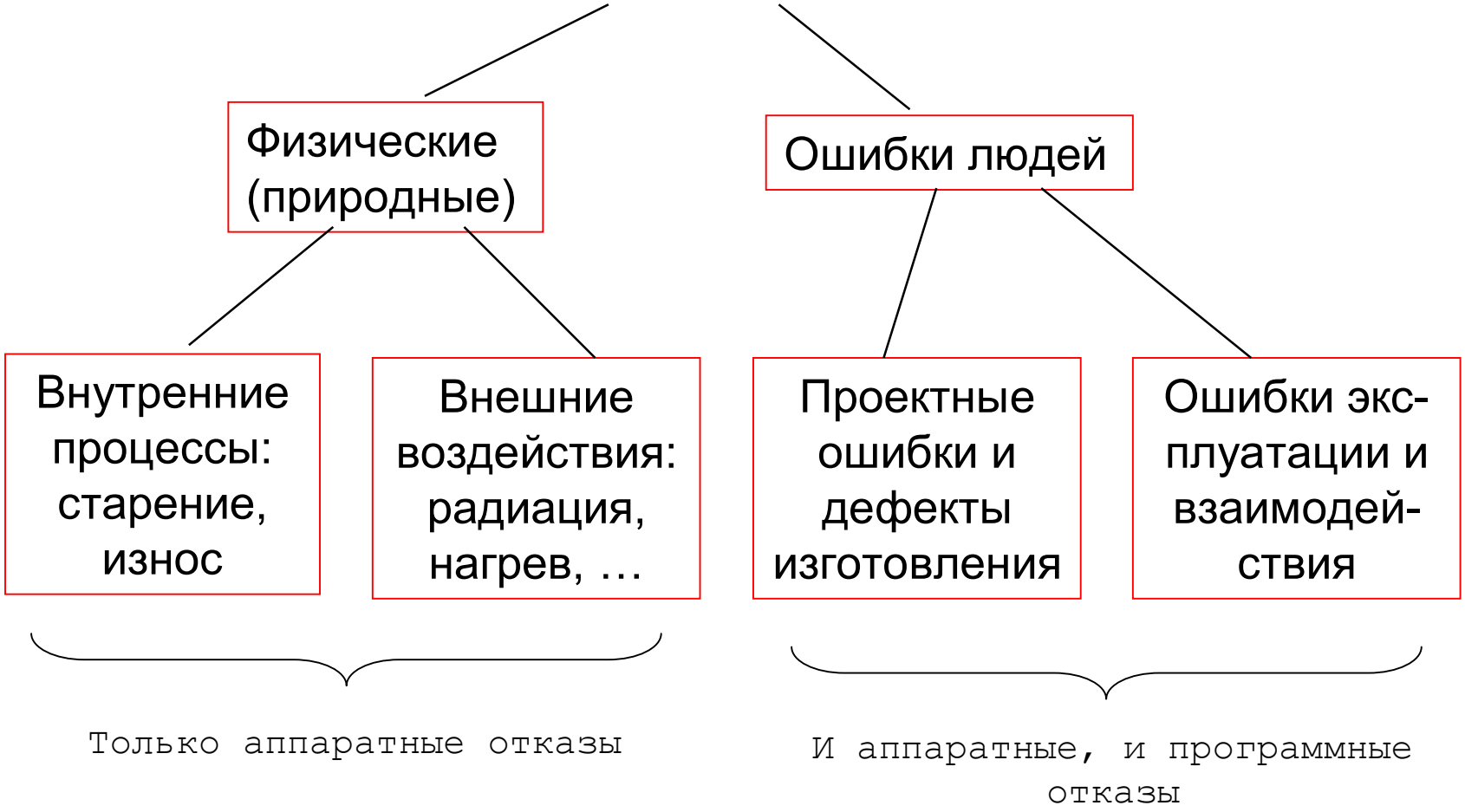
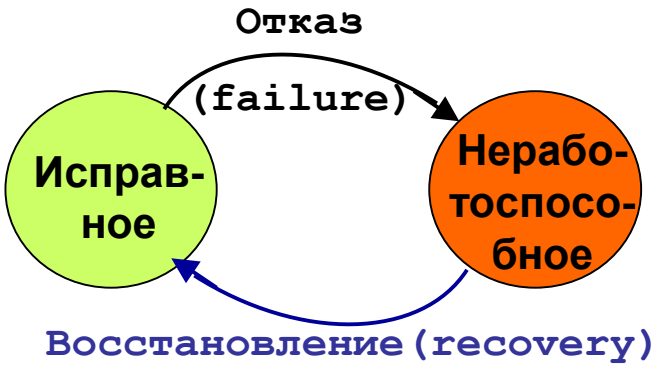


Диаграмма состояний при отказе

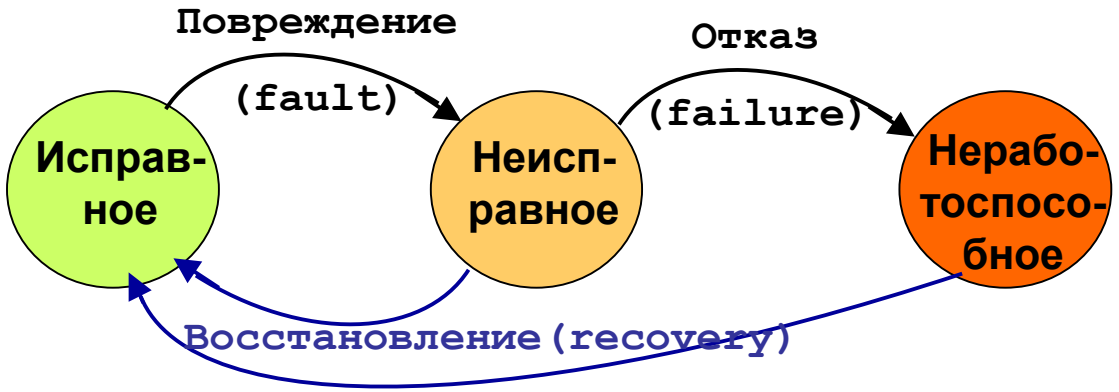


Исправное (работоспособное) состояние: нормальная работа

Неработоспособное состояние: в результате отказа система не выполняет функцию, специфицированную в технической документации

- Аппаратный отказ – проявление неисправности в аппаратуре
  - «Сгорел» транзистор; альфа-частица повлияла на p-n переход, …
- Программный отказ - проявление ошибки в программе
  - Неправильный оператор -> ошибка в коде -> неверный результат
  - Ошибка пользователя или неверные входные данные, не обнаруженные программой

Детализация диаграммы состояний

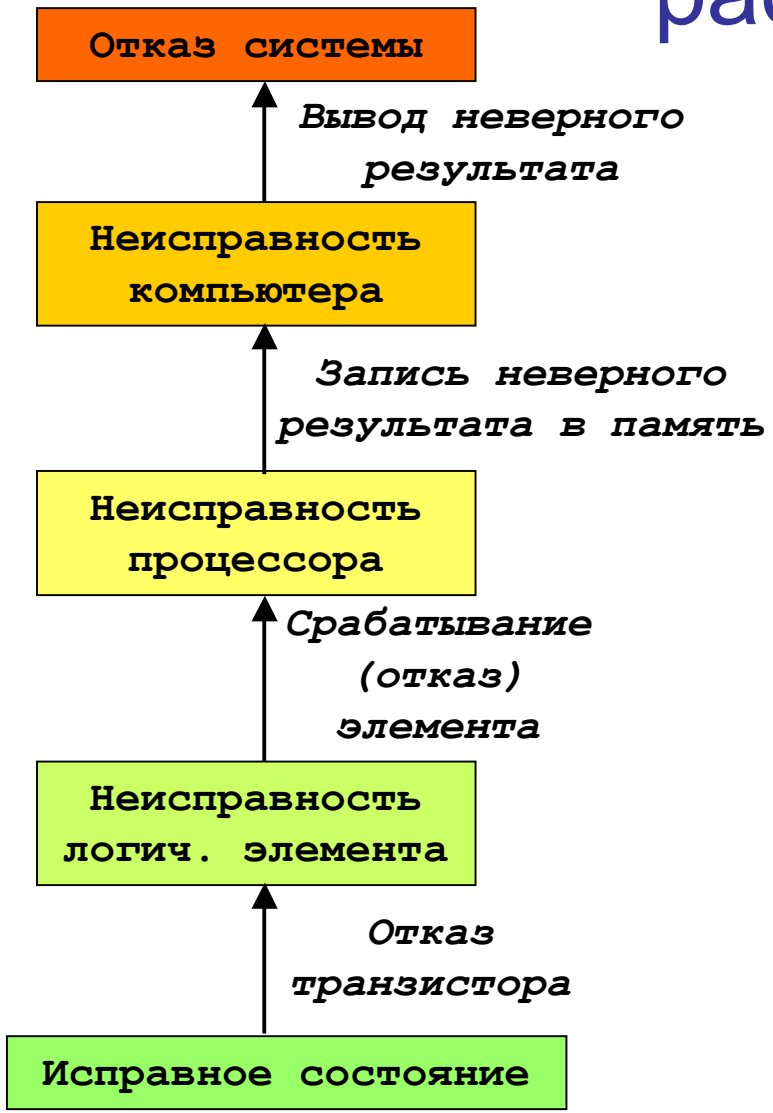


Неисправное состояние - промежуточное, когда в результате повреждения (fault) неисправность уже появилась, но еще не проявилась вовне в виде отказа

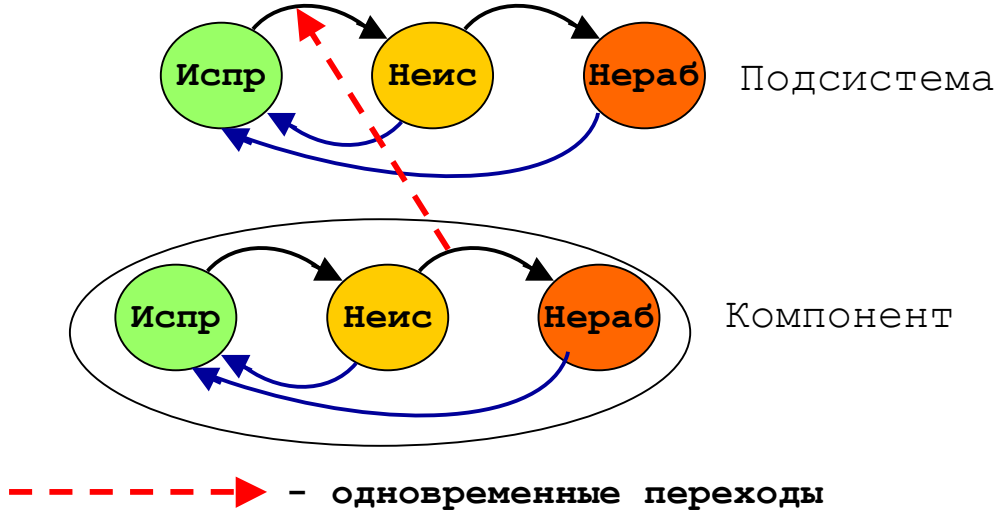
Напр., из-за отказа ячейки памяти исказилось ее содержимое (данные), но оно еще не прочитано и эта неисправность не повлияла на результат вычислений

Время нахождения системы в неисправном, но еще работоспособном состоянии называется латентным (скрытым) периодом отказа

Восходящее каскадное распространение отказа

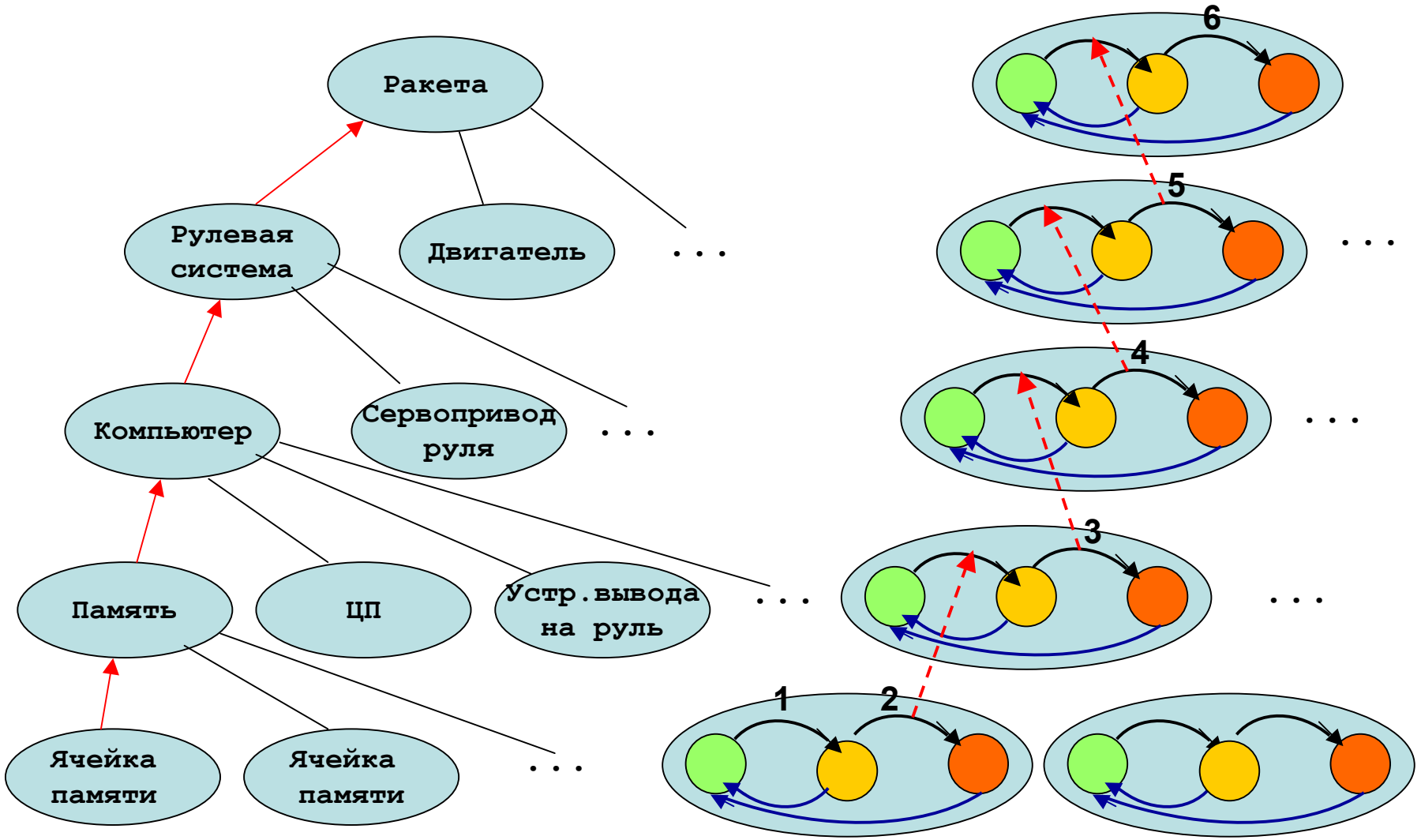


Отказ компонента - это повреждение подсистемы:



Отказ на самом верхнем уровне – отказ системы – называется аварией (crash)

Иерархия структуры объекта и отказов



(Вопрос 1)

1 – искажено содержимое ячейки памяти
2 – неверное содержимое ячейки считано ЦП
3 – вычислен неверный результат, …, 6- авария

Отказы и сбои

- Восстановление (recovery) - это возврат в исправное состояние путем:
  - а) ручного ремонта, замены, исправления
  - б) автоматически - задействованием резервов (средства отказоустойчвости)
  - в) самопроизвольно (обычно быстро, за доли сек.)
- В случае в) отказ называется сбоем (transient fault), остальные отказы называются устойчивыми
  - По умолчанию отказ - устойчивый
- Сбои происходят на порядок чаще устойчивых отказов
  - Их причины в электронной аппаратуре - флюктуации питания, ситуации "гонок" сигналов, альфа-частицы (радиация) и др.
  - В программах как сбои проявляются время-зависимые ошибки – их иногда называют "мерцающими" (blinking bugs)
    - ✓ «Самовосстановление» выглядит в этом случае как правильное поведение при повторном выполнении отказавшей функции.

Сбой - это кратковременный самоисправляющийся отказ

NB: в прессе сбоем часто называют любой отказ, но это неправильно!

Количественные характеристики надежности

Характеристики безотказности:

- Интенсивность отказов (failure rate) – среднее их количество в единицу времени
- Среднее время безотказной работы (MTBF – Mean Time Between Failures)

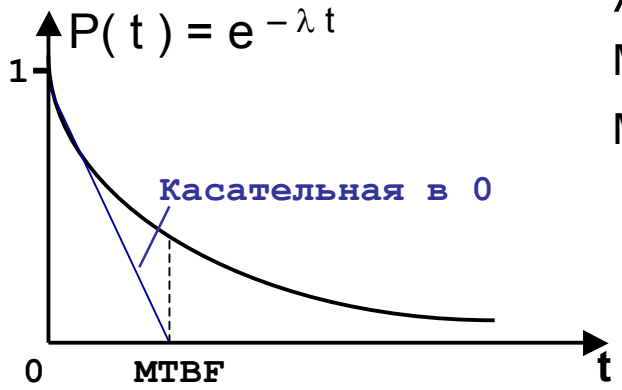
Сводная характеристика надежности (безотказность + скорость восстановления):

- Коэффициент готовности (availability)

В предположении случайного *простейшего потока* отказов:

- ✓ т.е., отказы независимы, редки и их вероятность неизменна во времени

P(t) – функция распределения вероятности безотказной работы в течение периода времени t – описывается законом Пуассона (экспоненциальной функцией распределения вероятностей):



λ - *интенсивность отказов* (обычно в 1/час)

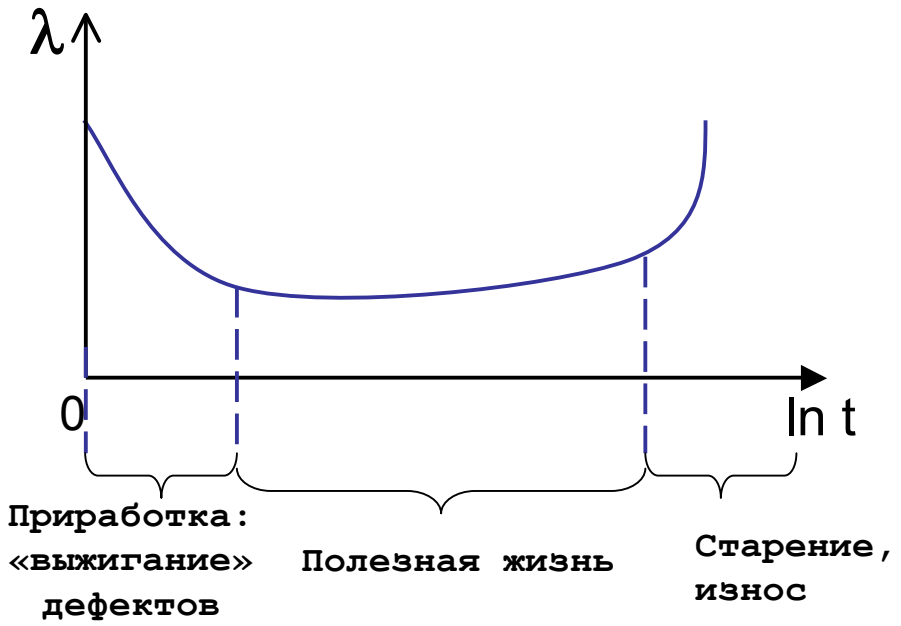
Математическое ожидание

M (P) = MTBF = $1/\lambda$ (результат интегрирования $e^{-\lambda t}$)

В теории надежности технических систем это основная модель потока отказов, простая и хорошо соответствующая реальности

Изменение интенсивности отказов во времени

(Характерно для всех видов технических систем)



Характеристики безотказности компонентов вычислительных систем

(Примеры и средние значения)

Вид компонента	λ 1/час	MTBF, час	MTBF, лет
Обычная электромеханическая аппаратура (вентилятор)	10^{-3}	10^{3}	10^{-1}
Обычная электронная аппаратура (блок питания)	10^{-4}	10^{4}	1
Большие интегральные схемы – БИС (ЦП)	$10^{-6} – 10^{-8}$	$10^{6} – 10^{8}$	$10^{2} – 10^{4}$
Электронная аппаратура на БИС (материнская плата)	10^{-5}	10^{5}	10^{1}
Программы общего назначения (Windows, Office)	10^{-2}	10^{2}	10^{-2}
Высоконадежные программы (для критических применений)	10^{-5}	10^{5}	10^{1}

(Вопросы 1-3)

Надежность дисковой памяти

Одно из наименее надежных устройств компьютера:
электромеханика -> износ движущихся частей, поломки деталей

Среднее время безотказной работы магнитного диска

- Объявленное производителем – порядка 10^6 часов
  - Напр., Seagate Barracuda (180 Гбайт): MTBF =1 200 000 часов, т.е. в среднем один отказ за 140 лет
- Реальная статистика работы, снятая на 10^5 дисков разных типов:
  - Диски повышенной надежности: 9 - 11 лет
  - Обычные диски: 5 – 7 лет

Твердотельный (флэш) диск: производители объявляют
MTBF= 300000 час = 34 года

Статистика надежности кластеров Google

Типичный первый год для нового кластера из ~3000 серверов:

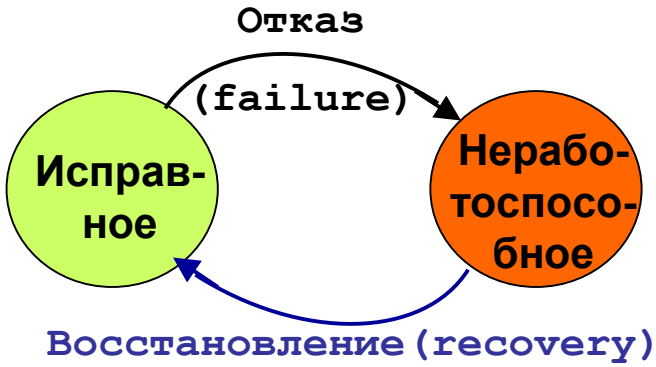
~0.5 отказа всего кластера из-за перегрева (~1-2 дня ремонта)
~1 отказ модуля связи с интернетом (~6 часов ремонта)
~1 переналадка сети (~5% серверов отключаются на пару дней
~20 отказов стоек (40-80 серверов в каждой, 1-6 часа ремонта)
~5 сбоев стоек (40-80 серверов теряют 50% пакетов данных)
~8 отказов локальной сети
~12 перезагрузок маршрутизаторов
~3 отказа маршрутизаторов (потеря трафика на 1 час)
~ десятки 30-секундных сбоев DNS
~1000 сбоев/отказов отдельных серверов
~ несколько тысяч сбоев/отказов магнитных дисков

(Вопрос 4)

Требования к надежности критически важных систем

- Пример Mission-critical системы: бортовая система управления космическим зондом Pioneer-10
- Было поставлено требование $\lambda = 10^{-9}$, чтобы вероятность устойчивого отказа в первые 10 лет работы была не более 10^{-4} (или вероятность безотказной работы 0,9999) т.е. MTBF = 100 тысяч лет!
- Сигналы этого зонда, запущенного в 1972 г., до сих пор принимаются на Земле, хотя он уже давно покинул пределы солнечной системы

Ремонтопригодность



Если произошел отказ, то время восстановления должно быть минимальным !

Интегральная характеристика надежности - коэффициент готовности, или просто «готовность»:

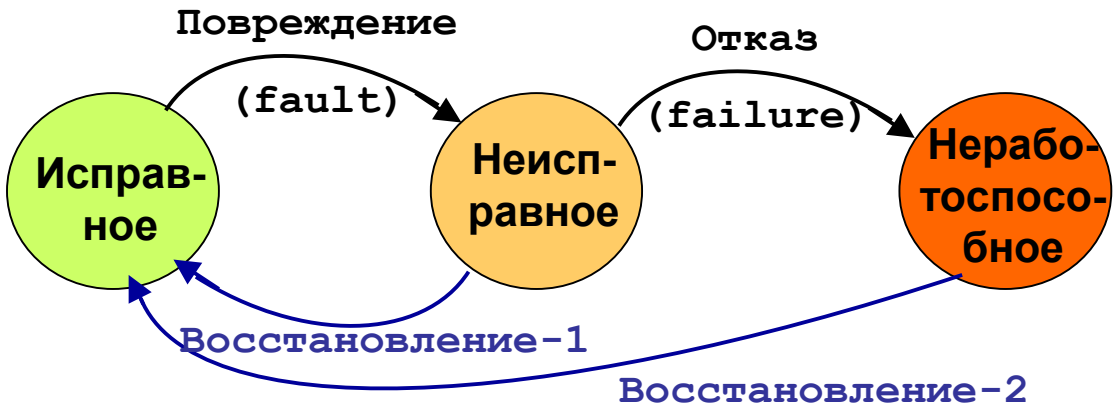
$$k = (T - T_{пр}) / T$$

где T – общее время работы, Тпр – суммарное время простоев в неработоспособном состоянии

- Обычный Windows -сервер имеет k = 0,999 (в среднем 10 часов простоя в год)
- В особо ответственных системах требуется k = 0,99999 (5 мин в год)
- Для критически важных систем:
  - для цифровых телефонных станций – 2 часа простоя за 15 лет
  - для системы управления воздушным движением – 3 сек за год

NB:Англ. «готовность» – availability – неверно переводят как «доступность»

Идеальная отказоустойчивость



Если *всегда* выполнять восстановление -1, пока отказ находится в латентном периоде - тогда повреждение *никогда* не приведет к отказу системы

Для этого нужно постоянно контролировать исправность компонентов и моментально обнаруживать их повреждения

«Fault tolerance» буквально = «устойчивость к повреждениям»

Методы отказоустойчивости

Отказоустойчивость (ОУ, fault tolerance) – это способность быстрого автоматического восстановления системы после отказа

Принципы ОУ

- Средства ОУ – общие для аппаратных и программных отказов
  - ✓ часто даже не производится диагностика, где именно произошел отказ
- ОУ всегда достигается путем введения избыточности (redundancy), т.е. резервирования компонентов и/или времени

Фазы управления ОУ

1. Контроль исправности – обнаружение отказавших элементов путем:
   - фоновый прогон тестов аппаратуры или программ
   - контрольные суммы кодов (в т.ч. контроль четности)
   - дублирование аппаратуры / данных со сравнением копий
   - контроль в программах с помощью утверждений (asserts) и исключительных ситуаций (exceptions)
2. Локализация области повреждения при обнаружении неисправности
   - в пространстве: область памяти, компоненты, устройства
   - во времени, поскольку точный момент повреждения может быть неизвестен
3. Восстановление путем подключения резервов

Виды резервирования для контроля и восстановления

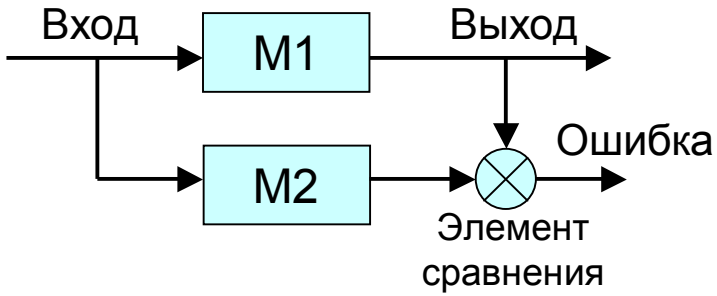
1. Структурное: аппаратная избыточность – дублирование или многократное резервирование модулей аппаратуры
   - Горячий резерв - постоянно включенный и работающий
   - Холодный – включается на замену отказавшего модуля
2. Информационное: избыточность в представлении данных:

   а) избыточное кодирование
   - коды с обнаружением ошибки (четность, циклическая сумма)
   - коды с исправлением ошибки (код Хэмминга, циклический код)

   б) дублирование или многократное резервирование данных на дисках
3. Программное (program diversity): аналог структурного, но резервные модули - не аппаратные, а программные:
   - разработаны независимыми командами
   - и/или выполняют упрощенные версии алгоритмов → постепенное ухудшение (graceful degradation) качества работы системы, но без полного останова
4. Временное: повторение действий, начиная с некоторого запомненного прошлого состояния процесса – контрольной точки (checkpoint) или, иначе, точки восстановления (recovery point)
   - Возврат к точке восстановления называется откатом (rollback)

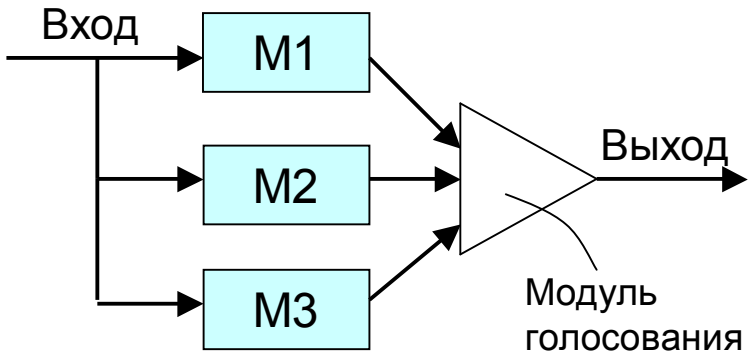
(Вопросы 5, 6)

Структурное резервирование



Дублирование модулей
(Dual Modular Redundancy, DMR)
Средство контроля: обнаруживается отказ одного из модулей

(Вопрос 7)



Троирование модулей
(Triple Modular Redundancy, TMR)
Средство *маскирования* неисправности: отказ одного из модулей не виден извне

- Модуль голосования – узкое место в смысле надежности

- Модули – отдельные устройства или целые компьютеры
- Главный бортовой вычислитель космического корабля Аполлон троирован, у Шаттла – четверирован (Вопрос 8)

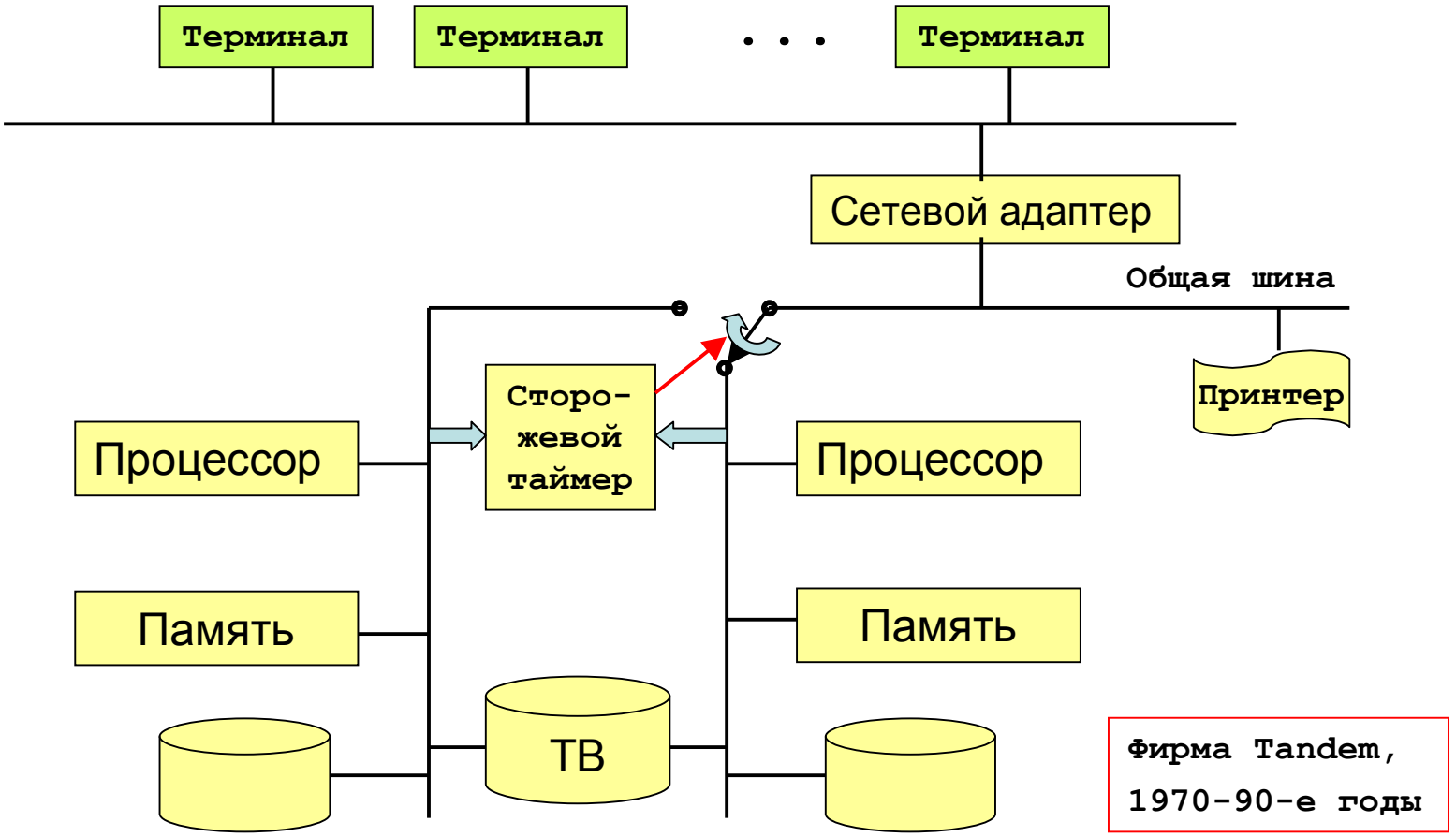
Временное резервирование

- Дешевый вид резервирования (как и информационное резервир.)
- Эффективное средство против сбоев
- Точку восстановления (ТВ) – состояние для возможного отката – можно реализовать двумя способами:

1. В устойчивой памяти (напр., на диске) запоминается текущий контекст процесса и его промежуточные результаты (данные в локальной памяти и на диске)
   – Если единственным результатом процесса является файл, то достаточно запоминать только его текущее состояние – *резервную копию (backup copy)* файла
2. Запоминается последовательность промежуточных действий (шагов) процесса, а откат производится путем выполнения обратных действий – функция Undo
   – Это удобно в диалоговых системах, где естественным образом вычленяются шаги диалога
   – В базах данных и файловых системах этот способ называется *журнализацией* (logging) событий

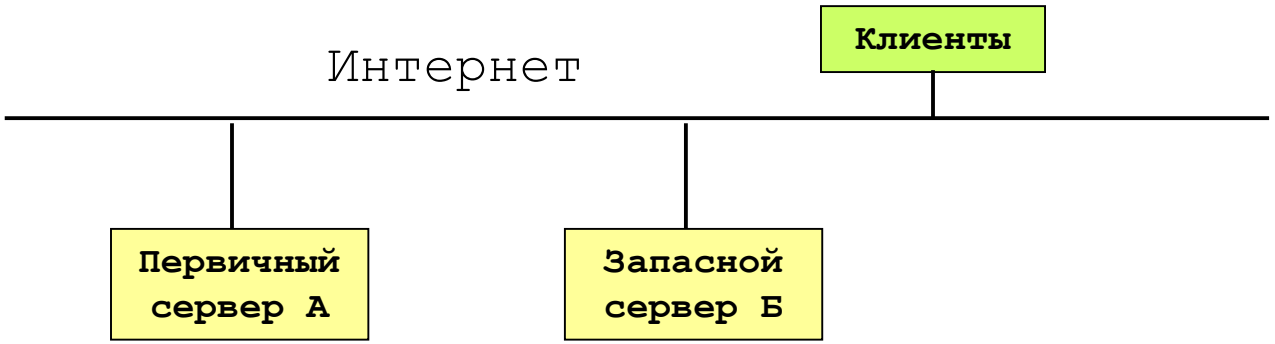
- NB: в приложениях MS Office применяются оба эти способа

Дублирование компьютеров



- ТВ – точка восстановления на двухвходовом диске
- Сторожевой таймер (Watch Dog) переключает шину при прекращении синхросигналов (heartbeats) от активного процессора

Дублирование серверов



- Оба сервера имеют одинаковый IP-адрес
- Интернет-протокол VRRP (Virtual Router Redundancy Protocol) – протокол резервирования с помощью виртуального рутера:
  - нормальный режим: виден только первичный сервер А
  - при крахе А автоматически становится виден запасной Б
    - т.е., машина Б активируется только при *недоступности* машины А
    - отказу подвергаются все текущие сеансы клиентов; они не рестартуют автоматически на машине Б

Отказоустойчивые транзакции

- Транзакция (Тр) – это *составная* операция с данными файлов - последовательность элементарных операций (read, write) для выполнения определенной функции
  - Пример Тр: обновление нескольких взаимосвязанных записей в базе данных (БД), скажем, перевод суммы денег с одного счета на другой в банковской системе и регистрация этой операции в специальном файле – аналоге бухгалтерской книги
- Тр завершается либо операцией commit (фиксировать), либо abort (аварийно завершить)
- Во время выполнения Тр может быть аварийно завершена из-за локального (не системного) отказа
  - напр., из-за отсутствия элемента данных с заданным именем
- Чтобы при этом не возникало противоречивого состояния данных, Тр должна быть атомарной: либо выполняться полностью, либо не выполняться вовсе

Атомарные транзакции

- Журнализация для атомарности – записывать в журнале (log) информацию обо всех модификациях данных, сделанных Tp (вначале только в основной памяти, затем на диске)
- Каждая запись журнала содержит поля:
  - Имя Tp, которая выполнила write; имя записанного элемента данных
  - Старое значение элемента данных до write и новое значение после write
    - *эти записи предшествуют реальному выполнению write*
  - Информация о начале Tp и ее завершении (commit либо abort)
- Журнал используется для восстановления после любого отказа, не приводящего к потере информации в нем. Алгоритм восстановления использует две процедуры:
  - Undo - возвратить значения всех измененных данных к старым значениям
  - Redo - установить значения всех измененных данных в новые значения
- Если Tp выполнила abort, то выполняется undo
- Если произошел системный отказ (при нем обычно теряются данные в основной памяти), то:
  - для всех незавершенных Tp выполняется undo
  - для всех завершенных Tp выполняется redo **(Вопрос 9)**
- Если отказ произошел при записи в журнал, то не страшно:
  - транзакция либо еще и не начиналась (идет только попытка записать намерение ее произвести)
  - либо уже закончилась - идет попытка записать, что транзакция уже выполнена

Точка восстановления транзакций

- Чтобы не просматривать весь журнал и делать лишние redo, периодически ставится контрольная точка
  - все записи, находящиеся в основной памяти, записываются на диск и в журнал выводится запись checkpoint
- После этого применять алгоритм восстановления нужно только для тех транзакций, которые выполнены позже последней контрольной точки
- Таким образом выполняется откат к контрольной точке, поэтому ее называют также точкой восстановления (recovery point)

Спецификация RAID

- RAID – Redundant Array of Independent Disks – избыточный массив независимых дисков
- Цель – повышение производительности и отказоустойчивости дисковых подсистем
  - Производительность повышается благодаря обмену с несколькими дисками одновременно
  - Устойчивость – относительно как искажения блоков данных на дисках, так и отказов дисков / контроллеров в целом
    - Контроль исправности – с помощью контрольных сумм блоков
    - Восстановление – с помощью дублирования блоков и контроля четности
- Различные варианты (комбинации) “распыления” блоков файла по дискам, их дублирования и контроля называются уровнями RAID
- Стандартизовано 6 уровней, из которых наиболее употребительны 0, 1, 3 и 5 уровни.

Уровни RAID (1)

RAID 0 – параллельные диски

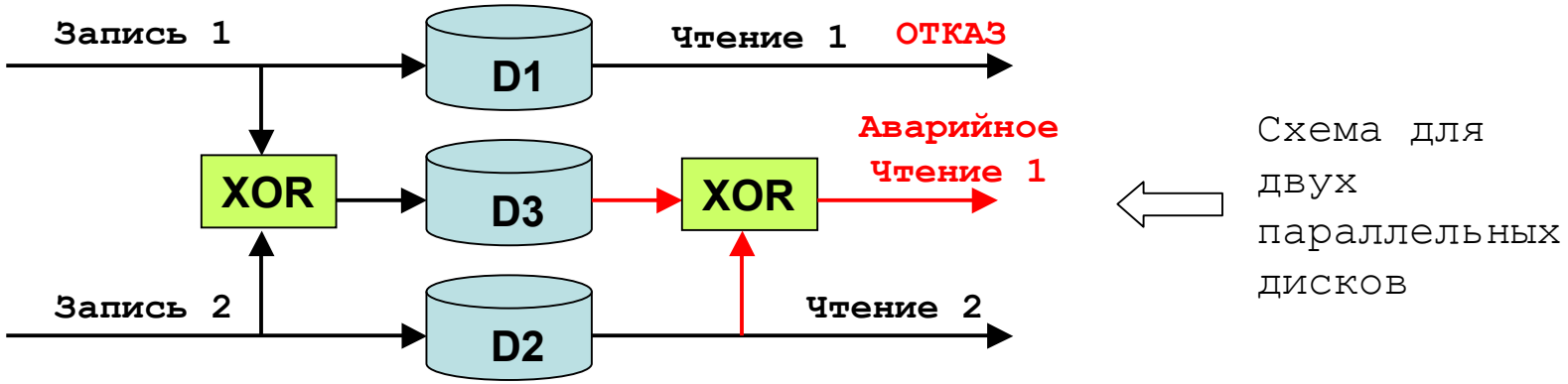
- Блоки файла «распыляются» по нескольким параллельно работающим дискам
- Производительность возрастает, надежность уменьшается по сравнению с одним диском.

RAID 1 – зеркальные диски

- Блоки файла дублируются на двух разных дисках
- Специальный RAID-контроллер считывает обе копии и выдает только те блоки, которые прошли контроль исправности
  - без такого контроллера процесс эмулируется программно
- Объем дискового оборудования удваивается
- Система работоспособна даже при полностью отказавшем одном диске

Уровни RAID (2)

RAID 3 – параллельные диски + контрольный диск четности:



- Диск D3 хранит контрольные суммы (попарно) всех блоков, записанных на D1 и D2 - поразрядные суммы двоичных кодов по модулю 2
- При отказе одного из дисков D1 или D2 содержимое отказавшего блока восстанавливается путем повторного применения XOR
- Параллельных дисков м.б. > 2, тогда сумма XOR вычисляется для всех параллельных блоков сразу
- Система нечувствительна к отказу одного диска
- Объем дискового оборудования увеличивается на 1 диск, а не вдвое, как при простом дублировании в RAID 1

RAID 5 – распыление блоков + контроль четности

- Как RAID 3, но без специального контрольного диска: блоки данных и XOR-кодов записываются на всех дисках вперемешку

Средства отказоустойчивости в W2k (1)

- Атомарные транзакции в NTFS
  - у дисковых файлов не бывает ошибочных промежуточных состояний
- Программная эмуляция RAID 0, 1 или 5 (по выбору) – только для серверов
  - Вариант рабочей станции поддерживает только RAID 0
- Консоль восстановления системы (System Restore)
  - возвращает компьютер к предыдущему состоянию, если установка нового приложения или драйвера привело к отказу системы
  - по умолчанию точки восстановления (ТВ) создаются ежедневно, а также после установки приложений или драйверов
  - пользователь может в любой момент создать собственную ТВ
- Средство автоматического восстановления системы (Automated System Recovery, ASR)
  - поддерживает перезапуск приложений после отказа

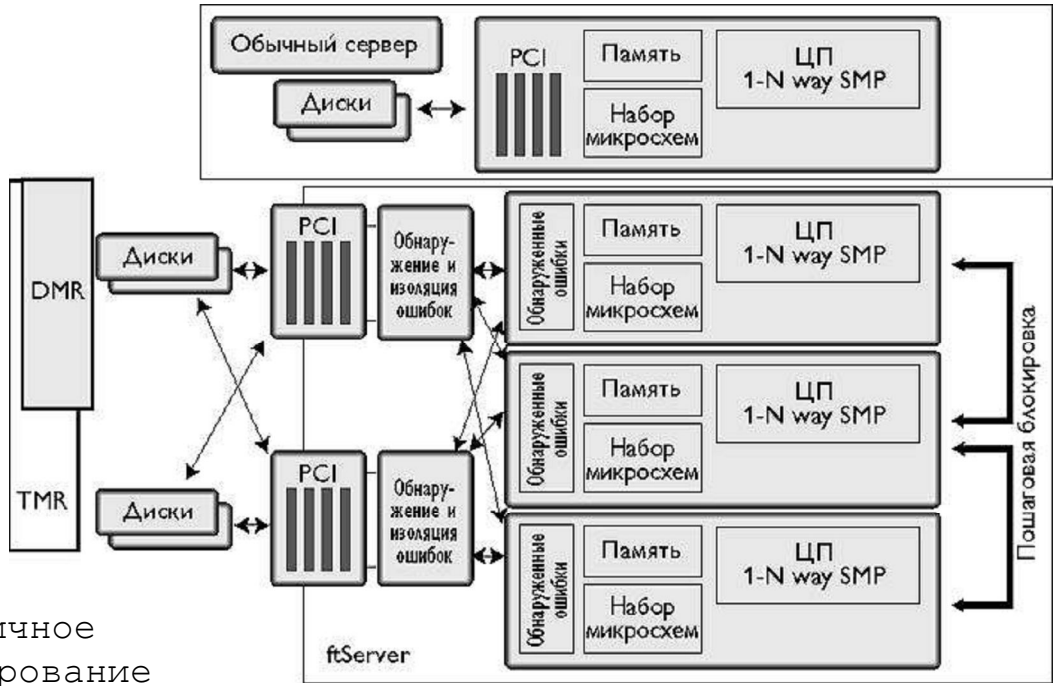
Средства отказоустойчивости в W2k (2)

- Служба теневого копирования томов (Volume Shadow Copy Service, VSS)
  - периодически делает "снимок" текущего содержимого общего ресурса (диска или папки), а затем отслеживает изменения этого ресурса
  - записываются только изменения файлов
  - хранятся до 64 последовательных версий, в зависимости от размера выделенного дискового пространства
  - VSS действует только в томах NTFS, а разделы FAT32 игнорирует
  - существует служба VSS на сервере Windows Server 2003 и клиентская часть VSS

Отказоустойчивое расширение W2k

Отказоустойчивый сервер ftServer фирмы Stratus

- Каждый аппаратный компонент, включая процессоры, дублирован (DMR), а материнские платы троированы (TMR)
- Каждая команда обрабатывается на каждом из резервированных элементов синхронно и параллельно
- В случае ошибки вывод данных с неисправного блока на системную шину автоматически блокируется, а блок переходит в режим самотестирования



SMP-симметричное мультипроцессирование

Отказоустойчивость в Unix/Linux

- Атомарные транзакции в файловой системе Linux XFS
  - XFS может восстанавливаться после отказов менее чем за секунду
  - журнализация XFS позволяет отказаться от длительных проверок целостности файловой системы программой FSCHK, часами работающей на больших системах
- Программный RAID в ОС FreeBSD:

❖ Дисковая подсистема управляется встроенным в ядро механизмом GEOM - модульной дисковой структурой, на основе которой созданы:

  - gstripe (RAID 0)
  - gmirror (RAID 1)
  - graid3 (RAID 3)
  - gconcat (объединение нескольких дисков в единый дисковый раздел

Заключение

- Повышение надежности вычислительных систем достигается как улучшением их качества для уменьшения вероятности отказов (безотказность), так и быстрым автоматическим восстановлением (отказоустойчивость)
- Идеальная отказоустойчивость – восстановление в латентном периоде отказа: оно не дает повреждению проявиться вовне в виде отказа
- Отказоустойчивость достигается введением избыточности – резервов четырех видов и средств контроля и управления резервами
- Самое распространенное (из-за дешевизны) резервирование – временное в сочетании с информационным
- Все больше средств отказоустойчивости встраивается в современные ОС

Вопросы

1. Понятно, как экспериментально измеряется MTBF порядка часов или дней: несколько образцов ставятся на испытания на месяцы или годы, и статистика их отказов усредняется по времени. А как оценить MTBF очень надежного элемента (например, интегральной схемы), если оно порядка сотен лет?
2. Почему надежность (безотказность) вычислительной аппаратуры удалось повысить в последние десятилетия на несколько порядков, а сделать это для программ – нет?
3. Одна из причин ненадежности программных продуктов (ПП) – их б**о**льшая сложность по сравнению с аппаратурой. Какое простое рассуждение подтверждает, что сложность даже не очень больших программ выше, чем сложность компьютеров, на которых они выполняются ?
4. Предположим, ферма содержит 10 тысяч серверов повышенной надежности с MTBF=30 лет. Сколько в среднем серверов будут отказывать за день после начального периода «выжигания дефектов», скажем, на второй год работы?
5. В протоколах TCP/IP реализовано циклическое кодирование 64-байтных блоков передаваемых данных (с восстановлением 2-кратных и обнаружением 3-х кратных ошибок при декодировании) и с переспросом непринятых или сильно искаженных блоков. Каким видам резервирования это соответствует ?
6. Какой из четырех видов резервирования пригоден для противостояния сбоям, но бесполезен при устойчивых отказах ?
7. Что должна делать система, если в схеме дублирования возник сигнал ошибки ?
8. Предложите схему и принцип четырехкратного модульного резервирования.
9. Почему это необходимо делать? Что случится, если не выполнять redo?c