МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

МИРЭА

НАДЕЖНОСТЬ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Методические указания

по выполнению лабораторных работ для магистров, обучающихся

по направлению 38.04.05

МОСКВА 2016

Составитель: А.А. Маркин Редактор: А.А. Миронов

В методических указаниях излагаются рекомендации и методи- ческий материал для выполнения лабораторных работ по дисциплине

«Надежность и тестирование программного обеспечения».

Печатается по решению редакционно-издательского совета уни- верситета.

Рецензенты: Е.Г. Андрианова,

В.Н. Цыпкин

© МИРЭА, 2016

# Лабораторная работа № 1

**ОПРЕДЕЛЕНИЕ ВЕРОЯТНОСТЕЙ СОСТОЯНИЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ НА**

# ОСНОВЕ ЦЕПЕЙ МАРКОВА С НЕПРЕРЫВНЫМ ВРЕМЕНЕМ

1. **Цель работы**

Изучение модели надежности отказоустойчивой системы на основе цепей Маркова.

# Краткие теоретические сведения

* 1. Если случайная величина не является постоянной в некотором процес- се, то такая величина характеризуется случайной функцией, которая в течении процесса может принять тот или другой вид, заранее не известный. Если аргу- ментом случайной функции является время, то такая случайная функция назы- вается вероятностным (случайным) процессом. Функционирование информа- ционной или вычислительной системы представляет собой реализацию вероят- ностных процессов.

В технических и информационных системах потоки событий и случайные процессы взаимосвязаны. Например, процесс смены состояний объекта, может вызываться потоками отказов и потоками восстановлений.

Чтобы охарактеризовать вероятностный процесс, необходимо указать тип процесса и его числовые характеристики. Существует большое число различ- ных типов вероятностных процессов. Наиболее подходящим для описания про- цессов, происходящих во многих технических объектах, является Марковский процесс. Марковский процесс - процесс, у которого в каждый момент времени вероятность перехода в любое другое состояие зависит только от текущего со- стояния объекта и не зависит от того, каким образом объект пришел в это со- стояние.

В исследованиях надежности АСУ и ИС теория марковских процессов получила весьма широкое применение, так как процесс функционирования элементов ИС, как правило, сопровождается простейшими потоками отказов и восстановлений. Экспоненциальное распределение времени работы до отказа и времени восстановления работоспособности - необходимое условие для мар- ковского процесса.

Важнейшая характеристика марковского процесса - вероятность перехода объекта в то или иное состояние за заданный промежуток времени. Информа- ция о вероятностях перехода объекта в различные состояния позволяет опреде- лить вероятности каждого из возможных состояний процесса.

Опишем методику определения вероятностей состояний марковского процесса. Пусть объект исследования может находиться в некоторых состояни- ях, число которых конечно (равно ***n***). Номера состояний 0, 1, 2, . . ., ***n*.** Из ***i*-**го состояния в ***j*-**е объект переходит с постоянной интенсивностью **λij** , обратно – с постоянной интенсивностью **μji**.

На рис.1 представлен граф состояний объекта с резервированием. В этом слу- чае число состояний такой схемы равно 3.

Рис. 1. Схема резервированного объекта ( а ) и граф его состояний ( б ).

Для рассматриваемого примера состояние 0 - оба элемента, входящие в объект, работоспособны; состояние 1 - один из элементов, входящих в объект, в отказо- вом состоянии; состояние 2 - оба элемента, входящие в объект, в отказовом со- стоянии.

Для определения вероятностей нахождения такой системы в каждом со- стоянии можно воспользоваться методом построения дифференциальных урав- нений, предложенным, академиком А.Н. Колмогоровым. Например, система дифференциальных уравнений для определения вероятностей каждого из со- стояний рассматриваемой системы будет иметь вид:

*dP*0 / *dt*= − λ01 *P*0 (*t*)+μ10 *P*1 (*t* )

*dP*1 /*dt* = λ01 *P*0 (*t*)− (λ 12+μ10 )*P*1 (*t*)+μ21 *P* 2(*t*) (1)

*dP*2 / *dt*= − μ21 *P*2 (*t* )+λ12 *P1*(*t* )

Получить систему уравнений (1) можно на основе графа состояний. Для этого необходимо использовать следующее правило: для каждого из возможных со- стояний объекта записывается уравнение, в левой части которого dPi/dt, а спра- ва — взвешенная сумма вероятностей перехода для каждой входящей или ис- ходящей стрелки. Вероятность перехода по стрелке определяется как произве- дение интенсивности перехода и вероятности исходного состояния. Вес веро- ятности перехода равен 1, если стрелка входящая, и -1 - если исходящая.

Решение системы уравнений (1) осуществляется по известным правилам решения систем дифференциальных уравнений. Однако его можно существен- но упростить, если предположить, что рассматриваемый процесс - стационар- ный марковский процесс, для которого производные dPi/dt можно принять рав- ными нулю (вероятности состояний не меняются с течением времени). Система дифференциальных уравнений (1) переходит при этом в систему алгебраиче- ских уравнений:

0= − λ01 *P*0+μ10 *P*1

0= λ01 *P*0+μ10 *P*1

0= λ01 *P*0− (λ12 +μ10 )*P*1 +μ21 *P*2

*P*0+ *P*1+*P* 2= 1 (2)

Четвертое уравнение этой системы (при трех неизвестных) становится необходимым потому, что первые три уравнения сводятся к двум. Решая систе-

му (2) одним из численных методов, получим вероятности состояний исследуе- мого объекта.

* 1. Пусть дана информационная (вычислительная) система, в процессе функционирования которой могут иметь место следующие события (рис.2.):

Исправление ошибки

Локализация ошибки

Обнаружение ошибки

Работоспособное состояние системы

Восстановление системы

Рис.2.

Введем следующие обозначения состояний ИС: 1 — работоспособное состояние ИС;

2 — в системе имеется ошибка (сбой или отказ); 3 — ошибка локализована

1. — ошибка исправлена
2. — ошибка не исправлена, система восстановлена

Модель состояний ИС можно представить в виде графа состояний и пере- ходов (рис. 3). Эта модель позволяет непосредственно определить вероятности пребывания ИС в отдельных состояниях путем решения системы дифференци- альных уравнений Колмогорова.



Рис. 3. Граф состояний и переходов ИС с учетом процесса восстановления.

# Задание на работу

1. Освоить методику составления системы дифференциальных уравнений для определения вероятностей каждого из состояний системы.
2. Запустить макет ИС на основе виртуальной машины lab1.vds
3. Собрать статистику переходов ИС из одного состояния в другое.
4. Найти интенсивности переходов между состояниями ИС.
5. По заданным графу состояний и интенсивностям переходов из одного состо- яния в другое определить вероятности состояний информационной системы (рис. 3).
6. Сравнить результаты, полученные экспериментальным путем с аналитиче- скими результатами п.5.
7. Составить отчет по работе, содержащий все пункты выполнения задания.

# 4. Контрольные вопросы

1. Приведите законы распределения вероятностей, наиболее распространенные в теории надежности.
2. Укажите признаки и свойства простейшего потока отказов.
3. Сформулируйте понятие «вероятностный процесс» и приведите примеры ве- роятностных процессов в информационных системах.
4. Изложите порядок определения вероятностей состояний стационарного Мар- ковского процесса по заданным интенсивностям переходов.

# Лабораторная работа № 2

**ОПРЕДЕЛЕНИЕ ЭФФЕКТИВНОСТИ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ**

# Цель работы

Изучение влияния надежности на показатели эффективности системы массового обслуживания, освоение методики определения этих показателей.

# Краткие теоретические сведения

* 1. Эффективностью системы называется свойство выдавать некоторый полезный результат (эффект) при использовании ее по назначению. Использо- вание понятия «эффективность» в теории надежности вызвано тем, что оно позволяет расширить представления о надежности сложных объектов (систем), для которых возможны не только полные, но и частичные отказы. В последнем случае необходимо определять степень значимости отказов - их влияние на об- щую эффективность системы. В результате осуществляется слияние показате- лей надежности и показателей эффективности в комплексный показатель надежности.

Таким комплексным показателем надежности является коэффициент со- хранения эффективности:

кэфф = Эр/Эном ( 3 )

где Эр - реальное значение эффективности, т.е. эффективность объекта не обла- дающего идеальной надежностью; Эном - номинальное значение эффективности, т.е. эффективность объекта без учета отказов. Технической эффективностью называется технический эффект, полученный при использовании объекта (ко- личество переданной информации, снижение затрат времени т.д.).

* 1. Система массового обслуживания (СМО) - система, состоящая из одного и более каналов обслуживания заявок. На вход такой системы поступает поток заявок с интенсивностью **λз** (средним числом заявок в единицу времени). Про- изводительность работы канала определяется интенсивностью выполнения (об- служивания) заявки **μз**. Граф состояний системы и уравнения Колмогорова, рассмотренные в лабораторной работе № 1 позволяют определять свойства СМО. Для этого составляется граф состояний системы на основе ее конфигура- ции. В настоящей лабораторной работе будет рассматриваться простейшая СМО с одним каналом обслуживания и без очереди. Граф состояний такой си- стемы представлен на рис. 1

Рис. 1. Граф состояний одноканальной СМО с безотказным каналом.

В такой системе состояние S0 соответствует состоянию ожидания очередной заявки на обслуживание, а S1 — состоянию обслуживания заявки, при которой система занята. Заявки, приходящие пока система находится в состоянии S1 от- брасываются (система без очереди) и остаются необслуженными. **λз и μз** обо- значают интенсивности поступления и обслуживания заявок соответственно.

Показателями технической эффективности СМО являются:

1. Относительная пропускная способность системы (средняя доля обслужива- ния заявок) А0, т.е. вероятность того, что заявка пришедшая в СМО будет вы- полнена;
2. Вероятность отказового состояния **Qотк**, т.е. вероятность того, что заявка, пришедшая в СМО, не будет выполнена;
3. Абсолютная пропускная способность системы **Аа = А0λз** - среднее число за- явок, обслуженных в единицу времени;
4. Среднее число занятых каналов **к = Аа/μз;**
5. Среднее число заявок, находящихся в системе **Lср = к + rоч,** где **rоч** – число заявок в очереди;
6. Среднее время пребывания заявки в системе **tср = Lср/μз.**

Различают СМО с очередью и без очереди, одноканальные и многока- нальные, марковские и немарковские, однофазные и многофазные, с приорите- тами и без приоритетов.

**Пример:** Определить показатели номинальной технической эффективности од- ноканальной марковской СМО с интенсивностью заявок λз =3 и интенсивно- стью выполнения заявки μз = 4, граф состояний которой приведен на рисунке 1. Система уравнений, определяющая состояние СМО, будет иметь следующий вид:

.

1. Вероятности состояний СМО: **P(S0) = μз/(λз + μз) = 0,571; P(S1) = λз/(λз + μз)**

# = 0,429.

1. Относительная пропускная способность системы **А0 = P(S0) = 0,571**.
2. Вероятность отказа системы **Qотк = P(S1) = 1 - A0 = 0,429.**
3. Абсолютная пропускная способность системы **Аа = А0λз = 1,713.**
4. Среднее число каналов, занятых обслуживанием **к = Аа/μз = 0,428.**
5. Среднее число заявок, находящихся в системе **Lср = к = 0,428.**
6. Среднее время пребывания заявки в системе **tср = Lср/λз = 0,143** ед. вр. Полученные результаты показывают, что СМО отказывает в выполнении за- данных функций: средняя доля обслуженных заявок составляет всего 0,571, из трех заявок в среднем выполняется лишь 1,71 заявки. Это обусловлено не отка- зами канала, а организацией рабочего процесса (соотношением между **λз** и **μз,** числом обслуживающих каналов и отсутствием очереди). Поэтому эффектив- ность СМО, вычислительных систем, систем передачи данных и в целом ИС за- висит не только от безотказности элементов, но и от организации в них рабочих процессов.

4. В реальных системах массового обслуживания каналы могут выходить из строя на некоторое время вследствие отказа, что приводит к снижению эффек- тивности. Для данного случая выражение (3) примет вид:

**кэфф = Аа р/Аа ном** ,

где Аа ном - номинальная абсолютная пропускная способность СМО; **Аа р** - ре- альная абсолютная пропускная способность системы, рассчитываемая с учетом вероятности нахождения канала в отказовом состоянии.

# 3. Задание на работу

1. Загрузить модель одноканальной СМО с отказами.
2. Задать интенсивность потока заявок **λз** = 3 и интенсивность обработки заявок

**μз** = 4.

1. Пронаблюдать обработку 20 заявок и экспериментально определить парамет- ры СМО: пропускную способность системы, среднее число каналов, занятых обработкой, среднее число заявок в системе.
2. Определить интенсивности отказов и восстановлений системы
3. Аналитически определить параметры СМО при отказах и восстановлениях работоспособности.
4. Сравнить результаты работы СМО в условиях работы с отказами и безотказ- ной работой.
5. Составить отчет по работе, содержащий все пункты выполнения задания.

# 4. Контрольные вопросы

1. Что называется эффективностью объекта? Чем различаются понятия «эффек- тивность номинальная» и «эффективность реальная»?
2. Перечислите основные показатели технической эффективности системы мас- сового обслуживания.
3. Каков порядок составления уравнений Колмогорова по графу состояний си- стемы?
4. Укажите последовательность расчета коэффициента сохранения эффективно- сти одноканальной марковской системы массового обслуживания?

# Лабораторная работа № 3

**ОЦЕНКА НАДЕЖНОСТИ СИСТЕМЫ НА ОСНОВЕ СТРУКТУРНО- ЛОГИЧЕСКОГО АНАЛИЗА ТЕХНИЧЕСКОЙ СИСТЕМЫ**

# Цель работы

Изучение логического метода анализа надежности структуры техниче- ской системы.

# Краткие теоретические сведения

Большинство технических систем реального мира являются сложными объектами и дорогими объектами. В большинстве случаев оценить надежность технической системы на основе концепции «черного ящика» является очень длительной и дорогостоящей процедурой. Поэтому для оценки надежности та- ких систем используют структурно-логический метод декомпозиции структуры ТС на более простые составляющие. Например, при анализе работоспособности технологической линии ее составляющими элементами могут считаться от- дельные установки и станки, транспортные и загрузочные устройства и т.д.. В свою очередь станки и устройства также могут считаться техническими систе- мами и при оценке их надежности они должны быть разделены на элементы - узлы, блоки, которые, в свою очередь - на детали и т.д.

При анализе структуры ТС в первую очередь необходимо оценить влия- ние работоспособности каждого элемента на работоспособность системы в це- лом. С этой точки зрения возможно разделение элементов на следующие типы:

* 1. Элементы, отказ которых практически не влияет на работоспособность системы (например, деформация корпуса, незначительные внешние изменения).
	2. Элементы, работоспособность практически не изменяется и вероят- ность безотказной работы которых близка к единице (корпусные детали, мало используемые компоненты с большим запасом прочности).
	3. Элементы, ремонт или замена которых возможна в ходе эксплуатации

ТС.

* 1. Элементы, отказ которых сам по себе или в сочетании с отказами дру-

гих элементов приводит к отказу системы.

При анализе надежности ТС эффективнее всего рассматривать элементы последней группы.

Для расчетов параметров надежности удобно использовать *структурно - логические схемы надежности* ТС, которые графически отображают взаимо-

связь элементов и их влияние на работоспособность системы в целом. Струк- турно - логическая схема представляет собой совокупность ранее выделенных элементов, соединенных друг с другом последовательно или параллельно. Кри- терием для определения вида соединения элементов (последовательного или параллельного) при построении схемы является влияние их отказа на работо- способность ТС.

*Последовательным* расположением элементов считается такое располо- жение, при котором отказ любого из этих элементов ведет к отказу ТС в целом.

В системе с последовательным соединением для безотказной работы в те- чении некоторого времени *t* необходимо и достаточно, чтобы каждый из ее n элементов работал безотказно в течении этой времени. Считая отказы элемен- тов независимыми, вероятность одновременной безотказной работы n элемен- тов определяется на основе теоремы умножения вероятностей: вероятность совместного появления независимых событий равна произведению вероятно- стей этих событий:

*n n*

*P*  *p*1 *p*2 ... *pn*   *pi*  (1 *qi* )

*i*1 *i*1

Соответственно, вероятность отказа такой ТС

*n n*

*Q*  1 *P*  1  *pi*  1(1 *qi* ).

*i*1 *i*1

Системой с *параллельным соединением элементов* называется система, отказ которой происходит только в случае отказа всех ее элементов. Такие схе- мы надежности характерны для ТС, в которых элементы дублируются или ре- зервируются.

Отказ системы заключается в совместном отказе всех элементов, вероят- ность чего (при допущении независимости отказов) может быть найдена по теореме умножения вероятностей как произведение вероятностей отказа эле- ментов:

*n*

*Q*  *q*1*q*2 ...*qn*  *qi*

*i*1

*n*

 (1  *pi* ).

*i*1

Соответственно, вероятность безотказной работы

*n*

*P*  1  *Q*  1  *qi*

*i*1

*n*

 1 (1  *pi* ).

*i*1

# Задание на работу

1. Составить структурную схему надежности по структурной схеме техниче- ской системы и изучить структурно-логический метод определения надежно- сти ТС.
2. Рассчитать надежность следующей логической схемы. Вероятности p1… pn выбрать в пределах [0,1 … 0,9].



1. Составить отчет о проделанной работе.

# Контрольные вопросы

1. Что называется структурно-логическим методом определения надежности технической системы?
2. Приведите примеры разных типов элементов в зависимости от влияния на работоспособность технической системы?
3. Последовательное соединение элементов технической системы. Преимуще- ства и недостатки.
4. Параллельное соединение технической системы. Преимущества и недостат- ки.

# Лабораторная работа №4

**ИССЛЕДОВАНИЕ НАДЕЖНОСТИ ПЕРЕДАЧИ ДАННЫХ ПО ИНФРАКРАСНОМУ КАНАЛУ**

# Цель работы

Изучение свойств передачи данных по инфракрасному каналу

# Краткие теоретические сведенья

Стек протоколов IrDA имеет три базовых уровня: физический уровень IrDA SIR, IrLAP и IrLMP. Данные могут передаваться IrLMP напрямую от приложений через прикладной программный интерфейс API или через дополнительный транспортный протокол стека (IrTP).

На физическом уровне базовый инфракрасный интерфейс использует характеристики универсального асинхронного приемника/передатчика (UART) в COM-порту. Такой порт имеется практически на всех компьютерах. UART имеет несколько темпов передачи данных в диапазоне от 2400 бит/c до 115 Кбит/c.

Оборудование IrDA-SIR состоит из нескольких компонентов для приема и передачи: кодировщика/декодировщика для кодировки инфракрасного сигнала при передаче и декодировки при приеме, инфракрасного преобразователя в составе драйвера вывода и инфракрасного излучателя для передачи, а также приемника/детектора. Кодировщик/декодировщик имеет интерфейс с UART.

При передаче приемопередатчики IrDA передают сигнал в поток ввода/вывода последовательного порта. Данные поступают передатчику IrDA через последовательный интерфейс c UART. И вместо того, чтобы посылать сигнал

по медной проволоке, как это имеет место при передаче по проводам, он извергает фотоны в воздух в направлении инфракрасного приемника.

Устройства IrDA превращают данные в свет: биты преобразуются в инфракрасный сигнал, при этом вспышка соответствует "0", а отсутствие сигнала - "1". ЦПУ на принимающем конце даже не подозревает о том, что данные передаются в виде света. Оно видит что и всегда, так как кодирование осуществляется передатчиками IrDA на передающем конце и детектором на принимающем конце инфракрасного канала.

С одной стороны, направленность инфракрасного пучка делает невозможным случайную передачу данных близрасположенным устройствам. А с другой стороны, поскольку пучок представляет из себя конус, пользователю незачем точно направлять карманное или портативное устройство на цель, чтобы установить соединение.

ИК-порт выглядит как окошечко из темной (или темно-красной) прозрачной пластмассы, за которым находятся светодиод и фотодиод. Цифровой сигнал после усиления направляется на светодиод одного устройства и посылается в виде инфракрасного излучения на фотодиод другого устройства. Скорость передачи данных составляет от 2,4 кбит/c до 115 кбит/c, а в случае с компьютерными ИК-портами может достичь 4 Мбит/c.

Вся инфракрасная передача происходит в инфракрасном диапазоне от 850 нанометров до 880 нанометров. При помощи ИК-порта можно связать только два устройства на расстоянии не более 1 м, при этом между ними не должно быть никаких преград. Угол, под которым устройства находятся по отношению друг к другу, не должен превышать 60 градусов.

# Порядок выполнения работы.

В лабораторной работе используются персональный компьютер и

различные типы мобильных компьютеров с ИК портом.

Передаются файлы размером – 128Кб, 512 Кб и измеряется время передачи файлов. При этом меняется расстояние между объектами и угол между осями ИК – портов. Результаты сводятся в таблицу.

Проанализировать зависимость стабильности передачи данных от расстояния, от помех. Определить максимальное расстояние, при котором возможна передача данных.

# Задание на работу

Задание 1.

* 1. Установить соединение мобильных компьютеров с локальным компьютером по ИК-порту
	2. Начать передачу данных с расстояния 50 см. Зафиксировать время, затраченное на передачу.
	3. Увеличить расстояние на 20 см, исследовать, как изменилась стабильность передачи.
	4. Повторить п.3

Задание 2.

1. Расположить мобильный компьютер перпендикулярно плоскости ИК- порта
2. Замерить скорость передачи.
3. Измерить устойчивость передачи, удаляя мобильный компьютер:
	* на 10, 20, 30, 40, 50, 60, 70, см параллельно плоскости ИК-порта;
4. Повторить п. 3 изменяя угол между осями ИК – портов на 10, 20, 30, 40 градусов.

На основании результатов измерений построить графики и сделать выводы.

# Контрольные вопросы

1. Базовые уровни IrDА
2. Максимальная скорость передачи данных по ИК-порту
3. Максимальное расстояние между приемопередатчиками ИК- сигнала.
4. Диапазон инфракрасной передачи.

# Лабораторная работа №5

**УСТАНОВКА И НАСТРОЙКА СЕРВЕРНОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ С ОБЕСПЕЧЕНИЕМ НАДЕЖНОСТИ ХРАНЕНИЯ ДАННЫХ**

В представленной лабораторной работе выполнение заданий осуществля- ется на виртуальных машинах. Виртуальная машина ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) virtual machine) — программная или аппаратная среда, исполняющая некоторый [код](http://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) (например, [байт-код](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82-%D0%BA%D0%BE%D0%B4), [шитый код](http://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%82%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), [p-код](http://ru.wikipedia.org/wiki/P-%D0%BA%D0%BE%D0%B4) или [машинный код](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) реального [процессора](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80)).

Для запуска и работы виртуальной машины будет использовано программное обеспечение VMware Player. Подробнее о его работе можно прочитать здесь <http://www.vmware.com/products/player/>.

Информационная система это совокупность технического, программного и организационного обеспечения, предназначенная для проведения операций с информацией. С развитием информационных технологий надежность инфор- мационных систем стала одним из приоритетов.

Главнейшей задачей обеспечения надежности информационной системы является надежность хранения данных. В отличие от остальных компонентов, которые обеспечивают лишь передачу и обработку информации, компоненты хранения данных информационной системы обеспечивают длительное беспе- ребойное хранение информации без изменений. Таким образом, в случае отказа

этого оборудования, произойдет потеря важной информации, восстановить ко- торую в некоторых случаях будет невозможно. Естественно, главным критери- ем сохранности информации является качество аппаратного обеспечения, в частности речь идет о накопителях на жестких магнитных дисках. В настоящее время эта технология получила исключительное развитие и это касается не только объема для хранения данных. Технологический процесс постоянно мо- дернизируется, внедряются новые технологии и решения, в результате достиг- нуты не только рекордные объемы накопителей, но и непревзойденное качество работы. Средняя наработка на отказ (см. MTBF) достигла величин, при которых замена оборудования – ―upgrade‖ – произойдет гораздо быстрее первого сбоя. Однако, это не значит, что оборудование работает безотказно и нет необходи- мости обеспечивать дополнительную надежность для хранения данных.

**RAID** (англ. redundant array of independent disks — избыточный массив независимых жѐстких дисков) — массив из нескольких дисков, управляемых контроллером, взаимосвязанных скоростными каналами и воспринимаемых внешней системой как единое целое. В зависимости от типа используемого массива может обеспечивать различные степени отказоустойчивости и быстро- действия.

Калифорнийский университет в Беркли представил следующие уровни спецификации RAID, которые были приняты как стандарт де-факто:

* RAID 0 представлен как дисковый массив повышенной производительно- сти и меньшей отказоустойчивости.
* RAID 1 определѐн как зеркальный дисковый массив.
* RAID 2 зарезервирован для массивов, которые применяют код Хемминга.
* RAID 3 и 4 используют массив дисков с чередованием и выделенным диском четности.
* RAID 5 используют массив дисков с чередованием и "невыделенным диском четности".
* RAID 6 используют массив дисков с чередованием и двумя независимы- ми "четностями" блоков.
* RAID 10 — RAID 0, построенный из RAID 1 массивов
* RAID 50 — RAID 0, построенный из RAID 5
* RAID 60 - RAID 0, построенный из RAID 6 Рассмотрим некоторые уровни RAID подробнее. RAID 0



RAID 0 (striping — «чередование») — дисковый массив из двух или более жѐстких дисков с отсутствием резервирования. Информация разбивается на блоки данных (Ai) и записывается на оба/несколько дисков одновременно.

RAID 1



Имеет высокую надѐжность — работает до тех пор, пока функционирует хотя бы один диск в массиве. Вероятность выхода из строя сразу двух дисков равна произведению вероятностей отказа каждого диска. На практике при выходе из строя одного из дисков следует срочно принимать меры — вновь восстанавли- вать избыточность. Для этого с любым уровнем RAID (кроме нулевого) реко- мендуют использовать диски горячего резерва. Достоинство такого подхода — поддержание постоянной доступности. Минимальное количество, для построе- ния «зеркального» RAID 1 — два диска.

RAID 5



Основным недостатком уровней RAID от 2-го до 4-го является невозмож- ность производить параллельные операции записи, так как для хранения ин- формации о четности используется отдельный контрольный диск. RAID 5 не имеет этого недостатка. Блоки данных и контрольные суммы циклически запи- сываются на все диски массива, нет асимметричности конфигурации дисков. Под контрольными суммами подразумевается результат операции XOR(исключающее или). Xor обладает особенностью, которая применяется в RAID 5, которая даѐт возможность заменить любой операнд результатом, и применив алгоритм xor, получить в результате недостающий операнд. Напри- мер: a xor b = c (где a, b, c — три диска рейд-массива), в случае если a откажет, мы можем получить его, поставив на его место c и проведя xor между c и b: c xor b = a. Это применимо вне зависимости от количества операндов: a xor b xor c xor d = e. Если отказывает c тогда e встаѐт на его место и проведя xor в ре- зультате получаем c: a xor b xor e xor d = c. Этот метод по сути обеспечивает отказоустойчивость 5 версии. Для хранения результата xor требуется всего 1 диск, размер которого равен размеру любого другого диска в raid.

RAID5 получил широкое распространение, в первую очередь, благодаря своей экономичности. Объем дискового массива RAID5 рассчитывается по формуле (n-1)\*hddsize, где n — число дисков в массиве, а hddsize — размер наименьшего диска. Например, для массива из 4-х дисков по 80 гигабайт общий объем будет (4 — 1) \* 80 = 240 гигабайт. На запись информации на том RAID 5 тратятся до- полнительные ресурсы и падает производительность, так как требуются допол- нительные вычисления и операции записи, зато при чтении (по сравнению с от- дельным винчестером) имеется выигрыш, потому что потоки данных с не- скольких дисков массива могут обрабатываться параллельно.

Производительность RAID 5 заметно ниже, в особенности на операциях типа Random Write (записи в произвольном порядке), при которых производитель- ность падает на 10-25% от производительности RAID 1 (или RAID 10), так как

требует большего количества операций с дисками (каждая операция записи сервера заменяется на контроллере RAID на три - одну операцию чтения и две операции записи). Недостатки RAID 5 проявляются при выходе из строя одного из дисков — весь том переходит в критический режим (degrade), все операции записи и чтения сопровождаются дополнительными манипуляциями, резко па- дает производительность. При этом уровень надежности снижается до надеж- ности RAID-0 с соответствующим количеством дисков (то есть в n раз ниже надежности одиночного диска). Если до полного восстановления массива про- изойдет выход из строя, или возникнет невосстановимая ошибка чтения хотя бы на еще одном диске, то массив разрушается, и данные на нем восстановле- нию обычными методами не подлежат. Следует также принять во внимание, что процесс RAID Reconstruction (восстановления данных RAID за счет избы- точности) после выхода из строя диска вызывает интенсивную нагрузку чтения с дисков на протяжении многих часов непрерывно, что может спровоцировать выход какого-либо из оставшихся дисков из строя в этот наименее защищен- ный период работы RAID, а также выявить ранее необнаруженные сбои чтения в массивах cold data (данных, к которым не обращаются при обычной работе массива, архивные и малоактивные данные), что повышает риск сбоя при вос- становлении данных. Минимальное количество используемых дисков равно трѐм.

RAID 10

RAID 10 — зеркалированный массив, который затем записывается последова- тельно на несколько дисков, как RAID 0. Эта архитектура представляет собой массив типа RAID 0, сегментами которого являются массивы RAID 1. Он объ- единяет в себе высокую отказоустойчивость и производительность. Нынешние контроллеры используют этот режим по умолчанию для RAID 1. То есть, 1 диск основной, 2-й диск — зеркало, причем чтение производится с них пооче- редно, как для RAID 0. Собственно, сейчас можно считать что RAID 10 и RAID

1+0 — это просто разное название одного и того же метода аппаратного зерка- лирования дисков. Но не стоит забывать, что полноценный RAID 1+0 должен содержать как минимум 4 диска.

Комбинированные уровни

Помимо базовых уровней RAID 0 — RAID 5, описанных в стандарте, суще- ствуют комбинированные уровни RAID 1+0, RAID 3+0, RAID 5+0, RAID 1+5, которые различные производители интерпретируют каждый по-своему.

* RAID 1+0 — это сочетание зеркалирования и чередования (см. выше).
* RAID 5+0 — это чередование томов 5-го уровня.
* RAID 1+5 — RAID 5 из зеркалированных пар.

Комбинированные уровни наследуют как преимущества, так и недостатки сво- их «родителей»: появление чередования в уровне RAID 5+0 нисколько не до- бавляет ему надѐжности, но зато положительно отражается на производитель- ности. Уровень RAID 1+5, наверное, очень надѐжный, но не самый быстрый и, к тому же, крайне неэкономичный: полезная ѐмкость тома меньше половины суммарной ѐмкости дисков.

Стоит отметить, что количество жестких дисков в комбинированных массивах также изменится. Например для RAID 5+0 используют 6 или 8 жестких дисков, для RAID 1+0 — 4, 6 или 8.

Программный ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) software) RAID

Для реализации RAID можно применять не только аппаратные средства, но и полностью программные компоненты ([драйверы](http://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B0%D0%B9%D0%B2%D0%B5%D1%80)). Например, в системах на [яд-](http://ru.wikipedia.org/wiki/Linux_%28%D1%8F%D0%B4%D1%80%D0%BE%29) [ре Linux](http://ru.wikipedia.org/wiki/Linux_%28%D1%8F%D0%B4%D1%80%D0%BE%29) существуют специальные модули [ядра](http://ru.wikipedia.org/wiki/Kernel), а управлять RAID- устройствами в [GNU/Linux](http://ru.wikipedia.org/wiki/GNU/Linux) можно с помощью утилиты [mdadm](http://ru.wikipedia.org/wiki/Mdadm). Программный RAID имеет свои достоинства и недостатки. С одной стороны, он ничего не стоит. С другой стороны, программный RAID использует ресурсы [центрально-](http://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80) [го процессора](http://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80), и в моменты пиковой нагрузки на дисковую систему процессор

может значительную часть мощности тратить на обслуживание RAID- устройств.

Ядро GNU/Linux 2.6.28 (последнее из вышедших в 2008 году) поддерживает программные RAID следующих уровней: 0, 1, 4, 5, 6, 10. Реализация позволяет создавать RAID на отдельных разделах дисков.

**LVM** (англ. Logical Volume Manager) — менеджер логических томов операци- онных систем. LVM — это метод распределения пространства жѐсткого диска по логическим томам, размер которых можно легко менять, в отличие от разде- лов.

Физический том (англ. physical volume) — устройство, представляющееся си- стеме как один диск (жесткий диск или его раздел, RAID-массив).

Группа томов (англ. volume group) — набор физических томов в один объект.

Логический том (англ. logical volume) — аналогичен разделу (hda1, sdb3, etc) на не-LVM системах. Так же, как и на них, представляется как блочное устройство и может нести файловую систему. Физический диапазон (англ. physical extent) — диапазоны (обычно несколько мегабайт) физического диска, подобные кластерам.

Логический диапазон (англ. logical extent) — диапазоны, на которые разбивает- ся логический том. Объѐм логических экстентов одинаков по всей группе то- мов.

LVM предоставляет собой дополнительный уровень абстракции между физиче- скими/логическими дисками (привычными разделами, с которыми работает fdisk и аналогичные программы) и файловой системой. Это достигается путѐм разбивки изначальных разделов на небольшие блоки (экстенты, обычно 4—32 МБайт) и объединения их в единый виртуальный том, точнее группу томов (volume group), которая далее разбивается на логические тома (logical volume). Для файловой системы логический том представлен как обычное блочное

устройство, хотя отдельные экстенты тома могут находиться на разных физиче- ских устройствах (и даже сам экстент может быть распределѐн подобно [RAID](http://ru.wikipedia.org/wiki/RAID)).

LVM увеличивает гибкость файловой системы, однако, являясь просто проме- жуточным слоем, не отменяет ограничения и использование других слоѐв и усложняет работу. То есть, по-прежнему нужно создавать и изменять разделы, форматировать их; изменение размера должно поддерживаться также и самой файловой системой.

Для повышения надѐжности можно использовать зеркалирование данных. При этом каждому логическому блоку соответствует не один физический блок, а два или более на разных физических томах. Если один из дисков выйдет из строя, зеркалированный логический том всѐ равно будет доступен.

Создать логический том с одним зеркалом можно с помощью команды lvcreate с параметром -m (см. man lvm). При этом должно быть достаточное количество физических блоков на трѐх физических томах: для самого логического тома, для зеркала и для журнала. Можно обойтись без журнала, но тогда каждый раз при инициализации логического тома (например, при запуске компьютера) бу- дет происходить синхронизация зеркал. Если свободные блоки есть только на двух физических томах, то можно поместить журнал на одном физическом то- ме с зеркалом.

# Порядок выполнения работы.

Для выполнения лабораторной работы необходимо:

* Персональный компьютер
* Установленное приложение VMware Player
* Образ установочного диска операционной системы
1. Запустите приложение VMware Player. Откройте мастер создания виртуальной машины. С помощью него создайте виртуальную машину со

следующими параметрами:

* + Имя ВМ по вашему усмотрению;
	+ Количество оперативной памяти 256 Мб;
	+ Два жестких диска объемом по 4 Гб;
	+ В качестве загрузочного носителя – образ установочного диска операционной системы Ubuntu Server или Redhat Enterprise Linux;
	+ Сетевой адаптер в режиме Bridged.
1. Включите виртуальную машину. Дождитесь предложения об установке ОС. Во время установки придерживайтесь следующих параметров:

Для Redhat Enterprise Linux

* + Язык – русский;
	+ Язык клавиатуры – ru;
	+ Пропустить ввод ключа Код установки;
	+ Подтверждать инициализацию дисков;
	+ Создать собственное разбиение;
	+ Пароль суперпользователя – P@ssw0rd;

Для Ubuntu Server

* + Язык – русский;
	+ Установить Ubuntu Server;
	+ Страна – РФ;
	+ Не определять раскладку автоматически;
	+ Имя сервера – на ваше усмотрение;
	+ Часовой пояс – по-умолчанию;
	+ Имя пользователя – udmin, пароль – 123456;
1. В первой части выполнения лабораторной работы, для обеспечения надежности хранения данных будет использоваться технология RAID.

Для того чтобы создать два раздела защищенных RAID – раздел подкачки swap и раздел для корневого каталога, разбейте два диска одинаково согласно следующей конфигурации: раздел в 400 Мб для подкачки и раздел, занимающий оставшееся свободное место, для корневого каталога. При этом раздел для корневого каталога должен быть первым на обоих дисках. Все четыре раздела должны иметь тип «физический том для RAID». Затем, с помощью вкладки настройка программного RAID сконфигурируйте зеркалирование данных.

*Примечание. Для установки операционной системы семейства GNU/Linux необходимо создать, как правило, два раздела: раздел подкачки (swap) и раздел для монтирования корневого каталога (точка монтирования -* /*). В некоторых случаях (например, при использовании LVM) возникает необходимость в создании дополнительных разделов.*

Далее, завершите установку операционной системы.

1. Смотрите контрольные задания, относящиеся к RAID-технологии.
2. Для выполнения второй части лабораторной работы, необходимо начать установку операционной системы заново. Для этого создайте новую виртуальную машину с параметрами из пункта 1. Процесс установки начнется заново. Следуя пункту 2, вы попадете в интерфейс настройки таблицы разделов. Теперь пользуясь LVM (Logical Volume Manager) создайте логические тома для операционной системы. Учтите, что загрузочные файлы ядра операционной системы (точка монтирования –

/boot) не могут находиться на логическом томе LVM. Поэтому этот раздел вы должны разместить в первую очередь, вне физического тома LVM. В случае сбоя физического устройства, каталог /boot не страшно потерять, так как он не содержит личных данных и его легко восстановить с установочного образа. На оставшемся месте первого

жесткого диска разместите физический том LVM, а затем логический том для корневого каталога размером 1 Гб (точка монтирования - /) и раздел подкачки (swap) объемом 400 Мб. Второй жесткий диск оставьте пустым. Для создания логического тома, необходимо создать группу томов (используйте имя «vg» для группы томов, имя «lvol01» для логического тома).

1. Теперь конфигурация вашего первого жесткого диска (/dev/sda) должна быть такой:

|  |  |  |
| --- | --- | --- |
| Раздел | Тип/Точкамонтирования | Описание/Размер |
| /dev/sda1 | ext4 для /boot | Раздел для загрузочных компонентов ОС, которые используются при загрузкесистемы. Размер 100 Мб. |
| /dev/sda2 | Physical volume | Физический том для LVM. Используется оставшеесяместо. |

Группа томов vg находится на физическом томе /dev/sda2 и содержит в себе один логический том lvol01 объемом 1 Гб и раздел подкачки объемом 400 Мб. Проделав оставшиеся шаги, завершите установку операционной системы.

1. Конфигурация второго диска (/dev/sdb) должна быть приведена к виду:

|  |  |  |
| --- | --- | --- |
| Раздел | Тип/Точкамонтирования | Описание/Размер |
| /dev/sdb1 | Physical volume | Физический том для LVM.50% от объема. |
| /dev/sdb2 | Physical volume | Физический том для LVM.50% от объема. |

Используйте для этого утилиты для работы с разделами (parted, fdisk) и LVM (смотрите man lvm).

1. Для того чтобы использовать физические тома на втором жестком диске расширьте группу томов vg на новые физические тома. Используйте команду vgextend. Теперь группа томов vg имеет в своем составе три физических тома, таким образом, имеет смысл создать логический том с зеркалированием. Создайте такой логический том объемом 100 Мб с именем data с одним зеркалом в составе группы томов vg. Для этого используйте команду lvcreate.
2. Создайте файловую систему на логическом томе data (man mkfs) и примонтируйте том к вашему домашнему каталогу - /home/udmin (man mount). Поместите какие-либо данные в указанный каталог.
3. Смотрите контрольные задания, относящиеся к LVM.

# Контрольные задания.

1. RAID. Выключите виртуальную машину. В параметрах виртуальной машины отключите первый жесткий диск. Включите виртуальную машину. Как отреагировала операционная система на отсутствие оборудования? Какие действия вам необходимо сделать, чтобы восстановить первоначальную конфигурацию оборудования?
2. LVM. Выключите виртуальную машину. В параметрах виртуальной машины отключите второй жесткий диск. Включите виртуальную машину. Как отреагировала операционная система на отсутствие оборудования? Как отреагировал LVM на отсутсвие оборудования? Какие действия вам необходимо сделать, чтобы восстановить доступ к данным на логическом томе data? А что бы изменилось, если бы «отказал» первый жесткий диск?

# Контрольные вопросы

1. Уровни RAID. Достоинства и недостатки программного и аппаратного RAID.
2. LVM. Принципы функционирования. Достоинства и недостатки.

# ЛИТЕРАТУРА

1. Королев В. Л., Сугак Е. В. «Расчеты структурной надежности систем». Методические указания и задания к курсовому проектированию по курсу

―Надежность и испытания МЭУ‖ для студентов направления 55.07.00. Красноярский государственный технический институт, 1997. 40 с.

1. Курс теории надежности систем: Учеб. пособие для вузов / И. А. Уша- ков. — М.: Дрофа, 2008. — 240 с.: ил
2. Левин В.И. Логическая теория надежности сложных систем. - М.: Энер- гоатомиздат, 1985. - 128 с.
3. Надѐжность аппаратно-программных комплексов: Учеб. пособие для ву- зов / Г. Н. Черкесов. — СПб.: Питер, 2005. — 478 с.: ил
4. Нечипоренко В.И. Структурный анализ систем (эффективность и надѐж- ность). - М.: Сов. радио, 1977. - 214 с.
5. Основы тестирования программного обеспечения: Учеб. пособие для ву- зов / В. П. Котляров, Т. В. Коликова. — М.: ИНТУИТ; БИНОМ. ЛЗ, 2006. — 286 с.: ил.
6. Рябинин И.А., Черкесов Г.Н. Логико-вероятностные методы исследова- ния надежности структурно-сложных систем. - М.: Радио и связь, 1981. - 216 с.