МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Кузбасский государственный технический университет имени Т. Ф. Горбачева»

Кафедра информационных и автоматизированных производственных систем

Составитель
С. А. Асанов

ИНЖЕНЕРНО-ТЕХНИЧЕСКАЯ ПОДДЕРЖКА СОПРОВОЖДЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Методические материалы

Рекомендовано цикловой методической комиссией специальности
СПО 09.02.07 Информационные системы и программирование
в качестве электронного издания
для использования в образовательном процессе

Кемерово 2018

Рецензенты:

Ванеев О. Н. – кандидат технических наук, доцент кафедры информационных и автоматизированных производственных систем ФГБОУ ВО «Кузбасский государственный технический университет имени Т. Ф. Горбачева»

Сыркин И. С. – кандидат технических наук, доцент кафедры информационных и автоматизированных производственных систем ФГБОУ ВО «Кузбасский государственный технический университет имени Т. Ф. Горбачева»

Асанов Сергей Александрович

Инженерно-техническая поддержка сопровождения информационной системы: методические материалы [Электронный ресурс] для студентов специальности СПО 09.02.07 Информационные системы и программирование очной формы обучения / сост. С. А. Асанов; КузГТУ. – Электрон. издан. – Кемерово, 2018

Приведен теоретический и практический материал, необходимый для успешного изучения дисциплины.

Методические материалы дисциплины «Инженерно-техническая поддержка сопровождения информационной системы» описывают содержание практических, лабораторных и самостоятельной работ.

Предисловие

Целью освоения дисциплины «Инженерно-техническая поддержка сопровождения информационной системы» является формирование целостного представления об основных требованиях, предъявляемым к работе администратора при настройке и эксплуатации информационных систем.

Основными задачами изучения дисциплины «Инженерно-техническая поддержка сопровождения информационной системы», являются:

- изучение базовых понятий и основных принципов построения систем резервного копирования;
- изучение методов восстановления данных и работоспособности систем;
- формирование навыков поиска и устранения ошибок в работе информационной системы.

Содержание дисциплины в соответствии с учебным планом

В соответствии с учебным планом изучение дисциплины «Инженерно-техническая поддержка сопровождения информационной системы» предусматривает проведение лекционных, практических, лабораторных занятий и самостоятельной работы обучающимися очной формы обучения.

Общая трудоемкость дисциплины составляет 126 часов.

Промежуточный контроль – дифференцированный зачет (8 семестр).

Содержание практических занятий

При подготовке к практическим и лабораторным занятиям обучающиеся самостоятельно изучают основную и дополнительную литературу, готовят конспекты по темам, предложенным преподавателем.

На практических и лабораторных занятиях преподаватель осуществляет контроль подготовки качества знаний обучающегося, используя: опрос, обсуждение вопросов по темам изучаемой дисциплины, письменный опрос при текущем контроле и предоставление отчетов по практическим и лабораторным занятиям.

Практическое занятие № 1. Разработка плана резервного копирования

Целью работы является изучение порядка составления плана резервного копирования. Результатом практической работы является отчет, в котором должны быть отражены ход работ и подготовленный план резервного копирования.

Для выполнения практической работы № 1 студент должен изучить приведенный ниже теоретический материал. Отчет сдается в распечатанном и электронном (файл Word) видах.

План резервного копирования

План резервного копирования это комплекс мер и последовательность действий для создания актуальной копии защищаемых данных на резервном носителе. План может состоять из одного или нескольких заданий, включающих в себя следующие параметры:

- Объекты копирования: сервера, виртуальные машины, диски, тома, папки, файлы, приложения, базы данных.
- Способ копирования: полное, инкрементальное, дифференциальное.
- Расписание: дата, время, период или событие для запуска копирования данных.
- Хранилище: место для хранения резервной копии (облако, локальные или сетевые папки и диски, устройства хранения).
- Параметры хранения: жизненный цикл резервной копии в хранилище и действия, выполняемые после окончания срока хранения.
- Дополнительные параметры: уровень компрессии (сжатия), шифрование потока, проверка целостности резервной копии и др.

Разработка и реализация плана резервного копирования

При создании плана ответьте на следующие вопросы:

- **Насколько важны данные?** Этот критерий поможет решить, как, когда и какую информацию архивировать. Для критичной информации, например, баз данных, следует создавать избыточные архивные наборы, охватывающие несколько периодов архи-

вации. Для менее важной информации, например, для текущих пользовательских файлов, сложный план архивации не нужен, достаточно регулярно сохранять их и уметь легко восстанавливать.

- **К какому типу относится архивируемая информация?** Тип информации поможет определить необходимость архивации данных: как и когда данные должны быть сохранены.
- **Как часто изменяются данные?** Частота изменения влияет на выбор частоты архивирования. Например, ежедневно меняющиеся данные необходимо сохранять каждый день.
- **Нужно ли дополнить архивацию созданием теневых копий?** При этом следует помнить, что теневая копия – это дополнение к архивации, но ни в коем случае не ее замена.
- **Как быстро нужно восстанавливать данные?** Время – важный фактор при создании плана архивации. В критичных к скорости системах нужно проводить восстановление очень быстро.
- **Какое оборудование оптимально для архивации и есть ли оно у вас?** Для своевременной архивации вам понадобится несколько архивирующих устройств и несколько наборов носителей. Аппаратные средства архивации включают ленточные накопители (это наименее дорогой, но и самый медленный тип носителя), оптические диски и съемные дисковые накопители.
- **Кто отвечает за выполнение плана архивации и восстановления данных?** В идеале и за разработку плана, и собственно за архивацию и восстановление должен отвечать один человек.
- **Какое время оптимально для архивации?** Архивация в период наименьшей загрузки системы пройдет быстрее, но не всегда возможно провести ее в удобные часы. Поэтому с особой тщательностью архивируйте ключевые данные.
- **Нужно ли сохранять архивы вне офиса?** Хранение архивов вне офиса – важный фактор на случай стихийного бедствия. Вместе с архивами сохраните и копии ПО для установки или переустановки ОС.

Для построения правильной и эффективной системы резервного копирования необходимо детально изучить и задокументировать все файловые ресурсы, используемые в компании, а затем тщательно спланировать стратегию резервного копирования и реализовать ее на практике

Контрольные вопросы

1. Каковы причины резервирования данных?
2. Какие типы резервного копирования вы знаете? В чем их особенности?
3. Кто планирует, какие данные нужно резервировать?
4. Какие данные необходимо резервировать?
5. Из каких разделов состоит план резервного копирования?

Лабораторная работа № 1.
Создание резервной копии информационной системы

Целью работы является изучение основных методов резервного копирования, приобретение навыков выполнения резервного копирования. Результатом практической работы является отчет, в котором должно быть приведено описание процесса создания резервной копии и снимки экрана, подтверждающие успешное завершение резервного копирования.

Для выполнения лабораторной работы № 1 студент должен изучить приведенный ниже теоретический материал. Отчет сдается в распечатанном и электронном (файл Word) видах.

Резервное копирование информационной системы

Утилиты архивации и восстановления создают копию файлов и папок на указанном пользователем носителе информации. В случае потери или повреждения пользовательских данных их можно восстановить из файла резервной копии. Работу информационной системы без выполнения регулярного резервного копирования нельзя признать удовлетворительной. Частота архивации (резервного копирования) определяется планом резервного копировая (см. Практическое занятие № 1). Администратор может выбирать различные типы архивации в зависимости от его требований:

- Для типа Обычная (Normal) происходит архивация всех выбранных файлов и системных настроек для определенной папки или диска, и каждый файл маркируется как прошедший архивацию (имеющий резервную копию).
- Для типа Копирование (Copy) происходит архивация всех выбранных файлов и системных настроек для определенной папки или диска, но файлы не маркируются как прошедшие архивацию.
- Для типа Добавочная (Incremental) происходит архивация только тех файлов, которые были созданы или изменены вслед за последней обычной или добавочной архивацией, и каждый файл маркируется как прошедший архивацию.
- Для типа Разностная (Differential) происходит архивация только тех файлов, которые были созданы или изменены вслед за

последней обычной или добавочной архивацией, но файлы не маркируются. Для типа

- Ежедневная (Daily) происходит архивация только тех файлов, которые были созданы или изменены в данный день, но файлы не маркируются.

Тип архивации, который применяется, определяет, насколько сложным будет процесс восстановления. Для восстановления после нескольких добавочных или разностных архиваций необходимо выполнить восстановление из последней обычной резервной копии и из всех добавочных или разностных копий, полученных после обычной архивации и вплоть до настоящего момента. Выполняя архивацию данных, администратор указывает имя и место для файла резервной копии. Файлы архивации можно сохранять на жестком диске или на любом другом типе съемного носителя. При выборе места для резервной копии нужно учитывать размер файла архивации, типы имеющихся носителей, а также возможное требование того, что файлы резервных копий нужно хранить отдельно от компьютера на случай катастрофы.

Функция восстановления информационной системы

Восстановление системы позволяет выполнить откат состояния информационной системы к одной из точек восстановления, фиксирующих состояние на момент, когда система стабильно работала. Преимуществом данной функции заключается в том, что она предоставляет возможность быстрого восстановления («отката» состояния системы к состоянию, в котором она находилась в один из предыдущих моментов во времени) без переустановки системы, а также не подвергает риску случайного перезаписывания рабочих файлов пользователей.

Возможно выполнение отката к любому из следующих типов контрольных точек и точек восстановления:

- Начальная контрольная точка (initial system checkpoint) системы создается при первом запуске компьютера с вновь установленной ОС.

- Точки восстановления для автоматических обновлений (Automatic update restore points) создаются, когда инсталлируются

обновления, которые загружаются с помощью штатных средств обновления информационной системы.

- Точки восстановления при восстановлении с резервной копии (Backup recovery restore points) создаются, когда пользователь использует утилиты восстановления данных из резервной копии.
- Пользователь может создавать свои собственные точки восстановления вручную («ручные» контрольные точки – manual checkpoints) в любой момент с помощью утилит управления информационной системы.
- Точки восстановления при инсталляции программ (Program name installation restore points) создаются, при установке программного обеспечения.
- Системные контрольные точки (System checkpoints) – это запланированные точки восстановления, которые создаются компьютером регулярно, даже если пользователь не вносил никаких изменений в систему.
- Точки восстановления для неопознанного устройства (Unsigned device driver restore points) создаются, когда устанавливается драйвер устройства, который не был опознан или сертифицирован.

Средства восстановления системы обычно сохраняют набор контрольных точек восстановления за период от одной до трех недель. Количество контрольных точек восстановления, доступных в любой заданный момент времени, ограничено объемом пространства, которое выделено пользователем для работы системы восстановления.

Контрольные вопросы

1. Укажите особенности различных типов резервного копирования.
2. Что необходимо учитывать при назначении инкрементального или дифференциального резервного копирования?
3. Какие атрибуты файловой системы учитываются системами резервного копирования?
4. Для чего нужна функция восстановления информационной системы?
5. Какие виды контрольных точек существуют?

Лабораторная работа № 2.
Создание резервной копии базы данных

Целью работы является изучение и приобретение навыков настройки служб резервного копирования баз данных. Результатом практической работы является отчет, в котором должно быть приведено описание процесса создания резервной копии базы данных и снимки экрана, подтверждающие успешное завершение процесса резервного копирования.

Для выполнения лабораторной работы № 2 студент должен изучить приведенный ниже теоретический материал. Отчет сдается в распечатанном и электронном (файл Word) видах.

Модели восстановления

При работе с базами данных до настройки резервного копирования следует выбрать модель восстановления. Для оптимального выбора следует оценить требования к восстановлению и критичность потери данных, сопоставив их с накладными расходами на реализацию той или иной модели.

Как известно, база данных MS SQL состоит из двух частей: собственно базы данных и лога транзакций к ней. База данных содержит пользовательские и служебные данные на текущий момент времени, лог транзакций включает в себя историю всех изменений базы данных за определенный период, располагая логом транзакций, мы можем откатить состояние базы на любой произвольный момент времени.

Для использования в производственных средах предлагается две модели восстановления: простая и полная. Существует также модель с неполным протоколированием, но она рекомендуется только как дополнение к полной модели на период крупномасштабных массовых операций, когда нет необходимости восстановления базы на определенный момент времени.

- Простая модель предусматривает резервное копирование только базы данных, соответственно восстановить состояние БД мы можем только на момент создания резервной копии, все изменения в промежуток времени между созданием последней резервной копии и сбоем будут потеряны. В тоже время простая схема имеет не-

большие накладные расходы: вам необходимо хранить только копии базы данных, лог транзакций при этом автоматически усекается и не растет в размерах. Также процесс восстановления наиболее прост и не занимает много времени.

- Полная модель позволяет восстановить базу на любой произвольный момент времени, но требует, кроме резервных копий базы, хранить копии лога транзакций за весь период, для которого может потребоваться восстановление. При активной работе с базой размер лога транзакций, а, следовательно, и размер архивов, могут достигать больших размеров. Процесс восстановления также гораздо более сложен и продолжителен по времени.

Для баз с небольшим объемом добавления информации может быть выгоднее использовать простую модель с большой частотой копий, которая позволит быстро восстановиться и продолжить работу, введя потерянные данные вручную. Полная модель в первую очередь должна использоваться там, где потеря данных недопустима, а их возможное восстановление сопряжено со значительными затратами.

Виды резервных копий базы данных

Полная копия базы данных – как следует из ее названия, представляет собой содержимое базы данных и часть активного лога транзакций за то время, которое формировалась резервная копия (т. е. сведения обо всех текущих и незавершенных транзакциях). Позволяет полностью восстановить базу данных на момент создания резервной копии.

Разностная копия базы данных – полная копия имеет один существенный недостаток, она содержит всю информацию базы данных. Если резервные копии нужно делать довольно часто, то сразу возникает вопрос неэкономного использования дискового пространства, так как большую часть хранилища будут занимать одинаковые данные. Для устранения этого недостатка можно использовать разностные копии базы данных, которые содержат только изменившуюся со времени последнего полного копирования информацию.

Резервная копия журнала транзакций – применяется только при полной модели восстановления и содержит копию журнала

транзакций начиная с момента создания предыдущей копии. Важно помнить следующий момент – копии журнала транзакций никак не связаны с копиями базы данных и не содержат информацию предыдущих копий, поэтому для восстановления базы вам необходимо иметь непрерывную цепочку копий того периода, в течении которого вы хотите иметь возможность откатывать состояние базы. При этом момент последнего успешного копирования должен быть внутри этого периода.

Контрольные вопросы

1. В чем отличие устройства баз данных от других видов хранилищ?
2. Что такое модель восстановления? Какие виды моделей вы знаете?
3. Для чего используется журнал транзакций базы данных? Нужно ли включать его в резервную копию?
4. Какие виды резервных копий баз данных вы знаете?

Лабораторная работа № 3. Восстановление данных

Целью работы является освоение приобретение навыков восстановления данных из ранее созданных резервных копий. Результатом практической работы является отчет, в котором должно быть приведено описание процесса восстановления данных из резервной копии и снимки экрана, подтверждающие успешное завершение процесса восстановления.

Для выполнения лабораторной работы № 3 студент должен изучить приведенный ниже теоретический материал. Отчет сдается в распечатанном и электронном (файл Word) видах.

Общие сведения

Восстановлением называется процедура, которая выполняется для перемещения на жесткие диски компьютера вместо потерянного или испорченного файла или набора файлов их работающей копии из архивных (резервных) данных.

Управление восстановлением – это важная часть процесса аварийно-восстановительных работ. От того, как поддерживается готовность к аварийно-восстановительным работам, будут зависеть время простоя системы и эффективность восстановления потерянного массива информации, полученного между последним архивированием и аварией.

При восстановлении используются следующие основные модели:

- простое восстановление,
- полное восстановление,
- массовое восстановление.

В простой модели восстановления данные могут быть восстановлены только на момент последнего резервного копирования. Эта модель обеспечивает высокую эффективность выполнения массовых операций загрузки данных. Как следует из названия, простая модель копирования и восстановления наиболее легкая и удобная по сравнению с другими моделями. Максимально возможный объем данных, которые могут быть потеряны, определяется периодом времени между созданиями резервных копий.

В модели полного восстановления данные могут быть восстановлены в том виде, в котором она находилась вплоть до аварии. Модель поддерживает восстановление до контрольной точки, помеченной именованной транзакцией. Транзакция – это некоторое законченное, с точки зрения пользователя, действие в информационной системе. В модели полного восстановления массовые операции импорта протоколируются в журнал транзакций и, следовательно, могут быть полностью или частично восстановлены.

В модели массового восстановления операции импорта протоколируются в минимальном объеме. Это обеспечивает высокую производительность массовых операций загрузки, однако делает невозможным восстановление на любой заданный момент времени.

В зависимости от интервала времени, затрачиваемого на воссоздание информации, восстановление подразделяется на следующие виды:

- Восстановление в реальном времени (то есть сразу) или достаточно близко к нему. Данные, созданные не более чем несколько секунд назад, должны быть немедленно доступны пользователям и системам, даже если источник этих данных отключен. Это касается промышленных и медицинских систем, в которых задержка, определяемая восстановлением, допускается в течение долей и единиц секунд. Уровень времени в несколько секунд называется уровнем критического восстановления.
- Если восстановление требуется в течение десяти минут, пока отключен первоначальный источник, то такое восстановление называется экстренным восстановлением.
- Когда на восстановление можно затратить один час, оно называется срочным восстановлением.
- Восстановление, требующее от одного до четырех часов, называется важным восстановлением.
- Все другие восстановления, которые выполняются за интервал времени, больший предыдущих, называются рядовыми.

Стратегия архивации и восстановления

Архивирование не производится ежеминутно, поэтому полностью восстановить все данные на тот момент, когда произошла авария (если только авария не произошла сразу же после завершения

архивирования), с помощью резервных копий невозможно. При восстановлении возможна потеря данных из-за того, что будут устанавливаться устаревшие данные из резервных копий. Необходимо решить, за какое время работы информационной системы потеря информации допустима. Затем, когда установлен этот допустимый уровень, необходимо отработать способы, которые позволят его поддерживать. Также необходимо определить, сколько такая поддержка будет стоить.

Так как при восстановлении будут использоваться устаревшие данные, необходимо определить срок их годности для восстановления. Существует следующий список сроков годности. Для восстановления можно использовать данные, созданные:

- месяц назад или ранее;
- от одной до четырех недель назад;
- от четырех до семи дней назад;
- один–три дня назад;
- шесть–двенадцать часов назад;
- от двух до пяти часов назад;
- от одной до 60 минут назад.

Исходя из сроков годности данных, необходимо определить, какую из технологий архивирования можно использовать. Например, при сохранении на ленте последний уровень срока годности из-за низкой скорости копирования реализован быть не может.

При выработке стратегии восстановления необходимо оценить, насколько требуется немедленное восстановление данных (в реальном времени), и уточнить еще один фактор – возможны ли изъяны в резервных копиях. Это нужно принимать во внимание, если данные непрерывно меняются. Кроме того, файлы могут быть повреждены вирусами. Причем не только в работающей системе, но и в их резервной копии. Архивирование испорченных или зараженных вирусами файлов не обеспечивает достаточную сохранность информации. Гарантировать безопасность копий можно только с помощью предварительного тестирования их антивирусными программами высокой сложности или алгоритмов контроля качества данных.

Безопасность восстановления существенным образом зависит от требуемого времени восстановления. Чем быстрее реакция на запрос по восстановлению, тем выше шанс получить испорченные

данные. Однако это не означает, что критические восстановления всегда рискованные, а восстановленные с их помощью данные – бракованные. Это означает другое. У данных, резервные копии которых получены ближе всего к моменту аварии, больше вероятность оказаться бракованными, чем у тех, что получены за несколько часов или дней до нештатной ситуации. Если авария произошла из-за порчи данных или вирусной инфекции, то вероятнее всего, что недавно резервированные данные также заражены.

Другим фактором, который следует учитывать, является то, что самые «чистые» резервные копии больше всего отличаются от данных, которые нужно восстанавливать, то есть они самые устаревшие.

При восстановлении данных необходимо руководствоваться следующим:

- Проверять, не вызовет ли восстановление данных их потерю, если файлы восстанавливаются на то же место, откуда их копировали. Например, если файл испорчен, но еще не устарел, а при восстановлении его заменяет файл, который не испорчен, но устарел, то потери могут быть большими. Лучше всего, прежде чем заменить испорченный файл, проверить возможность его исправления.
- Учитывать последствия восстановления. При восстановлении файла восстанавливается не только его содержимое, но также все атрибуты и вся информация, известная об этом файле на момент архивирования. Все новое, касающееся файла и его отношений с остальным миром, уже не будет отражено в восстановленном варианте. Примером является восстановление папки, к которой со времени последнего архивирования получило доступ несколько новых групп и пользователей. Восстановление заблокирует этих пользователей.
- Во время восстановления в то же место, откуда делалась резервная копия, доступ пользователей в это место следует заблокировать. Попытка пользователей открыть еще не полностью восстановленные файлы может привести к аварийному завершению восстановления.

Контрольные вопросы

1. Какую задачу решает процедура восстановления информации?
2. Перечислите виды восстановления информации.
3. Какие требования учитываются при разработке стратегии архивирования?
4. Опишите процедуру восстановления информации до момента сбоя в системе.
5. Какие особенности следует учитывать при выборе стратегии восстановления информации?

Лабораторная работа № 4.
Восстановление работоспособности системы

Целью работы является изучение алгоритмов восстановления работоспособности системы с помощью консоли восстановления. Результатом практической работы является отчет, в котором должны быть приведены команды, использованные для восстановления работоспособности системы и снимки экрана, демонстрирующие успешный запуск системы после восстановления.

Для выполнения лабораторной работы № 4 студент должен изучить приведенный ниже теоретический материал. Отчет сдается в распечатанном и электронном (файл Word) видах.

Инструментарий восстановления

Современные операционные системы довольно устойчивы к сбоям и, как правило, стабильность системы тем выше, чем меньше изменений вносится в систему в процессе работы. Однако вносить изменения в конфигурацию операционной системы (установка нового ПО, обновление системы или драйверов, изменение системных параметров и компонент) все же необходимо, в результате чего работоспособность системы может быть нарушена. Обычно, процесс загрузки в операционной системе разделен на несколько частей:

- инициализация;
- работа загрузчика;
- загрузка ядра;
- регистрация.

Соответственно, если проблемы возникают на какой-либо из этих фаз, то операционная система не может выполнить успешную загрузку.

В Windows присутствуют различные средства восстановления, которые вы можете использовать для восстановления работоспособности Windows. Это Безопасный Режим (Safe Mode), Консоль Восстановления (Recovery Console) и Диск Аварийного Восстановления (Automatic System Recovery). Для выбора этих режимов необходимо войти в меню дополнительных вариантов загрузки, для этого во время загрузки системы нажать клавишу F8.

Использование Последней Удачной Конфигурации (Last Known Good Configuration) Если проблема возникла сразу после изменения настроек системы (как правило, после установки нового драйвера), следует воспользоваться загрузкой Windows в режиме Последней Удачной Конфигурации (Last Known Good Configuration). Этот режим восстанавливает информацию реестра и настройки драйвера, которые были использованы, когда система последний раз успешно загружалась. При этом восстанавливается только ветвь реестра HKLM\System\CurrentControlSet и поэтому не решаются проблемы, вызванные повреждением или потерей системных разделов или файлов. Если удалось загрузить Windows в режиме Последней Удачной Конфигурации, то последние изменения, которые были сделаны в системе, скорее всего и были причиной, препятствующей корректному запуску. Удалите или выполните обновление сбойной программы или драйвера, затем загрузитесь в обычном режиме.

Загрузка системы в Безопасном Режиме (Safe Mode) При загрузке в Безопасном Режиме (Safe Mode) Windows загружает только драйвера и службы, которые необходимы для работы. В том случае, если загрузка в Безопасном Режиме была выполнена успешно, то необходимо определить причину возможного сбоя в процессе загрузки. В операционной системе имеется несколько инструментов, которые могут в этом помочь. Выполните вход под учетной записью с правами администратора системы и просмотрите журналы событий (eventvwr.msc). Необходимо провести анализ журнала системы и журнала приложений на наличие предупреждений и сообщений об ошибках, обращайте внимания на источники событий.

Программа просмотра Сведений о Системе (msinfo32.exe) выводит различную информацию об оборудовании, системных компонентах и программном окружении. Если проблемное устройство обнаружено, отключите, перенастройте или попробуйте обновить используемый им драйвер. Для отключения устройства и драйверов используйте Диспетчер Устройств из оснастки Администрирование – Управление Компьютером. Если конфликтов оборудования не обнаружено, просмотрите раздел Программная среда – Автоматически загружаемые программы. Попробуйте запретить программы, загружаемые автоматически, и перезагрузите компьютер. Для настройки запрета воспользуйтесь программой Настройки Системы

(Msconfig.exe), если после запрета загрузка проходит нормально, разрешайте по одной автозагрузку программ. Если и это не помогло, воспользуйтесь режимом Диагностического Запуска, который можно установить в программе Настройки Системы.

Консоль восстановления

Консоль Восстановления это набор средств командной строки, способных помочь восстановить Windows в том случае если компьютер не может выполнить загрузку. Доступ к Консоли можно запустить двумя способами: с загрузочного CD Windows Server 2003 или если Консоль Восстановления была уже установлена на компьютере. Консоль следует запускать в том случае, если ни Режим Последней Удачной Конфигурации, ни запуск в Режиме Восстановления положительного эффекта не дали.

Для того чтобы вывести на экран все доступные команды Консоли Восстановления наберите в командной строке help (или help < command> для получения справки по конкретной команде). Прежде чем начать работу с командами проверьте состояние вашего жесткого диска. Для этого воспользуйтесь командой chkdsk /F /R. Если chkdsk не может исправить проблемы с жестким диском, то ваши файловая система или основная загрузочная запись возможно повреждены или недоступны. Попытайтесь использовать команды Fixmbr и Fixboot для восстановления, в противном случае придется создать разделы заново и переформатировать жесткий диск или обратиться в специализированные компании, которые занимаются восстановлением жестких дисков.

Кроме того, невозможность использования Безопасного Режима для загрузки системы может быть обусловлена повреждением системного реестра Windows или загрузочных файлов. Загрузочные файлы (Ntldr, Ntdetect.com, Boot.ini, Ntbootdd.sys – для контроллеров SCSI, bootfont.bin – для локализованных версий Windows), расположенные в корне системного раздела, могут быть восстановлены из каталога i386 на установочном дистрибутиве Windows Server 2003.

Файлы системного реестра, каждый раз после создания копии Состояния Системы, копируются на системный раздел в каталог % Systemroot%\Repair. Используя Консоль Восстановления можно

восстановить поврежденные файлы реестра из этого каталога в исходный – % Systemroot%\system32\config. Не забудьте предварительно сохранить текущие файлы в другой каталог перед выполнением этой процедуры восстановления. После этого реестр Windows будет содержать информацию, которая была на момент выполнения последнего копирования Состояния Системы. Изменения в системе, начиная с этого момента, будут после восстановления потеряны. Если резервное копирование ни разу не производилось, то в каталоге Repair будет содержаться копия данных сделанная после непосредственно после установки Windows.

Однако не во всех проблемах виновна операционная система и иногда сбой в загрузке возникает еще до начала самой загрузки. Например, если какой-либо другой раздел Вы пометите по ошибке как «активный», не будет содержать файлы загрузки операционной системы, компьютер не запустится. В этом случае при помощи Консоли Восстановления необходимо вернуть метку активного раздела системному разделу. Для этого следует воспользоваться командой diskpart. Предварительно необходимо выбрать ваш системный раздел с файлами запуска (параметры select disk < n> и select partition < m> – где n, m номера, удовлетворяющие соглашениям об именовании ARC), после чего воспользуйтесь параметром active, чтобы пометить его как активный.

Восстановление из резервной копии

Самым последним вариантом является восстановление из резервной копии, которую вы обязательно должны были делать регулярно на работающей системе. Для ее использования необходимо установить новую копию Windows. Если локальный диск является работоспособным, то удаляем существующий системный раздел и создаем новый (при этом размер нового раздела должен быть не менее чем у прежнего). Устанавливаем новую копию Windows Server 2003 на тот же самый раздел, где размещалась Windows ранее. После этого можно приступить к восстановлению из резервной копии.

Контрольные вопросы

1. Для чего предназначена цифровая подпись системных файлов?
2. С помощью какой утилиты осуществляется проверка системных файлов? Какие функции она выполняет?
3. Какие функции выполняет утилита Msconfig?
4. Что такое безопасный режим загрузки Windows? Какие задачи с помощью его решаются?
5. Что такое точки восстановления системы? Как с помощью их решается проблема устранения проблем, вызванных установкой нового приложения?
6. Для чего служит консоль восстановления? Какие способы запуска её вы знаете?

Лабораторная работа № 5. Сбор информации об ошибках

Целью работы является получение навыков по созданию политики сбора информации об ошибках и управлению настройками программного обеспечения сбора информации. Результатом практической работы является отчет, в котором должны быть приведены параметры политики сбора информации, приведены примеры собранных данных.

Для выполнения лабораторной работы № 5 студент должен изучить приведенный ниже теоретический материал. Отчет сдается в распечатанном и электронном (файл Word) видах.

Понятие отчета об ошибках

Отчёт об ошибке (англ. error report или crash report) – это файл, содержащий техническую информацию об исключительной ситуации (исключении), произошедшей в программе на компьютере пользователя.

Отчеты об ошибках создаются, когда в системе возникают неполадки с аппаратным или программным обеспечением. Отчеты об ошибках содержат следующие разделы: сведения о состоянии компьютера при возникновении ошибки, версия используемой операционной системы и аппаратного обеспечения, а также цифровой код продукта, используемый для определения лицензии. Также передается IP-адрес компьютера, поскольку для отправки отчета необходимо подключиться к сетевой службе. Отчеты об ошибках могут содержать данные файлов журналов, например, имена пользователей, IP-адреса, URL-адреса, имена файлов и пути к ним, а также адреса электронной почты. Отчеты об ошибках отправляются с использованием шифрования в базу данных с ограниченным доступом и не используются в каких-либо коммерческих целях.

Настройка параметров отбора событий

Можно настроить параметры сбора данных диагностики для ведения журнала событий. События можно регистрировать как в журнале событий Windows, так и журнале трассировки. Можно настроить параметры регулирования событий, чтобы задавать число

событий, регистрируемых в каждом журнале в соответствии с критичностью события. Для расширения возможностей управления при регулировании событий можно задать регулирование всех событий или любой отдельной категории событий. Доступно несколько категорий событий в зависимости от служб и возможностей.

Категории событий могут определяться по отдельным службам или по группам связанных событий. К категориям выбранных событий относятся:

- для всех;
- категории, определенные в соответствии с используемым продуктом, например, Office SharePoint Server 2007 или Microsoft Office Project Server 2007;
- административные функции, такие как «Администрирование», «Резервное копирование и восстановление», и др.

Для выбранной категории устанавливается минимальный уровень критичности событий, которые следует регистрировать в журнале событий Windows и в журнале трассировки. В каждом журнале будут регистрироваться события этого или более высокого уровня критичности. Записи в этом списке сортируются в порядке от максимального уровня критичности к минимальному.

События журнала Windows могут иметь следующие уровни:

- не используется;
- error (ошибка);
- warning (предупреждение);
- не удалось выполнить аудит;
- успешное выполнение аудита;
- information (информация).

События журнала трассировки могут иметь следующие уровни:

- не используется;
- unexpected (непредвиденный);
- monitorable (контролируемый);
- high (высокая);
- medium (средняя);
- verbose (подробный).

Структура отчета об ошибке

ID (Идентификатор, Номер). Каждой записи в системе учета ошибок присваивается уникальный идентификатор или номер. Как правило, он задается самой системой по определенному шаблону. Это может быть просто числовой номер (1,2,3,…3487), а может быть идентификатор вида ПРОЕКТ-НОМЕР, например, TPO-2367, REG-335 и так далее.

Тип (Трекер). Системы управления задачами, как правило, содержат в себе записи различных типов – задача (Task), улучшение (Feature), баг (Bug), пользовательская история (User Story) и так далее. Для каждого типа может быть свой набор полей и свой жизненный цикл.

Заголовок (Тема, Title). Краткое описание проблемы. Оно отображается в списках, результатах поиска, фильтрах и позволяет быстро понять, в чем суть проблемы. Оно не должно быть слишком коротким и общим, но одновременно и не должно быть слишком длинным. Существует мнемоническое правило для грамотного составления описания «Что? Где? Когда?»: описание должно описывать, что именно сломалось, где сломалось и при каких условиях. Например, «Не работает поиск» – плохое описание ошибки, а «В форме поиска после отправки запроса выдается ошибка «Internal Error» вместо результатов» – уже лучше.

Проект. Как правило, один большой продукт подразделяется на несколько проектов для более удобного управления, и в системе учета задач необходимо указать, к какому именно проекту относится данная ошибка.

Версия. Версия продукта, в которой ошибка была обнаружена.

Компонент. Компонент продукта, где была обнаружена ошибка. Как правило, список доступных компонентов ограничен и создается администратором системы.

Приоритет (Priority). Приоритет показывает, с какой срочностью должен быть исправлен дефект.

Серьезность (Важность, Severity). Серьезность показывает степень влияния дефекта на систему.

Окружение. Описание системы – программного и аппаратного обеспечения, на котором воспроизводится данный дефект.

На кого назначен. Кто будет ответственен за решение данного дефекта. В зависимости от принятых правил и процессов в компании, это может быть конкретный разработчик, руководитель группы разработчиков, или же поле по умолчанию остается пустым, а разработчики сами решают, кто будет править этот дефект.

Описание. Подробное описание проблемы. Иногда бывает одно поле для описания проблемы, и тогда в нем нужно указать шаги для воспроизведения, фактический результат, ожидаемый результат. Иногда вместо этого поля могут быть 3 разных – для шагов, фактического и ожидаемого результатов.

Шаги для воспроизведения. Не всегда этот пункт выделяется как отдельное поле. Если его нет, то нужно шаги для воспроизведения написать в поле «Описание».

Фактический результат. Должен быть обязательно – нужно указать, что мы получили в результате выполнения указанных шагов.

Ожидаемый результат. Необходимо указать, чтобы было понятно, как система должна работать. Если есть возможность, то необходимо давать ссылку на документацию, требования или иные документы, в которых описывается ожидаемое поведение системы. В некоторых случаях этот пункт можно опустить, если ожидаемый результат тривиален (сервер отдает ошибку 500, программа аварийно завершается и т. п.)

Вложения (скриншоты, файлы журнала и пр.). Файлы, которые могут помочь для понимания, воспроизведения, локализации и исправления дефекта.

Контрольные вопросы

1. Какие типы журналов можно просматривать средствами утилиты просмотра событий? Для чего предназначен каждый из них?
2. Какие уровни событий предусмотрены в журнале?
3. Какова структура отчета об ошибках?
4. Что такое отчет об ошибках?
5. Каковы источники информации для создания отчета об ошибках?

Лабораторная работа № 6. Формирование отчетов об ошибках

Целью работы является изучить процесс формирования протокола ошибок, приобрести навыки управления процессом формирования протокола. Результатом практической работы является отчет, в котором должны быть приведены настройки утилит формирования протоколов об ошибках, приведены примеры записей протокола.

Для выполнения лабораторной работы № 6 студент должен изучить приведенный ниже теоретический материал. Отчет сдается в распечатанном и электронном (файл Word) видах.

Протоколы ошибок

Процесс регистрации ошибки начинается с момента, когда ошибка обнаружена модулем операционной системы. Сегмент кода, отвечающий за обнаружение ошибок, передает сведения об ошибке либо в службы ядра errsave и errlast, либо в функцию errlog. В обоих случаях данные заносятся в особый файл /dev/error. Вместе с данными об ошибке записывается время ее обнаружения. Демон errdemon постоянно проверяет наличие новых записей в файле /dev/error, а при поступлении новых данных выполняет стандартную процедуру обработки.

Прежде чем добавить запись в протокол ошибок, демон errdemon сравнивает метку, полученную от ядра или приложения, с содержимым реестра шаблонов ошибок. Если в реестре есть запись, соответствующая метке, демон начинает сбор данных из других областей системы.

Для создания записи в протоколе ошибок демон errdemon считывает шаблон из реестра, имя ресурса блока, обнаружившего ошибку, и сведения об ошибке. Если ошибка свидетельствует об аппаратной неполадке и для нее предусмотрены специальные данные в реестре аппаратного обеспечения (VPD), то демон считывает VPD из ODM. При обращении к протоколу ошибок с помощью SMIT или команды errpt данные протокола форматируются в соответствии с шаблонами в реестре шаблонов и представляются в виде краткого или подробного отчета. Большинство записей в протоколе ошибок связано с программными и аппаратными неполадками, однако в нем могут быть и информационные сообщения.

Команда diag применяется для диагностики аппаратных неполадок на основе содержимого протокола ошибок. Для правильной диагностики новых неполадок система удаляет из протокола записи об аппаратных ошибках старше 90 дней. Записи о программных ошибках удаляются через 30 дней после занесения в протокол.

Передача протокола ошибок в другую систему

Команды errclear, errdead, errlogger, errmsg и errpt входят в состав дополнительного пакета Software Service Aids (bos.sysmgt.serv_aid). Этот пакет применяется для создания отчетов на основе протокола ошибок и удаления записей из протокола ошибок. Вы можете установить пакет Software Service Aids в своей системе, либо передать файл с протоколом ошибок в другую систему, в которой установлен этот пакет.

Существует несколько способов передать файл в другую систему. Например, файл можно скопировать в смонтированную файловую систему из удаленной системы с помощью команды cp. Файл можно передать по сетевому соединению с помощью команды rcp, ftp или tftp. Кроме того, файл можно скопировать на съемный носитель, а затем восстановить его в другой системе.

Для создания отформатированных отчетов на основе протокола ошибок, скопированного из другой системы, служит команда errpt с флагом -i. С флагом -i можно задать каталог, в котором расположен файл протокола ошибок, если этот файл расположен не в каталоге по умолчанию. Для удаления записей из протокола ошибок, скопированного из другой системы, служит команда errclear с флагом -i.

Удаление записей из протокола ошибок

Записи удаляются из протокола ошибок при вызове команды errclear пользователем root, при вызове команды errclear ежедневно выполняемым заданием cron, либо при начале нового цикла записи в файл протокола ошибок после того, как был достигнут максимальный размер файла. После того как размер файла протокола ошибок достигает ограничения, указанного в базе данных конфигу-

рации протокола ошибок, самые старые записи протокола начинают заменяться на новые записи.

По умолчанию команда crontab автоматически удаляет записи об аппаратных ошибках, занесенные более 90 дней назад, и остальные записи, занесенные более 30 дней назад.

Команда errclear позволяет выборочно удалить записи из протокола ошибок. В качестве критерия выбора записей можно указать ИД ошибки, порядковый номер, метку ошибки, имя ресурса, класс ресурса, класс ошибки и тип ошибки. Кроме того, необходимо указать минимальное время создания записей. Команда удалит все записи, соответствующие заданному критерию и созданные позже указанного времени.

Занесение в протокол информации об обслуживании

С помощью команды errlogger системный администратор может добавлять записи в протокол ошибок. При выполнении обслуживания системы рекомендуется заносить в системный протокол ошибок информацию о выполненных действиях, например, об очистке протокола ошибок, замене аппаратного компонента или применении исправления.

Контрольные вопросы

1. Что такое протокол ошибок? Какая информация в нем содержится?
2. Как формируется протокол ошибок?
3. Назовите основные команды управления протоколированием.
4. Как можно очистить протокол ошибок? В каких случаях очистка проводится автоматически?
5. Какие возможности имеются у администратора системы для внесения произвольных данных в протокол ошибок?

Лабораторная работа № 7. Выявление и устранение ошибок программного кода информационных систем

Целью работы является ознакомление с методами обнаружения и устранения ошибок в информационных системах. Результатом практической работы является отчет, в котором должно быть приведено описание указанного преподавателем метода обнаружения ошибок и продемонстрированы результаты его применения на практике.

Для выполнения лабораторной работы № 7 студент должен изучить приведенный ниже теоретический материал. Отчет сдается в распечатанном и электронном (файл Word) видах.

Методы обнаружения ошибок

Принимая во внимание, что в программном обеспечении будут ошибки, то, очевидно, в первую очередь следует принять меры для их обнаружения. Более того, если необходимо принимать дополнительные меры (например, исправлять ошибки или их последствия), то все равно сначала нужно уметь обнаруживать ошибки.

По способу обнаружения ошибок можно выявить два метода:

- пассивные попытки обнаружить симптомы ошибки в процессе «обычной» работы программного обеспечения;
- активные попытки программной системы периодически обследовать свое состояние в поисках признаков ошибок.

Пассивное обнаружение ошибок

Меры по обнаружению ошибок могут быть приняты на нескольких структурных уровнях программной системы. Наиболее продуктивными являются меры, применяемые при переходе от одной компоненты к другой, а также внутри одной компоненты.

Пассивное обнаружение ошибок базируется на следующих принципах:

Взаимное недоверие. Каждая из компонент должна предполагать, что все другие содержат ошибки. Когда она получает какие-нибудь данные от другой компоненты или из источника вне систе-

мы, она должна предполагать, что данные могут быть неправильными, и пытаться найти в них ошибки.

Немедленное обнаружение. Ошибки необходимо обнаружить как можно раньше. Это не только ограничивает наносимый ими ущерб, но и значительно упрощает задачу отладки.

Избыточность. Все средства обнаружения ошибок основаны на некоторой форме избыточности (явной или неявной).

Используя данные принципы можно сформулировать конкретные меры по обнаружению ошибок:

1. Проверяйте атрибуты любого элемента входных данных. Если входные данные должны быть числовыми или буквенными, проверьте это. Если число на входе должно быть положительным, сравните его с нулем. Если известно, какой должна быть длина входных данных, проверьте ее.

2. Применяйте метки и пояснения в таблицах, записях и управляющих блоках и проверяйте с их помощью допустимость входных данных. Добавьте поле записи, явно указывающее на ее назначение.

3. Проверяйте, находится ли входное значение в установленных пределах. Например, если входной элемент – адрес в основной памяти, проверяйте его допустимость. Всегда проверяйте поле адреса или указателя на нуль и считайте, что оно неверно, если равно нулю. Если входные данные – таблица вероятностей, проверьте, находятся ли все значения между нулем и единицей.

4. Проверяйте допустимость всех вариантов значений. Если входное поле – код, обозначающий один из десяти районов, никогда не предполагайте, что если это не код ни одного из районов 1, 2,…, 9, то это обязательно код района 10.

5. Если во входных данных есть какая-либо явная избыточность, воспользуйтесь ею для проверки данных.

6. Там, где во входных данных нет явной избыточности, введите ее. Если ваша система использует крайне важную таблицу, подумайте о включении в нее контрольной суммы. Всякий раз, когда таблица обновляется, следует просуммировать (по некоторому модулю) ее поля и результат поместить в специальное поле контрольной суммы. Подсистема, использующая таблицу, сможет теперь проверить, не была ли таблица случайно испорчена, – для этого только нужно выполнить контрольное суммирование.

7. Сравнивайте, согласуются ли входные данные с какими-либо внутренними данными. Если на входе операционной системы возникает требование освободить некоторый блок памяти, она должна убедиться, что этот блок в данный момент действительно занят.

После обнаружения ошибки неизбежно возникает вопрос, что делать дальше. Наилучшее решение – немедленно завершить выполнение программы или (в случае операционной системы) перевести ЦП в состояние ожидания. С точки зрения предоставления человеку, отлаживающему программу, например системному программисту, самых благоприятных условий для диагностики ошибок немедленное завершение представляется наилучшей стратегией. Всегда, когда это возможно, лучше приостановить выполнение программы, чем регистрировать ошибки (либо обеспечить как дополнительную возможность работу системы в любом из этих режимов).

Активное обнаружение ошибок

Не все ошибки можно выявить пассивными методами, поскольку эти методы обнаруживают ошибку лишь тогда, когда ее симптомы подвергаются соответствующей проверке. Можно выполнять и дополнительные проверки, если спроектировать специальные программные средства для активного поиска признаков ошибок в системе. Такие средства называются средствами активного обнаружения ошибок.

Активные средства обнаружения ошибок обычно объединяются в диагностический монитор: параллельный процесс, который периодически анализирует состояние системы с целью обнаружить ошибку. Большие программные системы, управляющие ресурсами, часто содержат ошибки, приводящие к потере ресурсов на длительное время. Например, управление памятью операционной системы выделяет блоки памяти программам пользователей и другим частям операционной системы. Ошибка в этих самых «других частях» системы может иногда вести к неправильной работе блока управления памятью, занимающегося возвратом сданной ранее выделенной процессу памяти, что вызывает постепенное ухудшение работы системы.

Диагностический монитор можно реализовать как периодически выполняемую задачу (например, она планируется на каждый час) либо как задачу с низким приоритетом, которая планируется для выполнения в то время, когда система переходит в состояние ожидания. Как и прежде, выполняемые монитором конкретные проверки зависят от специфики системы, но некоторые идеи будут понятны из примеров. Монитор может обследовать основную память, чтобы обнаружить блоки памяти, не выделенные ни одной из выполняемых задач и не включенные в системный список свободной памяти. Он может проверять также необычные ситуации: например, процесс не планировался для выполнения в течение некоторого разумного интервала времени. Монитор может осуществлять поиск «затерявшихся» внутри системы сообщений или операций ввода-вывода, которые необычно долгое время остаются незавершенными, участков памяти на диске, которые не помечены как выделенные и не включены в список свободной памяти, а также различного рода странностей в файлах данных.

Иногда желательно, чтобы в чрезвычайных обстоятельствах монитор выполнял диагностические тесты системы. Он может вызывать определенные системные функции, сравнивая их результат с заранее определенным и проверяя, насколько приемлемо время выполнения. Монитор может также периодически выдавать системе «пустые» или «легкие» задания, чтобы убедиться, что система функционирует хотя бы самым минимальным образом.

Контрольные вопросы

1. Какие методы обнаружения ошибок вы знаете? В чем их суть (кратко)?
2. На каких принципах базируется пассивное обнаружение ошибок?
3. Как работает активное обнаружение ошибок?
4. Приведите пример алгоритма активного обнаружения ошибок.
5. Какие требования к проверке входных данных предъявляются с целью минимизации воздействия ошибок?

Лабораторная работа № 8. Выполнение обслуживания информационной системы в соответствии с пользовательской документацией

Целью работы является ознакомление с процессом сопровождения программной системы. Результатом практической работы является отчет, в котором должен быть приведен расчет стоимости сопровождения системы.

Для выполнения лабораторной работы № 8 студент должен изучить приведенный ниже теоретический материал. Отчет сдается в распечатанном и электронном (файл Word) видах.

Общие положения

Сданный в эксплуатацию программный продукт в подавляющем большинстве случаев будет изменяться, поскольку в нем не исключены дефекты, а у его пользователей могут возникнуть новые требования, изменятся условия его эксплуатации и т. д. Весь спектр деятельности, направленный на обеспечение эффективной (с позиции затрат) поддержки программных систем, называется сопровождением программного обеспечения (software maintenance).

Стандарт IEEE 1219 определяет сопровождение как модификацию программного продукта после передачи в эксплуатацию для устранения сбоев, улучшения показателей производительности и/или других характеристик (атрибутов) продукта или его адаптацию для использования в модифицированном окружении.

Сопровождение программы может обходиться значительно дороже стоимости создания базовой версии приложения, поскольку позволяет однажды разработанную программную систему посредством ее адаптации использовать в течение длительного отрезка времени в изменяющихся внешних условиях.

Общей задачей сопровождения является поддержка функционирования программной системы на протяжении всего периода её эксплуатации. Содержательная сторона сопровождения во многом определяется запросами на модификацию, исходящими от пользователей. При этом запросы наиболее интенсивно поступают в службу поддержки в первые шесть недель с момента сдачи системы в

эксплуатацию. Дальнейшие запросы, как правило, связаны с адаптацией ПО или с расширением его функциональности.

При работе с заказчиком в обязанности инженеров службы сопровождения входит: проверка пользовательского сценария, приводящего к сбою; идентификация причин сбоя; устранение причин сбоя или их обход (workaround); документирование всех работ и операций; внесение описания проблемы и ее решения в базу знаний службы сопровождения; передача всей информации разработчикам; информирование пользователя о статусе запроса на сопровождение.

При сопровождении программной системы специалистами, не участвовавшими в её разработке, важным аспектом является трудоемкость её «понимания» (понимание предметной области, архитектуры, алгоритмов работы, исходного кода). «Понимание» программных систем напрямую связано с качеством результата управления конфигурациями: если документация согласована с кодом, то необходимый анализ кода и системы в целом не будет сопряжен с большими трудозатратами. В большинстве случаев правильное форматирование и присвоение переменным информативных имен (хороший стиль программирования) позволяют избежать многих трудностей.

Организация процесса сопровождения подразумевает выполнение следующих действий:

- определение цели и состава процессов сопровождения;
- определение причин и видов изменения программного средства в процессе его сопровождения;
- организация процессов и передача на сопровождение разработанного программного средства;
- заключение договора между заказчиком и исполнителем на сопровождение программного средства;
- разработка концепции методов и процессов сопровождения программного средства;
- разработка спецификации требований на модификации при сопровождении программного средства;
- утверждение заказчиком концепции, договора и технического задания на сопровождение программного средства;
- организация контроля реализации концепции и договора на сопровождение программного средства.

Управление процессом сопровождения

Управленческие вопросы можно разделить на следующие группы:

- согласование с организационными целями;
- кадровое обеспечение;
- организация процесса сопровождения;
- организационные аспекты сопровождения;
- аутсорсинг.

Согласование с организационными целями описывает, как осуществить возврат инвестиций от деятельности по сопровождению. Организационные цели сопровождения направлены на максимальное продление срока эксплуатации программной системы, а деятельность по сопровождению есть обновление и расширение программной системы как отклик на изменяющиеся потребности пользователей.

Проблемы кадрового обеспечения связаны с тем, что инженеры по сопровождению, как правило, считаются в компаниях-разработчиках специалистами «второго класса», поэтому часто возникают проблемы с удержанием квалифицированного персонала в отделах сопровождения.

Организация процесса сопровождения во многом схожа с организацией процесса создания программного обеспечения. В общем случае результатом итерации сопровождения является новая версия программной системы, которая проходит практически все этапы разработки.

Организационные аспекты сопровождения связаны в первую очередь с тем, кто (какая организация) будет осуществлять сопровождение системы. Если сопровождение будет осуществляться силами разработчика, то необходимо определиться со структурными подразделениями, участвующими в этом процессе. Возможно также привлечение сторонних организаций. Преимуществами последнего подхода являются возможность выбора сопровождающей организации из нескольких альтернатив (что позволяет выбрать подходящую стоимость сопровождения), а также появление у разработчика возможности заниматься другими видами работ (не сопровождением). Основным недостатком считается постепенная потеря разработчиками контроля над кодом созданной системы.

Аутсорсинг подразумевает полную передачу непрофильных работ сторонним организациям. Лишь незначительная часть крупных корпораций использует аутсорсинг и только для некритичных компонентов, выполняющих не очень важные бизнес-функции, поскольку они не хотят терять контроль над ассоциированными с этими системами данными или функциональностью. К тому же сама процедура передачи системы на внешнее сопровождение слабо отражена в стандартах, что затрудняет документальное определение предоставляемых аутсорсером сервисов.

Наилучшее решение – немедленно завершить выполнение программы или (в случае операционной системы) перевести ЦП в состояние ожидания. С точки зрения предоставления человеку, отлаживающему программу, например системному программисту, самых благоприятных условий для диагностики ошибок немедленное завершение представляется наилучшей стратегией. Всегда, когда это возможно, лучше приостановить выполнение программы, чем регистрировать ошибки (либо обеспечить как дополнительную возможность работу системы в любом из этих режимов).

Стоимость сопровождения

Стоимость сопровождения определяется усилиями, затраченными на понимание существующего кода системы и на разработку нового в рамках запроса на сопровождение, стоимость которого можно оценить по известной методике. На стоимость сопровождения в целом могут оказать влияние следующие факторы:

- тип приложения;
- новизна программного обеспечения;
- наличие и квалификация персонала по сопровождению;
- длительность использования программной системы;
- характеристики и специфика аппаратной части, телекоммуникационной инфраструктуры;
- качество дизайна (например, модульность или масштабируемость), кода, документации и соответствующих работ по тестированию системы.

На стоимость сопровождения влияет также сложность системы и ее компонентов: чем сложнее система, тем дороже ее сопровождение. Поэтому важно также осуществить прогнозирование количества запросов на изменение системы при ее разработке. При этом

следует учитывать связь системы с внешним окружением – для систем, находящихся в сложной взаимозависимости с внешним окружением, изменение последнего обязательно повлияет на систему. Для адекватной оценки этой взаимозависимости необходимо оценить следующие показатели:

- количество и сложность системных интерфейсов (чем больше системных интерфейсов и чем более сложными они являются, тем выше вероятность изменений в будущем);
- количество изменяемых системных требований;
- бизнес-процессы, в которых используется данная система (по мере развития бизнес-процессов появляются изменения в требованиях).

Если сопровождение осуществляется сторонней организацией или группой специалистов той же фирмы, которые не занимались разработкой данной программной системы, то на первый план выходит оценка трудозатрат, связанных с пониманием программного кода. В этом случае важно наличие согласованной документации и комментариев.

Для косвенной оценки трудозатрат на сопровождение (и стоимости сопровождения) можно применить доля комментариев в общем числе строк сопровождаемого кода. Вычислить долю комментариев можно с помощью специальных программ, в избытке присутствующих на рынке. При оценке стоимости сопровождения важно учесть затраты на распространение модифицированной программной системы (или ее части) среди ее пользователей. Данная статья расходов значительно возрастает для многотиражных систем и систем, нуждающихся в частом обновлении версий.

Контрольные вопросы

1. Что такое сопровождение ПО?
2. Какие виды работ выполняются при сопровождении?
3. Как влияет полнота документации на трудоемкость сопровождения?
4. Какие виды работ выполняются при осуществлении сопровождения?
5. Возможно ли осуществлять сопровождение ПО силами сторонних организаций, не принимавших участия в его создании?
6. Как можно оценить трудозатраты на сопровождение?

СОДЕРЖАНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Цель самостоятельной работы обучающихся – получить новые знания по дисциплине «Инженерно-техническая поддержка сопровождения информационной системы».

Самостоятельная работа необходима для формирования у обучающихся способности самостоятельно решать задачи профессиональной деятельности, формирования умения и навыков планирования времени, формирования стремления развиваться и совершенствоваться.

Виды самостоятельной работы обучающихся указаны в табл. 1.

Таблица 1

Виды самостоятельной работы

№ п/п	Вид СРС
1	Сопровождение ИС: стандарт IEEE-90
2	Практические примеры применения стандартов в сопровождении ИС
3	Настройка информационной системы под конкретного пользователя
4	Формирование отчетной документации по результатам выполнения работ
5	Аппаратно-программные платформы серверов и рабочих станций
6	Порядок установки и сопровождения серверного программного обеспечения
7	Журнал регистрации событий информационной системы
8	Программное обеспечение тестирования и выявления аппаратных ошибок
9	Программные и аппаратные средства резервного копирования

Обучающиеся должны изучить интернет-ресурсы и литературу по вопросам, представленные ниже.

Учебно-методические материалы по дисциплине

Основная литература

1. Перлова, О. Н. Соадминистрирование баз данных и серверов [Электронный ресурс] : учебник для студентов среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование / О. Н. Перлова, О. П. Ляпина. – Москва : Академия, 2018. – 304 с. – Режим доступа: http://www.academia-moscow.ru/catalogue/4831/345911/. – Загл. с экрана.

Дополнительная литература

1. Гончарук, С. В. Администрирование ОС Linux [Электронный ресурс]. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 165 с. – Режим доступа: http://biblioclub.ru/index.php?page=book_red&id=429014. – Загл. с экрана.

Программное обеспечение и интернет-ресурсы

1. Официальный сайт Кузбасского государственного технического университета имени Т. Ф. Горбачева. Режим доступа: www.kuzstu.ru

2. Электронные библиотечные системы:

- Университетская библиотека онлайн. Режим доступа: www.biblioclub.ru;
- Лань. Режим доступа: http://e.lanbook.com
- Электронно-библиотечная система Znanium.com
- Электронная библиотека издательства Юрайт https://biblio-online.ru/catalog/spo

3. Информатика и информационные технологии: конспект лекций. [Электронный ресурс]. – Режим доступа: http://fictionbook.ru

4. Современные тенденции развития компьютерных и информационных технологий: [Электронный ресурс]. – Режим доступа: http://www.do.sibsutis.ru

5. Единая коллекция Цифровых образовательных ресурсов [Электронный ресурс]. – Режим доступа: http://school-collection.edu.ru/, свободный. – Загл. с экрана.

6. Единое окно доступа к информационным ресурсам [Электронный ресурс]. – Режим доступа: http://window.edu.ru/, свободный. – Загл. с экрана.

7. Информационно-коммуникационные технологии в образовании [Электронный ресурс]. – Режим доступа: http://www.ict.edu.ru/, свободный. – Загл. с экрана.

8. Федеральный центр информационно-образовательных ресурсов [Электронный ресурс]. – Режим доступа: http://fcior.edu.ru/, свободный. – Загл. с экрана.

СОДЕРЖАНИЕ