

Виды ошибок. Классификация ошибок.

Отладкой называют процесс локализации и исправления ошибок, обнаруженных при тестировании программного обеспечения.

Локализацией является определение оператора/операторов программы, выполнение которого вызвало нарушение вычислительного процесса.

Для исправления ошибки необходимо определить ее причину, т.е. установить оператор или фрагмент, содержащие ошибку. Причины ошибок могут быть как очевидны, так и очень глубоко скрыты.

Классификация ошибок в соответствии с этапом обработки:

- *ошибки компиляции (синтаксические ошибки)* – ошибки, фиксируемые компилятором (транслятором, интерпретатором) при выполнении синтаксического и частично семантического анализа программы;
- *ошибки компоновки* – ошибки, обнаруженные компоновщиком (редактором связей) при объединении модулей программы;
- *ошибки выполнения* – ошибки, обнаруженные операционной системой, аппаратными средствами или пользователем при выполнении программы.

Ошибки компиляции относят к группе самых простых, так как синтаксис языка, как правило, строго формализован и ошибки сопровождаются развернутым комментарием с указанием ее местоположения. Следует иметь в виду, что чем лучше формализованы правила синтаксиса языка, тем больше ошибок из общего количества может обнаруживать компилятор и, соответственно, меньше ошибок будет обнаруживаться на следующих этапах. В связи с этим говорят о языках программирования с защищенным синтаксисом и незащищенным синтаксисом.

Ошибки компоновки связаны с проблемами, обнаруженными при разрешении внешних ссылок. Например, предусмотрено обращение к подпрограмме другого модуля, а при объединении модулей данная подпрограмма не найдена или не стыкуются списки параметров. В большинстве случаев ошибки такого рода также удастся быстро локализовать и устранить.

Ошибки выполнения являются самыми непредсказуемыми. Они могут иметь разную природу и соответственно по-разному проявляться. Часть ошибок обнаруживается и документируется операционной системой. Ошибки выполнения могут проявиться следующим образом:

- появление сообщения об ошибке, зафиксированной схемами контроля выполнения машинных команд, например, переполнении разрядов, деления на ноль, нарушении адресации;
- появление сообщения об ошибке, обнаруженной операционной системой, например, нарушении защиты памяти, попытке записи на устройства, защищенные от записи, отсутствии файла с заданным именем;
- «зависание» компьютера – иногда возможно завершить программу без перезагрузки операционной системы, а бывает, что для продолжения работы необходима перезагрузка;
- несовпадение полученных результатов с ожидаемыми.

Причины ошибок выполнения очень разнообразны, а потому и локализация может оказаться крайне сложной. Все возможные причины ошибок подразделяют на следующие группы:

- неверное определение исходных данных;
- логические ошибки;
- накопление погрешностей результатов вычислений.



Неверное определение исходных данных происходит, если возникают любые ошибки при выполнении операций ввода-вывода: ошибки передачи, ошибки преобразования, ошибки перезаписи и ошибки данных. Причем использование специальных технических средств и программирование с защитой от ошибок позволяет обнаружить и предотвратить только часть этих ошибок, о чем, безусловно, не следует забывать.

Логические ошибки имеют разную природу. Так, они могут следовать из ошибок, допущенных при проектировании, например, при выборе методов, разработке алгоритмов или определении структуры классов, а могут быть непосредственно внесены при кодировании модуля.

К ошибкам *кодирования* относят:

- *ошибки некорректного использования переменных*, например, неудачный выбор типов данных, использование переменных до их инициализации,

использование индексов, выходящих за границы определения массивов, нарушения соответствия типов данных при использовании явного или неявного переопределения типа данных, расположенных в памяти при использовании нетипизированных переменных, открытых массивов, объединений, динамической памяти, адресной арифметики;

- *ошибки вычислений*, например, некорректное использование неарифметических переменных, некорректная работа с целочисленными переменными, некорректное преобразование типов данных в процессе вычислений;
- *ошибки взаимодействия модулей*, т.е. межмодульного интерфейса, например, нарушение типов и последовательности при передаче параметров, несоблюдение единства единиц измерения формальных и фактических параметров, нарушение области действия локальных и глобальных переменных;
- игнорирование особенностей или ограничений конкретного языка программирования.

Накопление погрешностей результатов числовых вычислений возникает, например, при некорректном отбрасывании дробных чисел.

Процесс отладки является сложным процессом, требующим от программиста глубоких знаний специфики управления используемыми техническими средствами, операционной системы, среды и языка программирования, реализуемых процессов, природы и специфики различных ошибок, методик отладки и соответствующих программных средств. Сложность отладки может возрастать вследствие влияния следующих факторов:

- опосредованного проявления ошибок;
- возможности взаимного влияния ошибок;
- возможности получения внешне одинаковых проявлений разных ошибок;
- отсутствия повторяемости проявлений некоторых ошибок от запуска к запуску (стохастические ошибки);
- возможности устранения внешних проявлений ошибок в исследуемой ситуации при внесении некоторых изменений в программу, например, при включении в программу диагностических фрагментов может аннулироваться или измениться внешнее проявление ошибок;
- написания отдельных частей программы разными программистами.

Рассмотрим некоторые *категории программных ошибок*, которые встречаются наиболее часто.

Функциональные недостатки. Данные недостатки присущи программе, если она не делает того, что должна, выполняет одну из своих функций плохо или не полностью. Функции программы должны быть подробно описаны в ее спецификации, и именно на основе утвержденной спецификации тестировщик строит свою работу.

Недостатки пользовательского интерфейса. Оценить удобство и правильность работы пользовательского интерфейса можно только в процессе работы с ним. Желательно, чтобы в этой работе принимал участие сам пользователь. Этого можно добиться с помощью разработки прототипа программного продукта, на котором проводятся обкатка и согласование всех требований к пользовательскому интерфейсу с дальнейшей фиксацией их в спецификации требований. После утверждения спецификации требований любые отклонения от нее или невыполнение последних являются ошибкой. Это в полной мере касается и пользовательского интерфейса.

Недостаточная производительность. При разработке некоторого программного продукта очень важной его характеристикой может оказаться скорость работы, иногда этот критерий задается в требованиях заказчика к программному продукту. Плохо, если у пользователя создается впечатление, что программа работает медленно, особенно если конкурирующие программы работают ощутимо быстрее, но еще хуже, если программа не удовлетворяет заданным в спецификации требованиям. Это уже ошибка, которая должна быть устранена.

Некорректная обработка ошибок. Процедуры обработки ошибок – очень важная часть программы. Правильно определив ошибку, программа должна выдать о ней сообщение. Отсутствие такого сообщения является ошибкой в работе программы.

Некорректная обработка граничных условий. Существует много различных граничных ситуаций. Любой аспект работы программы, к которому применимы понятия «больше» или «меньше», «раньше» или «позже», «первый» или «последний», «короче» или «длиннее», обязательно должен быть проверен на границах диапазона. Внутри диапазонов программа может работать прекрасно, а вот на их границах могут происходить самые неожиданные ситуации, которые в свою очередь приводят к ошибкам в работе программного обеспечения.

Ошибки вычислений. К ошибкам вычислений относятся ошибки, вызванные неправильным выбором алгоритма вычислений, неправильными формулами, формулами, не применимыми к обрабатываемым данным. Самыми распространенными среди ошибок вычислений являются ошибки округления.

Ошибки управления потоком. По логике работы программы вслед за первым действием должно быть выполнено второе. Если вместо этого выполняется третье или четвертое действие, значит, в управлении потоком допущена ошибка.

Ситуация гонок. Предположим, в системе ожидаются два события: «А» и «Б». Если первым наступит событие «А», то выполнение программы продолжится, а если событие «Б», то в работе программы произойдет сбой.

Разработчики предполагают, что первым всегда должно быть событие «А», и не ожидают, что «Б» может выиграть гонки и наступить раньше. Такова классическая ситуация гонок.

Тестировать ситуации гонок довольно сложно. Наиболее типичны они для систем, где параллельно выполняются взаимодействующие процессы и потоки, а также для многопользовательских систем реального времени. Ошибки в таких системах трудно воспроизвести, и на их выявление обычно требуется очень много времени.

Перегрузки. Сбои в работе программы могут происходить из-за нехватки памяти или отсутствия других необходимых системных ресурсов. У каждой программы свои пределы, программа может не справляться с повышенными нагрузками, например со слишком большими объемами данных. Вопрос в том, соответствуют ли реальные возможности программы ее требования к ресурсам спецификации программы и как она себя поведет при перегрузках.

Некорректная работа с аппаратурой компьютера. Программы могут отправлять аппаратным устройствам неверные данные, игнорировать их сообщения об ошибках, пытаться использовать устройства, которые заняты или вообще отсутствуют. Даже если нужное устройство просто сломано, программа должна понять это, а не «зависать» при попытке к нему обратиться.