**Вопросы по лекции №12.**

**Низкоуровневые средства обнаружения ошибочного функционирования ПО и низкоуровневые методы защиты при конструировании безопасного ПО.**

Бригада СанУ11: Ковалев К.А., Терентьев Е.А., Щекочихин Д.А., Оганисян Т.С., Шабакаев А.Р.

Выполнил: Оганисян Т.С.

Группа 6411

**1.Какие виды низкоуровневых средств обнаружения ошибочных ситуаций в ПО Вам известны?**

1. Средства обнаруживающие ошибочные значения данных

2.Средства самой программы, контролирующие её функционал, которые могут обнаруживать неправильное функционирование программы во времени.

3. Средства, встроенные в саму программу и имеющие характер «утверждений» (Утверждения проверяют условия наличия событий, которые никогда в нормально работающей программе не должны происходит).

**2. Привести примеры исключительных ситуаций, обнаруживаемых внешними по отношению к ПО средствами**

Примеры исключительных ситуаций:

* арифметические переполнения, такие как деление на ноль, взятие логарифма 0 по любому основанию и другие;
* ошибка при попытке считать данные с внешнего устройства;
* исчерпание доступной памяти, т.е. система не в состоянии выделить необходимый объём оперативной памяти для нормальной работы программы.

**3.Можно ли программе игнорировать возникшую исключительную ситуацию и продолжать работать?**

Нет, простое игнорирования исключительной ситуации и продолжение работы программы – опасная тактика, так как приводит к ошибочным результатам работы программ или возникновению необратимых ошибок в последствии, что может только усугубить ошибочную ситуацию.

**4. Перечислите три основных шага для обнаружения и анализа ошибок во входных данных.**

Существует три основных шага:

1.Необходимо проверять все данные из внешних по отношению к программе источников. Получив данные из файла, от пользователя, из сети или любого другого внешнего интерфейса, необходимо удостоверится, что все значения попадают в допустимый интервал. Если строка должна содержать определенный набор значений, необходимо проконтролировать, что этот набор значений имеет место в данном случае, если же нет - эти данные следует отклонить.

2. Необходимо проверять значения всех входных параметров метода. Проверка значений входных параметров метода практически то же самое, что и проверка данных из внешнего источника, за исключением того, что данные поступают из другого метода, а не из внешнего интерфейса. Этот подход можно рассматривать как реализацию принципа эшелонированной обороны.

3. Необходимо решить, как обрабатывать неправильные входные данные. Что делать, если обнаружен неверный параметр или значение?

**5.Каким методом можно обнаружить ошибки в данных**

В приложениях, где происходит циклическое повторение алгоритмов управления с выбранным периодом, устойчивость к ошибкам данных вследствие воздействия помех возможно реализовать, исходя из того что для циклически работающих программ один неверный цикл – такт может не привести к резкому ухудшения общего результата управления и имеется большая вероятность исправления результатов на следующем цикле-такте исполнения программы при поступлении новой порции уже неискаженных входных данных.

**6.Каким методом можно обнаружить не выполнение программой требуемых временных соотношений**

Встроенный в ПО контроль «Завершение заданной функции за заданное время». В этом случае сигналы о фактическом начале и завершении выполнения какой-то внешней системной функции ПО могут быть заведены в ПО, например, через систему прерывания ЦВМ. В ПО через устройство отсчёта времени ЦВМ на момент начала исполнения заносится контрольная, рассчитанная в ПО же уставка времени. Если эта уставка времени истекла, а сигнал о завершении внешней системной функции не поступил, то ПО вырабатывает «ошибку». Иногда срабатывание этого контроля говорит не о наличии ошибки в ПО, а о наличие ошибки в управляемой от ПО аппаратуре – системной ошибке. В этом случае для точной диагностики необходимо привлечение дополнительной информации.

**7.Перечислить встроенные в ПО низкоуровневые методы обработки обнаруженных ошибок во входных и выходных данных**

1.Выдать нейтральное значение или вернуть ближайшее допустимое значение

2.Продолжение выполнения задачи с выдачей в качестве результата работы на такте, в котором обнаружена ошибка, заведомо безопасного значения.

3.Заменить ошибочные данные следующим корректным блоком данных- метод пропуска такта.

4.Выдать тот же результат, что и на предыдущем такте.

5.Прекратить выполнение.

Возможно, что остановку работы лучше делать по технологии «Мягкого останова» с привлечением аварийной защиты системного уровня.

Выполнил: Терентьев Е.А.

Группа 6411

**1. Какие виды низкоуровневых средств обнаружения ошибочных ситуаций в ПО Вам известны?**

Низкоуровневые средства, обнаруживающие ошибочное функционирование ПО, подразделяются на встроенные в ПО и внешние по отношению к ПО. Встроенные в ПО средства в свою очередь подразделяются на:

1. Средства обнаруживающие ошибочные значения данных, основанные на установке и проверке допустимых границ данных
2. Средства самой программы, контролирующие её функционал, которые могут обнаруживать неправильное функционирование программы во времени
3. Средства, встроенные в саму программу и имеющие характер «утверждений», проверяющих условия наличия событий, которые никогда в нормально работающей программе не должны происходить

**2. Привести примеры исключительных ситуаций, обнаруживаемых внешними по отношению к ПО средствами**

Примерами состояний, которые имеют название исключительные ситуации могут служить арифметические переполнения, ошибка при попытке считать данные с внешнего устройства, исчерпание доступной памяти и т.п.

**3. Можно ли программе игнорировать возникшую исключительную ситуацию и продолжать работать?**

Игнорирование исключительной ситуации и продолжение работы программы может привести к ошибочным результатам работы программы или возникновению необратимых ошибок в последствии, что может только усугубить ошибочную ситуацию. Поэтому игнорировать возникшую исключительную ситуацию нельзя.

**4. Перечислите три основных шага для обнаружения и анализа ошибок во входных данных**

Три основных шага при анализе входных данных для защиты от ошибок:

1.Необходимо проверять все данные из внешних по отношению к программе источников.

2. Необходимо проверять значения всех входных параметров метода.

3. Необходимо решить, как обрабатывать неправильные входные данные.

**5. Каким методом можно обнаружить ошибки в данных**

Чаще всего в процессе моделирования при проектировании системы на различных наборах исходных данных становятся известными границы изменения входных и выходных данных нормально функционирующей системы и нормально функционирующего ПО. Если текущие значения выходных и входных данных в процессе контроля находятся внутри данного диапазона - внутри граничных «ворот», то мы считаем, что система и ПО работают правильно. Если же выходные либо входные данные выходят за пределы назначенных границ, то это означает, что работа ПО и системы не верна.

**6. Каким методом можно обнаружить не выполнение программой требуемых временных соотношений?**

Часто выполнение функций программы привязаны ко времени. Если заданное время наступило, а функция, управляемая от программы, не завершена, то это признак неправильной работы ПО или системы. Такой встроенный в ПО контроль называется «Завершением заданной функции за заданное время». В этом случае сигналы о фактическом начале и завершении выполнения какой-то внешней системной функции ПО могут быть заведены в ПО, например, через систему прерывания ЦВМ. В ПО через устройство отсчёта времени ЦВМ на момент начала исполнения заносится контрольная рассчитанная в ПО же уставка времени. Если эта уставка времени истекла, а сигнал о завершении внешней системной функции не поступил, то ПО вырабатывает «ошибку».

**7. Перечислить встроенные в ПО низкоуровневые методы обработки обнаруженных ошибок во входных и выходных данных**

1. Выдача нейтрального значения или возврат ближайшего допустимого значения

2. Выдача в качестве результата работы на такте, в котором обнаружена ошибка, заведомо безопасного значения

3. Замена ошибочных данных следующим корректным блоком данных- метод пропуска такта

4. Выдача того же результата, что и на предыдущем такте

5. Прекращение выполнения ПО

Выполнил: Шабакаев А.Р.

Группа 6411

**1.Какие виды низкоуровневых средств обнаружения ошибочных ситуаций в ПО Вам известны?**

**Внутренние, встроенные в ПО:**

- Средства обнаруживающие ошибочные значения данных.

- Средства самой программы, контролирующие её функционал, которые могут обнаруживать неправильное функционирование программы во времени.

- Средства, встроенные в саму программу и имеющие характер «утверждений». Утверждения проверяют условия наличия событий, которые никогда в нормально работающей программе не должны происходить.

**Внешние по отношению к ПО**, находящиеся в системе программирования, операционной системе или в аппаратных средствах.

**2. Привести примеры исключительных ситуаций, обнаруживаемых внешними по отношению к ПО средствами**

Внешними средствами для ПО могут быть ОС, либо программный отладчик и другие средства.

ОС может обнаружить утечку памяти, например, при работе с динамическими массивами в программе, написанной на языке C++, где управление динамической памятью полностью осуществляется разработчиком.

Отладчик кода способен искать ошибки в программе, написанной человеком, и указывать на них, отмечая вид ошибки и даже строку, где допущена ошибка с точки зрения отладчика.

**3.Можно ли программе игнорировать возникшую исключительную ситуацию и продолжать работать?**

Игнорирование исключительных ситуаций возможна для программы, если она не задействована в критических сферах деятельности или производства, и когда работоспособность программы важнее результата работы программы. В целом данный метод устранения ошибок опасен, так как приводит к ошибочным результатам работы программ или возникновению необратимых ошибок в последствии, что может только усугубить ошибочную ситуацию либо привести к непредсказуемым последствиям.

**4. Перечислите три основных шага для обнаружения и анализа ошибок во входных данных.**

Первый шаг: необходимо проверять все данные из внешних по отношению к программе источников. Проверять все значения на попадание в допустимый интервал, также проверять допустимость того или иного набора значений.

Второй шаг: необходимо проверять значения всех входных параметров метода. Эшелонированная оборона программы, помимо внешнего интерфейса, также надо проверять все входные параметры методов ПО.

Третий шаг: необходимо решить, как обрабатывать неправильные входные данные. Что делать, если обнаружен неверный параметр или значение?

**5.Каким методом можно обнаружить ошибки в данных**

Для обнаружения ошибок в данных применяются **граничные «ворота» данных**. Всем входным и выходным данным задается граничное значение, за границы которого они не должны выходить.

**6.Каким методом можно обнаружить не выполнение программой требуемых временных соотношений**

Не выполнение программой требуемых временных соотношений можно обнаруживать методом **«Завершения заданной функции за заданное время»**. Заключается метод в том, что выполнение функций программы привязаны ко времени. Если заданное время наступило, а функция, управляемая от программы, не завершена, то это признак неправильной работы ПО или системы. Для отсчета фактического времени начала и завершения выполнения программа используем устройство отсчета времени ЦВМ, где заносится контрольная, рассчитанная в самом же ПО, уставка времени. Если эта уставка времени истекла, а сигнал о завершении внешней системной функции не поступил, то ПО вырабатывает «ошибку».

**7.Перечислить встроенные в ПО низкоуровневые методы обработки обнаруженных ошибок во входных и выходных данных**

- Выдать нейтральное значение или вернуть ближайшее допустимое значение;

- Продолжение выполнения задачи с выдачей в качестве результата работы на такте, в котором обнаружена ошибка, заведомо безопасного значения;

- Заменить ошибочные данные следующим корректным блоком данных- метод пропуска такта: не брать в обработку неверное значение, подождать (пропустить такт);

- Выдать тот же результат, что и на предыдущем такте;

- Выдать тот же результат, что и на предыдущем такте;

- Прекратить выполнение.

Выполнил: Щекочихин Д.А  
группа 6411

**1. Какие виды низкоуровневых средств обнаружения ошибочных ситуаций в ПО Вам известны?**

1. Граничные «ворота» данных.

2. «Завершением заданной функции за заданное время».

3. Утверждения проверяют условия наличия событий, которые никогда в нормально работающей программе не должны происходить.

**2. Привести примеры исключительных ситуаций, обнаруживаемых внешними по отношению к ПО средствами.**

1. При целочисленном делении на ноль конечного результата у данной операции быть не может. Поэтому ни дальнейшие вычисления, ни попытка использования результата такого деления не приведут к решению задачи.

2. Проигнорировав ошибку чтения из файла блока данных, программа получит в своё распоряжение не те данные, которые она должна была считать, а какие-то другие. Результаты их использования предугадать невозможно.

**3. Можно ли программе игнорировать возникшую исключительную ситуацию и продолжать работать?**

Простое игнорирования исключительной ситуации и продолжение работы программы – опасная тактика, так как приводит к ошибочным результатам работы программ или возникновению необратимых ошибок в последствии, что может только усугубить ошибочную ситуацию. Однако, прекращение исполнения программ во многих случаях в частности при управлении СТС не является наилучшей реакцией, поскольку за этим может стоять и фатальное прекращение функционирования системы.

**4. Перечислите три основных шага для обнаружения и анализа ошибок во входных данных.**

1.Необходимо проверять все данные из внешних по отношению к программе источников. Получив данные из файла, от пользователя, из сети или любого другого внешнего интерфейса, необходимо удостоверится, что все значения попадают в допустимый интервал. Если строка должна содержать определенный набор, необходимо проконтролировать, что этот набор значений имеет место в данном случае, если же нет - эти данные следует отклонить.

2. Необходимо проверять значения всех входных параметров метода. Проверка значений входных параметров метода практически то же самое, что и проверка данных из внешнего источника, за исключением того, что данные поступают из другого метода, а не из внешнего интерфейса. Этот подход можно рассматривать как реализацию принципа эшелонированной обороны.

3. Необходимо решить, как обрабатывать неправильные входные данные.

**5. Каким методом можно обнаружить ошибки в данных.**

Чаще всего в процессе моделирования при проектировании системы на различных наборах исходных данных становятся известными границы изменения входных и выходных данных нормально функционирующей системы и нормально функционирующего ПО – граничные «ворота» данных, соответствующие устраивающему нас функционированию ПО и системы. Тогда, если текущие значения выходных и входных данных в процессе контроля находятся внутри данного диапазона - внутри граничных «ворот», то мы считаем, что система и ПО работают правильно. Если же выходные либо входные данные выходят за пределы назначенных границ, то это означает, что работа ПО и системы не верна (либо границы назначены неправильно). Такой контроль результатов работы ПО применяется повсеместно.

**6. Каким методом можно обнаружить не выполнение программой требуемых временных соотношений.**

Часто выполнение функций программы привязаны ко времени. Если заданное время наступило, а функция, управляемая от программы, не завершена, то это признак неправильной работы ПО или системы. Такой встроенный в ПО контроль мы будем называть «Завершением заданной функции за заданное время». В этом случае сигналы о фактическом начале и завершении выполнения какой-то внешней системной функции ПО могут быть заведены в ПО, например, через систему прерывания ЦВМ. В ПО через устройство отсчёта времени ЦВМ на момент начала исполнения заносится контрольная рассчитанная в ПО же уставка времени. Если эта уставка времени истекла, а сигнал о завершении внешней системной функции не поступил, то ПО вырабатывает «ошибку». Иногда срабатывание этого контроля говорит не о наличии ошибки в ПО, а о наличие ошибки в управляемой от ПО аппаратуре – системной ошибке. В этом случае для точной диагностики необходимо привлечение дополнительной информации.

**7. Перечислить встроенные в ПО низкоуровневые методы обработки обнаруженных ошибок во входных и выходных данных.**

1. Выдать нейтральное значение или вернуть ближайшее допустимое значение.

2. Наилучшей реакцией на обнаруженные неправильные данные будет продолжение выполнения задачи с выдачей в качестве результата работы на такте, в котором обнаружена ошибка, заведомо безопасного значения.

3. Заменить ошибочные данные следующим корректным блоком данных- метод пропуска такта.

4. Выдать тот же результат, что и на предыдущем такте (если программа считывания показаний термометра, один раз не получила измерение, она может просто вернуть то же значение, что и в предыдущий раз).

5. Прекратить выполнение.

Выполнил: Ковалев К.А  
группа 6411

**1.Какие виды низкоуровневых средств обнаружения ошибочных ситуа-ций в ПО Вам известны?**

1. Средства обнаруживающие ошибочные значения данных

2.Средства самой программы, контролирующие её функционал, которые могут обнаруживать неправильное функционирование программы во времени.

3. Средства, встроенные в саму программу и имеющие характер «утверждений» (Утверждения проверяют условия наличия событий, которые никогда в нормально работающей программе не должны происходит).

**2. Привести примеры исключительных ситуаций, обнаруживаемых внешними по отношению к ПО средствами**

1. При целочисленном делении на ноль конечного результата у данной операции быть не может. Поэтому ни дальнейшие вычисления, ни попытка использования результата такого деления не приведут к решению задачи.

2. Проигнорировав ошибку чтения из файла блока данных, программа получит в своё распоряжение не те данные, которые она должна была считать, а какие-то другие. Результаты их использования предугадать невозможно.

**3.Можно ли программе игнорировать возникшую исключительную ситуацию и продолжать работать?**

Игнорирование исключительной ситуации и продолжение работы программы может привести к ошибочным результатам работы программы или возникновению необратимых ошибок в последствии, что может только усугубить ошибочную ситуацию. Поэтому игнорировать возникшую исключительную ситуацию нельзя.

**4. Перечислите три основных шага для обнаружения и анализа ошибок во входных данных.**

1.Необходимо проверять все данные из внешних по отношению к программе источников.

2. Необходимо проверять значения всех входных параметров метода.

3. Необходимо решить, как обрабатывать неправильные входные данные.

**5.Каким методом можно обнаружить ошибки в данных**

Например, анализируя границы изменения входных и выходных данных системы. Если текущие значения выходных и входных данных в процессе контроля находятся внутри заданного диапазона - внутри граничных «ворот», то мы считаем, что система и ПО работают правильно. Если же выходные либо входные данные выходят за пределы назначенных границ, то это означает, что работа ПО и системы не верна.

**6.Каким методом можно обнаружить не выполнение программой требуемых временных соотношений**

Часто выполнение функций программы привязаны ко времени. Если заданное время наступило, а функция, управляемая от программы, не завершена, то это признак неправильной работы ПО или системы. Такой встроенный в ПО контроль называется «Завершением заданной функции за заданное время». В этом случае сигналы о фактическом начале и завершении выполнения какой-то внешней системной функции ПО могут быть заведены в ПО, например, через систему прерывания ЦВМ. В ПО через устройство отсчёта времени ЦВМ на момент начала исполнения заносится контрольная рассчитанная в ПО же уставка времени. Если эта уставка времени истекла, а сигнал о завершении внешней системной функции не поступил, то ПО вырабатывает «ошибку».

**7.Перечислить встроенные в ПО низкоуровневые методы обработки обнаруженных ошибок во входных и выходных данных**

1. Выдача нейтрального значения или возврат ближайшего допустимого значения

2. Выдача в качестве результата работы на такте, в котором обнаружена ошибка, заведомо безопасного значения

3. Замена ошибочных данных следующим корректным блоком данных- метод пропуска такта

4. Выдача того же результата, что и на предыдущем такте

5. Прекращение выполнения ПО