МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Воронежский государственный

технический университет»

составитель С.И. Короткевич

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам по курсу

«Стандартизация и сертификация программного обеспечения»

Направление подготовки (специальность) 09.03.02 Информационные системы и технологии, профиль (специализация) Информационные системы и технологии цифровизации

Воронеж 2019

Лабораторная работа №1

«Использование методов динамической избыточности для повышения надежности функционирования ПО»

1. Цель лабораторной работы

Научиться использовать методы динамической избыточности.

2. Методические рекомендации для выполнения лабораторной работы

Один из подходов к динамической избыточности — мажоритарное резервирование (метод голосования). Данные обрабатываются независимо несколькими идентичными устройствами, и результаты сравниваются. Если большинство устройств выработало одинаковый результат, этот результат и считается правильным. И опять, вследствие особой природы ошибок в программном обеспечении, ошибка, имеющаяся в копии программного модуля, будет также присутствовать во всех других его копиях, поэтому идея голосования здесь, видимо, неприемлема. Предлагаемый иногда подход к решению этой проблемы состоит в том, чтобы иметь несколько неидентичных копий модуля. Это значит, что все копии выполняют одну и ту же функцию, но либо реализуют различные алгоритмы, либо созданы разными разработчиками. Этот подход бесперспективен по следующим причинам. Часто трудно получить существенно разные версии модуля, выполняющие одинаковые функции. Кроме того, возникает необходимость в дополнительном программном обеспечении для организации выполнения этих версий параллельно или последовательно и сравнения результатов. Это дополнительное ПО повышает уровень сложности системы, что, конечно, противоречит основной идее предупреждения ошибок — стремиться в первую очередь минимизировать сложность.

Второй подход к динамической избыточности — выполнять запасные копии только тогда, когда результаты, полученные с помощью основной копии, признаны неправильными. Если это происходит, система

автоматически вызывает запасную копию. Если и ее результаты неправильны, вызывается другая запасная копия и т.д.

Методы отступления называются также методами сокращенного обслуживания. Эти методы приемлемы обычно лишь тогда, когда для системы программного обеспечения существенно важно корректно закончить работу. Например, если ошибка оказывается в системе, управляющей технологическими процессами, и в результате эта система выходит из строя, то может быть загружен и выполнен особый фрагмент программы, призванный подстраховать систему и обеспечить безаварийное завершение всех управляемых системой процессов. Аналогичные средства часто необходимы в операционных системах. Если операционная система обнаруживает, что вот-вот выйдет из строя, она может загрузить аварийный фрагмент, ответственный за оповещение пользователей у терминалов о предстоящем сбое и сохранение всех критических для системы данных.

Методы изоляции ошибок. Основная идея методов данной подгруппы — не дать последствиям ошибки выйти за пределы как можно меньшей части системы ПО, так чтобы если ошибка возникла, то не вся система оказалась неработоспособной; отключаются лишь отдельные функции в системе либо некоторые ее пользователи. Например, во многих операционных системах изолируются ошибки отдельных пользователей, так что сбой влияет лишь на некоторое подмножество пользователей, а система в целом продолжает функционировать. В телефонных переключательных системах для восстановления после ошибки, чтобы не рисковать выходом из строя всей системы, просто разрывают телефонную связь.

Другие методы изоляции ошибок связаны с защитой каждой из программ в системе от ошибок других программ. Ошибка в прикладной программе, выполняемой под управлением операционной системы, должна оказывать влияние только на эту программу. Она не должна сказываться на

операционной системе или других программах, функционирующих в этой системе.

В информационной системе изоляция программ является ключевым фактором, гарантирующим, что ошибки в программе одного пользователя не приведут к ошибкам в программах других пользователей или к полному выводу системы из строя.

Основные правила изоляции ошибок в программах состоят в следующем.

1. Прикладная программа не должна иметь возможности непосредственно ссылаться на другую прикладную программу или данные в другой программе и изменять их.

2. Прикладная программа не должна иметь возможности непосредственно ссылаться на программы или данные операционной системы и изменять их. Связь между двумя программами (или программой и операционной системой) может быть разрешена только при условии использования четко определенных сопряжений и только в случае, когда обе программы дают согласие на эту связь.

3. Прикладные программы и их данные должны быть защищены от операционной системы до такой степени, чтобы ошибки в операционной системе не могли привести к случайному изменению прикладных программ или их данных.

4. Операционная система должна защищать все прикладные программы и данные от случайного их изменения операторами системы или обслуживающим персоналом.

5. Прикладные программы не должны иметь возможности ни остановить систему, ни вынудить ее изменить другую прикладную программу или ее данные.

6. Когда прикладная программа обращается к операционной системе, должна проверяться допустимость всех параметров. Прикладная программа не должна иметь возможности изменить эти параметры между моментами проверки и реального их использования операционной системой.

7. Никакие системные данные, непосредственно доступные прикладным программам, не должны влиять на функционирование операционной системы. Ошибка в прикладной программе, вследствие которой содержимое этой памяти может быть случайно изменено, приводит в конце концов к сбою системы.

8. Прикладные программы не должны иметь возможности в обход операционной системы прямо использовать управляемые ею аппаратные ресурсы. Прикладные программы не должны прямо вызывать компоненты операционной системы, предназначенные для использования только ее подсистемами.

9. Компоненты операционной системы должны быть изолированы друг от друга так, чтобы ошибка в одной из них не привела к изменению других компонентов или их данных.

10. Если операционная система обнаруживает ошибку в себе самой, она должна попытаться ограничить влияние этой ошибки одной прикладной программой и в крайнем случае прекратить выполнение только этой программы.

11. Операционная система должна давать прикладным программам возможность по требованию исправлять обнаруженные в них ошибки, а не безоговорочно прекращать их выполнение.

Реализация многих из этих принципов влияет на архитектуру аппаратного обеспечения. Хотя в формулировках некоторых вышеназванных принципов употребляется термин «операционная система», последний применим к любой программе (будь то операционная система, монитор

телеобработки или подсистема управления файлами), которая занята обслуживанием других программ.

Методы введения алгоритмической избыточности объединяют методы построения алгоритмов нечувствительных (или не критичных) к различного рода нарушениям информационного процесса путем использования алгоритмической избыточности.

3. Задание на лабораторную работу

Написать тестовую программу, реализующую один из методов алгоритмической избыточности.

4. Форма отчётности по лабораторной работе

Отчет о лабораторной работе – технический документ, который содержит систематизированные данные о лабораторной работе, описывает теорию, используемую в лабораторной работе, ход лабораторной работы, расчеты и результаты, полученные в ходе лабораторной работы

Отчет составляется по результатам выполнения студентом лабораторной работы.

Студент несет ответственность за достоверность данных, представленных в отчете по лабораторной работе

Структурными элементами отчёта по лабораторной работе являются:

- Титульный лист;
- Цель работы;
- Теоретические сведения;
- Основная часть (ход работы);
- Вывод.

Лабораторная работа №2

«Методы повышения надежности функционирования баз данных»

5. Цель лабораторной работы

Научиться использовать методы повышения надежности функционирования баз данных.

6. Методические рекомендации для выполнения лабораторной работы

Несколько специфичны вопросы обеспечения целостности базы данных в ИС. К надежности базы данных (БД) предъявляются особо жесткие требования, поскольку информация, хранимая в них, используется обычно многократно.

Под целостностью базы данных понимается такое ее состояние, когда имеет место полное и точное сохранение всех введенных в БД данных и отношений между ними, иными словами, если не произошло случайной или несанкционированной модификации, разрушения или искажения этих данных или их структуры.

Для сведения к минимуму потерь от случайных искажений данных необходимо иметь возможность своевременно обнаруживать и устранять возникающие ошибки на этапах хранения, обновления и реорганизации базы данных. Это требует большого набора вспомогательных программ обслуживания баз данных, возможно, даже автономных по отношению к системе управления базой данных.

В частности, к ним относятся программы:

- ведения системного журнала, подробно фиксирующего каждую операцию (транзакцию) над базой данных;
- эффективного контроля достоверности;
- репликации для получения копии базы данных (или ее частей) с целью последующего их восстановления при искажении;

- восстановления для возврата базы данных в первоначальное состояние при обнаружении искажения данных (используют копии базы данных и массивы изменений, формируемые в журнале).

Для надежной работы базы данных ИС осуществляются:

- непрерывное администрирование базы данных ИС;
- регистрация каждого имевшего место доступа к базе данных и выполненных изменений в журнале БД. Системный журнал изменений содержит хронологическую последовательность записей всей информации об изменениях, вносимых в базу данных. В частности, в этот журнал заносятся:
- текст запроса на изменение БД («журнал заявок»), содержащий описание транзакции, терминала и пользователя, время, текст исходного сообщения, тип и адрес изменения данных;
- копии файлов БД до внесения в нее изменений («до-журнал»);
- копии файлов БД после внесения в нее изменений («после-журнал»).
- использование средств СУБД для санкционированного доступа и защиты данных (формирование подсхем базы данных как подмножества структуры базы данных);
- создание страховых (резервных) копий базы данных, «зеркалирование» дисков;
- ведение четко регламентированной системы документооборота и форм документов, разрешенных к использованию;
- криптографирование базы данных;
- формирование групп пользователей и задание для них профилей работы и привилегий доступа к ресурсам БД.

Для обеспечения целостности баз данных могут устанавливаться специальные режимы использования файлов базы данных:

- монопольный — запрещающий обращения к БД от всех программ, кроме одной, вносящей изменения и считывающей информацию из полей базы данных;
- защищенный — вносить изменения в БД вправе лишь одна программа, а остальные программы могут только считывать информацию;
- разделенный — все программы могут и изменять и читать базу данных, но, если одна из них начала работать с БД, остальные ждут окончания этой работы.

Резервирование и восстановление баз данных при аварийных завершениях программы (отказ системы, повреждение носителя) выполняется также по нескольким стратегиям. В частности, резервирование файлов базы данных может выполняться:

- в одном поколении (создание точных копий — дублей файлов БД);
- в разных поколениях (хранятся дубли нескольких временных поколений файлов: «дед», «отец», «сын» и т. д., а также ведется системный журнал изменений);
- смешанное резервирование, использующее совместно две первые стратегии.

Наилучшие результаты обеспечивает смешанное резервирование с системным журналом и контрольными точками отката (рестарта).

Контрольные точки (точки рестарта, точки отката) — место повторного запуска программы при аварийном ее завершении. В контрольных точках обычно выполняются: внесение изменений в БД (в том числе всех изменений, ожидающих своей очереди — неоперативные файлы), разблокирование всех файлов, на обращение к которым был наложен запрет, запись информации о контрольной точке в системный журнал.

Использование массивов RAID (Redundant Array of Inexpensive Disks — избыточный массив недорогих дисков) существенно уменьшает риск простоя

системы из-за отказов накопителей на магнитных дисках, которые являются одним из наименее надежных компонентов современных компьютеров.

В качестве наиболее эффективных мер комплексного обеспечения надежности ИС можно назвать кластеризацию компьютеров и использование отказоустойчивых компьютеров.

7. Задание на лабораторную работу

Реализовать тестовую БД, учитывая один или несколько методов по повышению надежности функционирования баз данных.

8. Форма отчётности по лабораторной работе

Отчет о лабораторной работе – технический документ, который содержит систематизированные данные о лабораторной работе, описывает теорию, используемую в лабораторной работе, ход лабораторной работы, расчеты и результаты, полученные в ходе лабораторной работы

Отчет составляется по результатам выполнения студентом лабораторной работы.

Студент несет ответственность за достоверность данных, представленных в отчете по лабораторной работе

Структурными элементами отчёта по лабораторной работе являются:

- Титульный лист;
- Цель работы;
- Теоретические сведения;
- Основная часть (ход работы);
- Вывод.

Лабораторная работа №3

«Разработка тестов для программного обеспечения»

9. Цель лабораторной работы

Научиться разрабатывать тесты для программного обеспечения.

10.Методические рекомендации для выполнения лабораторной работы

Тестирование программного обеспечения (Software Testing) — проверка соответствия реальных и ожидаемых результатов поведения программы, проводимая на конечном наборе тестов, выбранном определённым образом.

Цель тестирования — проверка соответствия ПО предъявляемым требованиям, обеспечение уверенности в качестве ПО, поиск очевидных ошибок в программном обеспечении, которые должны быть выявлены до того, как их обнаружат пользователи программы.

Этапы тестирования:

- Анализ продукта
- Работа с требованиями
- Разработка стратегии тестирования и планирование процедур контроля качества
- Создание тестовой документации
- Тестирование прототипа
- Основное тестирование
- Стабилизация
- Эксплуатация

Требования — это спецификация (описание) того, что должно быть реализовано.

Требования описывают то, что необходимо реализовать, без детализации технической стороны решения.

Атрибуты требований:

Корректность — точное описание разрабатываемого функционала.

Проверяемость — формулировка требований таким образом, чтобы можно было выставить однозначный вердикт, выполнено все в соответствии с требованиями или нет.

Полнота — в требовании должна содержаться вся необходимая для реализации функциональности информация.

Недвусмысленность — требование должно содержать однозначные формулировки.

Непротиворечивость — требование не должно содержать внутренних противоречий и противоречий другим требованиям и документам.

Приоритетность — у каждого требования должен быть приоритет(количественная оценка степени значимости требования). Этот атрибут позволит грамотно управлять ресурсами на проекте.

Атомарность — требование нельзя разбить на отдельные части без потери деталей.

Модифицируемость — в каждое требование можно внести изменение.

Прослеживаемость — каждое требование должно иметь уникальный идентификатор, по которому на него можно сослаться.

11.Задание на лабораторную работу

Реализовать программу, провести полное тестирование, следуя всем этапам и соблюдая все требования.

12.Форма отчётности по лабораторной работе

Отчет о лабораторной работе – технический документ, который содержит систематизированные данные о лабораторной работе, описывает теорию, используемую в лабораторной работе, ход лабораторной работы, расчеты и результаты, полученные в ходе лабораторной работы

Отчет составляется по результатам выполнения студентом лабораторной работы.

Студент несет ответственность за достоверность данных, представленных в отчете по лабораторной работе

Структурными элементами отчёта по лабораторной работе являются:

- Титульный лист;
- Цель работы;
- Теоретические сведения;
- Основная часть (ход работы);
- Вывод.

Лабораторная работа №4

«Разработка комплексных тестов для ПО»

13. Цель лабораторной работы

Научиться разрабатывать комплексные тесты для программного обеспечения.

14.Методические рекомендации для выполнения лабораторной работы

Комплексное тестирование — процесс поиска несоответствия системы ее исходным целям. В процессе тестирования участвует система, описание целей продукта и вся документация, поставляемая вместе с системой.

Проектирование комплексного теста

Следующие 15 пунктов дают некоторое представление о том, какие виды тестов могут понадобиться.

1. Тестирование стрессов. Как правило, системы функционируют нормально при слабой или умеренной нагрузке, но выходят из строя при большой нагрузке и в стрессовых ситуациях реальной среды. Тестирование стрессов представляет собой попытки подвергнуть систему крайнему «давлению», например попытку одновременно подключить к системе большое количество терминалов, насытить банковскую систему мощным потоком входных сообщений.

2. Тестирование объема представляет собой попытку предъявить системе большие объемы данных в течение более длительного времени, чем п. 1. На вход компилятора следует подать огромную программу (например, программу обработки текстов). Очередь заданий операционной системы следует заполнить до предела. Цель — показать, что система не может обрабатывать данные в количествах, указанных в их спецификациях.

3. Тестирование конфигурации. Система должна быть проверена со всяким аппаратным устройством, которое она обслуживает, или со всякой программой, с которой она должна взаимодействовать. Если сама программная система допускает несколько конфигураций, должна быть тестирована каждая из них.

4. Тестирование совместимости. Как правило, разрабатываемые системы не являются совершенно новыми; они представляют собой улучшение прежних версий или замену устаревших. Тогда на систему накладывается дополнительное требование совместимости, в соответствии с которым взаимодействие пользователя с прежней версией должно полностью сохраниться и в новой системе.

5. Тестирование защиты. К большинству систем предъявляются определенные требования по обеспечению защиты от несанкционированного доступа. Цель тестирования защиты — нарушить секретность в системе. Один из методов — нанять профессиональную группу «взломщиков», т. е. людей с опытом

разрушения средств обеспечения защиты в системах. Одним из путей разработки подобных тестов является изучение известных проблем защиты в этих системах и генерация тестов, которые позволяют проверить, как решаются аналогичные проблемы в тестируемой системе.

6. Тестирование требований к памяти. При проектировании многих систем ставятся цели, определяющие объем основной и вторичной памяти, которую системе разрешено использовать при различных условиях. С помощью специальных тестов нужно попытаться показать, что система этих целей не достигает.

7. Тестирование производительности. Определяются такие характеристики, как время отклика и уровень пропускной способности при определенной нагрузке и конфигурации оборудования. Проверка системы в этих случаях сводится к демонстрации того, что данная программа не удовлетворяет поставленным целям.

8. Тестирование настройки. Тестирование процесса настройки системы очень важно, поскольку зачастую покупатель оказывается не в состоянии даже настроить новую систему.

9. Тестирование надежности/готовности заключается в попытке доказать, что система не удовлетворяет исходным требованиям к надежности (среднее время между отказами, количество ошибок, способность к обнаружению, исправлению ошибок и/или устойчивость к ошибкам и т. д.). Например, в систему можно намеренно внести ошибки (как аппаратные, так и программные), чтобы тестировать средства обнаружения, исправления и обеспечения устойчивости.

10. Тестирование средств восстановления. Можно намеренно ввести в операционную систему программные ошибки, чтобы проверить, восстановится ли она после их устранения. Неисправности аппаратуры, ошибки в данных (помехи в линиях связи и неправильные

значения указателей в базе данных) можно намеренно создать или промоделировать для анализа реакции на них системы.

11. Тестирование удобства обслуживания. Все документы, описывающие внутреннюю логику, следует проанализировать глазами обслуживающего персонала, чтобы понять, как быстро и точно можно указать причину ошибки, если известны только некоторые се симптомы.

12. Тестирование публикаций. Все комплексные тесты следует строить только на основе документации для пользователя. Любые примеры, приведенные в документации, следует оформить как тест и подать на вход программы.

13. Тестирование психологических факторов. Эта сторона не так важна, как другие, однако мелкие недостатки могут быть обнаружены и устранены при тестировании системы. Например, может оказаться, что ответы или сообщения системы плохо сформулированы или ввод команды пользователя требует постоянных переключений регистров.

14. Тестирование удобства установки.

15. Тестирование удобства эксплуатации.

Не все из перечисленных 15 пунктов применимы к тестированию всякой системы (например, когда тестируется отдельная прикладная программа), но тем не менее это перечень вопросов, которые разумно иметь в виду.

По своей природе комплексные тесты никогда не сводятся к проверке отдельных функций системы. Они часто пишутся в форме сценариев, представляющих ряд последовательных действий пользователя. Например, один комплексный тест может представлять подключение терминала к системе, выдачу последовательно 10—20 команд и затем отключение от системы. Вследствие их особой сложности тесты системы состоят из нескольких компонентов: сценария, входных данных и ожидаемых выходных

данных. В сценарии точно указываются действия, которые должны быть совершены во время выполнения теста.

15.Задание на лабораторную работу

Реализовать программу, провести полное тестирование, следуя всем этапам и соблюдая все требования.

16.Форма отчётности по лабораторной работе

Отчет о лабораторной работе – технический документ, который содержит систематизированные данные о лабораторной работе, описывает теорию, используемую в лабораторной работе, ход лабораторной работы, расчеты и результаты, полученные в ходе лабораторной работы

Отчет составляется по результатам выполнения студентом лабораторной работы.

Студент несет ответственность за достоверность данных, представленных в отчете по лабораторной работе

Структурными элементами отчёта по лабораторной работе являются:

- Титульный лист;
- Цель работы;
- Теоретические сведения;
- Основная часть (ход работы);
- Вывод.