**Вопросы по лекции №11.**

**Защитное программирование – создание устойчивых к**

**ошибкам программ Виды контроля работы 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. Варианты реакции ПО на проявившуюся ошибку в устойчивых к ошибкам программах**

В зависимости от проявившейся ошибки может быть и необходимо прекратить исполнение ПО, но сделать это надо организовано и безопасно, выключив всю включенную программой аппаратуру, могущую причинить непоправимый ущерб, и собрав всю необходимую для анализа причин ошибки информацию. Это тоже толерантность к ошибкам. Возможны и другие варианты реакции программы на обнаруженную ошибку, связанные с продолжением работы ПО или с продолжением работы со снижением качества функционирования ПО и системы.

**2. Виды контроля работы ПО.**

1. Автоматический контроль работы ПО встроенными в ПО средствами, когда некоторые части работающего ПО осуществляют контроль правильности его работы и работы системы.

2.Контроль работы ПО в точках его останова внешними по отношению к ПО средствами.

3. Контроль путем сбора и представления на более высокий уровень иерархии или оператору данных по работе ПО, полученных результатах и т.п. Сбор и представление этих данных должен происходить автоматически внутри работающего ПО фрагментами самого ПО.

Разница между п.1 и п.3 в том, что в первом случае анализ и принятие решения происходит сразу же и в самом ПО.

**3. Контроль работы ПО встроенными средствами без прекращения его функционирования**

С целью раннего обнаружения ошибок проводится контроль работы ПО встроенными в него программными средствами.

Автоматический контроль работы ПО встроенными в ПО средствами, когда некоторые части работающего ПО осуществляют контроль правильности его работы и работы системы. В этом случае контроль работы ПО осуществляется в реальном времени и без прекращения нормального функционирования системы. Правда, это приводит к увеличению размера ПО до ~20%.

**4. Как убедится в правильности полученных результатов работы ПО? Эталоны работы ПО и методы их получения.**

Сравниваем эти результаты с эталоном – заведомо правильными результатами решения задачи. Часто эталоны нам известны, как желаемые результаты работы ПО, в других случаях эталоны получаются, как результат моделирования системы на стадии ее проектирования. Распространенный прием - провести проверку работы программы или ПО в целом на варианте с специально подобранными исходными данными, на которых эталонный (правильный) результат известен или очевиден, но при этом алгоритм работы ПО не упрощается.

Если таких эталонов нет, то есть еще один выход – получить их, приближенно решая ту же задачу по-другому более простому алгоритму, результат работы которого более очевиден. Это сравнение с эталонным результатом можно делать по завершению работы ПО, а можно делать непрерывно (точнее квазинепрерывно). Учитывая периодичность работы ПО в составе систем управления, период проведения контроля целесообразно совместить с периодом работы ПО системы

**5. Что должна сделать частная программа комплекса взаимосвязанных программ, обнаружив у себя ошибочные входные или выходные данные. Зачем и в каких случаях нужен второй системный уровень защиты большого ПО от ошибок**

Решение о продолжении или организованном прекращении работы ПО, обнаружив у себя ошибочные входные или выходные данные, должно приниматься на системном втором уровне защиты ПО, который пишется специальными системными программистами, которые обладают знаниями о том как работает ПО в целом и как влияет та или иная ошибка на его работу. Во многих случаях низкоуровневая защита может и должна вызывать системную защиту, но это может происходить не всегда.

Системный уровень защиты ПО необходим для защиты программных комплексов от ошибок, т.к. не известно, как влияет результат программного модуля на работу всей системы.

**6. Иерархический принцип построения встроенной в ПО защиты от ошибок. Низкоуровневая и высокоуровневая встроенная защита от ошибок в программах сложного ПО**

Устойчивость (толерантность) ПО к ошибкам

Автоматический контроль работы ПО встроенными в него средствами (контроль диапазона изменения, выполнение функции за заданное время, утверждения)

Обобщённый контроль работы ПО средствами ЦВМ

Обобщённый контроль работы прикладного ПО средствами ОС

Наличие низкоуровневой защиты в программах и системного уровня защиты для ПО в целом

Передача обработки ошибки на системный уровень. Аварийная защита

Методы обработки обнаруженных ошибок в низкоуровневых программах

Конструирование низкоуровневой защиты от ошибок на низком уровне частных программ и модулей и аварийной защиты всего ПО на системном уровне обязательно должно быть использовано для создания качественного и безопасного программного обеспечения.

Таким образом, защита от ошибок, а точнее придание программному обеспечению свойств информационной устойчивости к ошибкам (толерантности) должна быть построена по иерархическому принципу.

Выполнил: Щекочихин Д.А.

Группа 6411

**1.Варианты реакции ПО на проявившуюся ошибку в устойчивых к ошибкам программах.**

а) Прекращение исполнения ПО, осуществленное организовано и безопасно.

б) Продолжение работы ПО.

в) Продолжение работы со снижением качества функционирования ПО и системы.

**2. Виды контроля работы ПО.**

а) Автоматический контроль работы ПО встроенными в ПО средствами.

б) Контроль работы ПО в точках его останова внешними по отношению к ПО средствами.

в) Контроль путем сбора и представления на более высокий уровень иерархии или оператору данных по работе ПО, полученных результатах и т.п.

**3. Контроль работы ПО встроенными средствами без прекращения его функционирования.**

Автоматический контроль работы ПО встроенными в ПО средствами, когда некоторые части работающего ПО осуществляют контроль правильности его работы и работы системы осуществляется в реальном времени и без прекращения нормального функционирования системы. Правда, это приводит к увеличению размера ПО до ~20%.

**4. Как убедится в правильности полученных результатов работы ПО? Эталоны работы ПО и методы их получения.**

Мы сравниваем полученные результаты с эталоном – заведомо правильными результатами решения задачи, для того чтобы убедится в правильности полученных результатов работы ПО. Часто эталоны нам известны, как желаемые результаты работы ПО, в других случаях эталоны получаются, как результат моделирования системы. Распространенный прием - провести проверку работы программы или ПО в целом на варианте с специально подобранными исходными данными, на которых эталонный (правильный) результат известен или очевиден, но при этом алгоритм работы ПО не упрощается.

Еще один выход – получить эталон приближенно, решая ту же задачу по-другому более простому алгоритму, результат работы которого более очевиден. Это сравнение с эталонным результатом можно делать по завершению работы ПО, а можно делать непрерывно (точнее квазинепрерывно).

**5. Что должна сделать частная программа комплекса взаимосвязанных программ, обнаружив у себя ошибочные входные или выходные данные. Зачем и в каких случаях нужен второй системный уровень защиты большого ПО от ошибок.**

Решение о продолжении или организованном прекращении работы ПО при обнаружении у себя ошибочных входных или выходных данных должна приниматься на системном втором уровне защиты ПО, так как устойчивая к ошибкам программа может «не догадываться» (конечно, не догадываются написавшие их программисты) как некорректность результата ее работы скажется на работоспособности ПО или СТС в целом. Поэтому частная программа комплекса передает обработку ошибки на системный уровень.

Системный уровень защиты ПО необходим для защиты программных комплексов от ошибок

.

**6. Иерархический принцип построения встроенной в ПО защиты от ошибок. Низкоуровневая и высокоуровневая встроенная защита от ошибок в программах сложного ПО.**

****

Конструирование низкоуровневой защиты от ошибок на низком уровне частных программ и модулей, и аварийной защиты всего ПО на системном уровне обязательно должно быть использовано для создания качественного и безопасного программного обеспечения. Таким образом, защита от ошибок, а точнее придание программному обеспечению свойств информационной устойчивости к ошибкам (толерантности) должна быть построена по иерархическому принципу

Выполнил: Шабакаев А.Р.

Группа 6411

**1.Варианты реакции ПО на проявившуюся ошибку в устойчивых к ошибкам программах**

- Прекращение исполнения ПО, осуществленное организовано и безопасно

- Продолжение работы ПО

- Продолжение работы со снижением качества функционирования ПО и системы

Принцип создания ПО, устойчивого к ошибкам: хорошая программа никогда не выдаст «мусор» на выходе независимо от того, что у нее было на входе. Вместо этого она должна сработать: «мусор на входе — ничего на выходе», «мусор на входе — сообщение об ошибке на выходе» или «мусор на входе – замена на не точный, но допустимый результат на выходе».

**2. Виды контроля работы ПО**

* Автоматический контроль работы ПО встроенными в ПО средствами
* Контроль работы ПО в точках его останова внешними по отношению к ПО средствами.
* Контроль путем сбора и представления на более высокий уровень иерархии или представления оператору данных по работе ПО, полученных результатов и т.п.

Разница между п.1 и п.3 в том, что в первом случае анализ и принятие решения происходит сразу же и в самом ПО.

**3. Контроль работы ПО встроенными средствами без прекращения его функционирования**

Автоматический контроль работы ПО встроенными в ПО средствами, когда некоторые части работающего ПО осуществляют контроль правильности его работы и работы системы. В этом случае контроль работы ПО осуществляется в реальном времени и без прекращения нормального функционирования системы. Правда, это приводит к увеличению размера ПО до ~20%.

О контроле без прекращения функционирования системы можно говорить и в случае контроля путем сбора и представления данных и результатов оператору по работе ПО, но запаздывание оператора на полученную контрольную информацию будет велико и в большинстве случаев не соответствует требованиям автоматических систем реального времени

**4. Как убедится в правильности полученных результатов работы ПО? Эталоны работы ПО и методы их получения**

В правильности полученных результатов работы ПО можем убедиться с помощью сравнения с эталоном, особенно когда получили первый результат работы ПО, когда не знаем, правильный результат или нет.

Эталон – заведомо правильные результаты решения задач.

Методы получения:

- уже известны, как желаемые результаты работы ПО;

- получение, как результат моделирования системы на стадии её проектирования;

- использование специально подобранных исходных данных, при которых правильный результат известен и очевиден (без упрощения алгоритма работы ПО)

- решение той же задачи по более простому алгоритму, результат которого более очевиден.

**5. Что должна сделать частная программа комплекса взаимосвязанных программ, обнаружив у себя ошибочные входные или выходные данные. Зачем и в каких случаях нужен второй системный уровень защиты большого ПО от ошибок**

Решение о продолжении или организованном прекращении работы ПО в случае обнаружения ошибочных входных или выходных данных частной программой должно приниматься на системном втором уровне защиты ПО, так как устойчивая к ошибкам программа может «не догадываться» (конечно, не догадываются написавшие их программисты) как некорректность результата их работы скажется на работоспособности ПО или СТС в целом. Следовательно, частная программа должна передавать обработку ошибки на системный уровень.

Второй системный уровень защиты большого ПО от ошибок нужен для защиты программного комплекса от ошибок так как неизвестно, как влияет результат отдельной программы-модуля на весь комплекс.

**6. Иерархический принцип построения встроенной в ПО защиты от ошибок. Низкоуровневая и высокоуровневая встроенная защита от ошибок в программах сложного ПО**

Конструирование низкоуровневой защиты от ошибок на низком уровне частных программ и модулей, и аварийной защиты всего ПО на системном уровне обязательно должно быть использовано для создания качественного и безопасного программного обеспечения.

Защита от ошибок, а точнее придание программному обеспечению свойств информационной устойчивости к ошибкам (толерантности) должна быть построена по иерархическому принципу.

**Наличие низкоуровневой защиты в программах и системного уровня защиты для ПО в целом**

**Методы обработки обнаруженных ошибок в низкоуровневых программах**

**Устойчивость(толерантность) ПО к ошибкам**

**Автоматический контроль работы ПО встроенными в него средствами (контроль диапазона изменения, выполнение функции за заданное время, утверждения)**

**Обобщённый контроль работы ПО средствами ЦВМ (аппаратными)**

**Обобщённый контроль работы прикладного ПО средствами ОС**

**Передача обработки ошибки на системный уровень. Аварийная защита**

Выполнил: Терентьев Е.А.

Группа 6411

**1. Варианты реакции ПО на проявившуюся ошибку в устойчивых к ошибкам программах.**

В зависимости от проявившейся ошибки может быть несколько вариантов реакции ПО на проявившуюся ошибку:

1. Прекращение исполнения ПО, осуществленное организовано и безопасно
2. Продолжение работы ПО
3. Продолжение работы со снижением качества функционирования ПО и системы

Для создания ПО, устойчивого к ошибкам, необходимо использовать принцип: хорошая программа никогда не выдаст «мусор» на выходе независимо от того, что у нее было на входе. Вместо этого она должна сработать: «мусор на входе — ничего на выходе», «мусор на входе — сообщение об ошибке на выходе» или «мусор на входе – замена на не точный, но допустимый результат на выходе».

**2. Виды контроля работы ПО.**

Контроль работы ПО проводится с целью раннего обнаружения ошибок. Чем раньше обнаружена ошибка, тем ранее будет прекращена неправильная работа ПО, тем к меньшему ущербу эта неправильная работа ПО приведет

Виды контроля:

1. Автоматический контроль работы ПО встроенными в ПО средствами

2. Контроль работы ПО в точках его останова внешними по отношению к ПО средствами.

3. Контроль путем сбора и представления на более высокий уровень иерархии или оператору данных по работе ПО, полученных результатах и т.п.

Разница между п.1 и п.3 в том, что в первом случае анализ и принятие решения происходит сразу же и в самом ПО.

**3. Контроль работы ПО встроенными средствами без прекращения его функционирования.**

Автоматический контроль работы ПО встроенными в ПО средствами, когда некоторые части работающего ПО осуществляют контроль правильности его работы и работы системы осуществляется в реальном времени и без прекращения нормального функционирования системы. Правда, это приводит к увеличению размера ПО до ~20%.

**4. Как убедится в правильности полученных результатов работы ПО? Эталоны работы ПО и методы их получения.**

Для того чтобы убедиться в правильности работы ПО, мы сравниваем полученные результаты с эталоном – заведомо правильными результатами решения задачи. Часто эталоны нам известны, как желаемые результаты работы ПО, в других случаях эталоны получаются, как результат моделирования системы. Распространенный прием - провести проверку работы программы или ПО в целом на варианте с специально подобранными исходными данными, на которых эталонный (правильный) результат известен или очевиден, но при этом алгоритм работы ПО не упрощается.

Еще один выход – получить эталон приближенно, решая ту же задачу по-другому более простому алгоритму, результат работы которого более очевиден.

Сравнение с эталонным результатом можно делать по завершению работы ПО, а можно делать непрерывно (точнее квазинепрерывно).

**5. Что должна сделать частная программа комплекса взаимосвязанных программ, обнаружив у себя ошибочные входные или выходные данные. Зачем и в каких случаях нужен второй системный уровень защиты большого ПО от ошибок.**

Решение о продолжении или организованном прекращении работы ПО при обнаружении частной программой комплекса у себя ошибочных входных или выходных данных приниматься на системном втором уровне защиты ПО, так как устойчивая к ошибкам программа может «не догадываться» как некорректность результата ее работы скажется на работоспособности ПО или СТС в целом. Поэтому частная программа комплекса передает обработку ошибки на системный уровень.

Системный уровень защиты ПО необходим для защиты программных комплексов от ошибок, т.к. не известно, как влияет результат программного модуля на работу всей системы.

**6. Иерархический принцип построения встроенной в ПО защиты от ошибок. Низкоуровневая и высокоуровневая встроенная защита от ошибок в программах сложного ПО**

Конструирование низкоуровневой защиты от ошибок на низком уровне частных программ и модулей, и аварийной защиты всего ПО на системном уровне обязательно должно быть использовано для создания качественного и безопасного программного обеспечения.

Защита от ошибок, а точнее придание программному обеспечению свойств информационной устойчивости к ошибкам (толерантности) должна быть построена по иерархическому принципу.

**Наличие низкоуровневой защиты в программах и системного уровня защиты для ПО в целом**

**Методы обработки обнаруженных ошибок в низкоуровневых программах**

**Устойчивость(толерантность) ПО к ошибкам**

**Автоматический контроль работы ПО встроенными в него средствами (контроль диапазона изменения, выполнение функции за заданное время, утверждения)**

**Обобщённый контроль работы ПО средствами ЦВМ (аппаратными)**

**Обобщённый контроль работы прикладного ПО средствами ОС**

**Передача обработки ошибки на системный уровень. Аварийная защита**

Выполнил: Ковалев К.А.

Группа 6411

**1.Варианты реакции ПО на проявившуюся ошибку в устойчивых к**

**ошибкам программах**

1) Безопасное и организованное прекращение исполнения ПО.

2) Продолжение работы ПО.

3) Продолжение работы со снижением качества функционирования ПО и системы.

**2. Виды контроля работы ПО.**

1) Автоматический контроль работы ПО встроенными в ПО средствами, когда некоторые части работающего ПО осуществляют контроль правильности его работы и работы системы.

2) Контроль работы ПО в точках его останова внешними по отношению к ПО средствами.

3) Контроль путем сбора и представления на более высокий уровень иерархии или оператору данных по работе ПО, полученных результатах и т.п. Сбор и представление этих данных должен происходить автоматически внутри работающего ПО фрагментами самого ПО. Разница между п.1 и п.3 в том, что в первом случае анализ и принятие решения происходит сразу же и в самом ПО.

**3. Контроль работы ПО встроенными средствами без прекращения его функционирования**

Автоматический контроль работы ПО встроенными в ПО средствами, когда некоторые части работающего ПО осуществляют контроль правильности его работы и работы системы. В этом случае контроль работы ПО осуществляется в реальном времени и без прекращения нормального функционирования системы. Правда, это приводит к увеличению размера ПО до ~20%.

**4. Как убедится в правильности полученных результатов работы ПО ? Эталоны работы ПО и методы их получения**

Для того чтобы убедиться в правильности работы ПО, проводится сравнение полученных результатов с эталоном – заведомо правильными результатами решения задачи. Часто эталоны нам известны, как желаемые результаты работы ПО, в других случаях эталоны получаются, как результат моделирования системы. Распространенный прием - провести проверку работы программы или ПО в целом на варианте с специально подобранными исходными данными, на которых эталонный (правильный) результат известен или очевиден, но при этом алгоритм работы ПО не упрощается.

Еще один выход – получить эталон приближенно, решая ту же задачу по-другому более простому алгоритму, результат работы которого более очевиден.

Сравнение с эталонным результатом можно делать по завершению работы ПО, а можно делать квазинепрерывно.

**5. Что должна сделать частная программа комплекса взаимосвязанных**

**программ, обнаружив у себя ошибочные входные или выходные данные.**

**Зачем и в каких случаях нужен второй системный уровень защиты большого**

**ПО от ошибок**

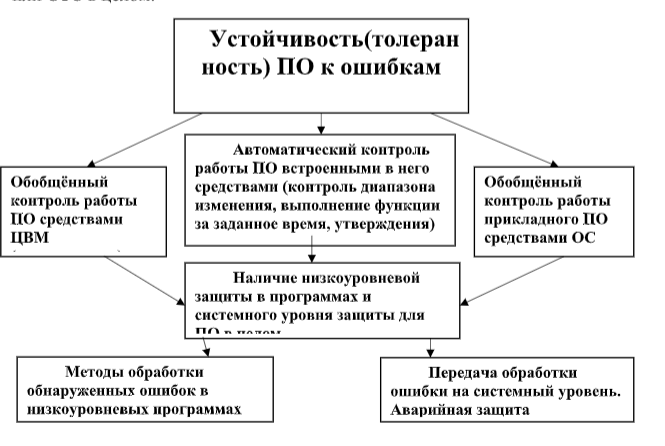
Решение о продолжении или организованном прекращении работы ПО, обнаружив у себя ошибочные входные или выходные данные, должно приниматься на системном втором уровне защиты ПО, который пишется специальными системными программистами, которые обладают знаниями о том как работает ПО в целом и как влияет та или иная ошибка на его работу. Во многих случаях низкоуровневая защита может и должна вызывать системную защиту, но это может происходить не всегда.

Системный уровень защиты ПО необходим для защиты программных комплексов от ошибок, т.к. не известно, как влияет результат программного модуля на работу всей системы.

**6. Иерархический принцип построения встроенной в ПО защиты от**

**ошибок . Низкоуровневая и высокоуровневая встроенная защита от ошибок в**

**программах сложного ПО**



Конструирование низкоуровневой защиты от ошибок на низком уровне частных программ и модулей, и аварийной защиты всего ПО на системном уровне обязательно должно быть использовано для создания качественного и безопасного программного обеспечения.

Таким образом, защита от ошибок, а точнее придание программному обеспечению свойств информационной устойчивости к ошибкам (толерантности) должна быть построена по иерархическому принципу.