

Методы поиска ошибок в программах. Классификация ошибок и тестов. Выявление ошибок системных компонентов

Отладка программы — один из самых сложных этапов разработки программного обеспечения, требующий глубокого знания:

- специфики управления используемыми техническими средствами,
- операционной системы,
- среды и языка программирования,
- реализуемых процессов,
- природы и специфики различных ошибок,
- методик отладки и соответствующих программных средств.

Отладка - это процесс локализации и исправления ошибок, обнаруженных при тестировании программного обеспечения. Локализацией называют процесс определения оператора программы, выполнение которого вызвало нарушение нормального вычислительного процесса. Для исправления ошибки необходимо определить ее причину, т. е. определить оператор или фрагмент, содержащие ошибку. Причины ошибок могут быть как очевидны, так и очень глубоко скрыты.

В целом сложность отладки обусловлена следующими причинами:

- требует 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;
- психологически дискомфортна, так как необходимо искать собственные ошибки и, как правило, в условиях ограниченного времени;
- возможно взаимовлияние ошибок в разных частях программы, например, за счет затирания области памяти одного модуля другим из-за ошибок адресации;
- отсутствуют четко сформулированные методики отладки.

В соответствии с этапом обработки, на котором проявляются ошибки, различают: 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.

Синтаксические ошибки. Синтаксические ошибки относят к группе самых простых, так как синтаксис языка, как правило, строго формализован, и ошибки сопровождаются развернутым комментарием с указанием ее местоположения. Определение причин таких ошибок, как правило, труда не составляет, и даже при нечетком знании правил языка за несколько прогонов удастся удалить все ошибки данного типа.

Следует иметь в виду, что чем лучше формализованы правила синтаксиса языка, тем больше ошибок из общего количества может обнаружить компилятор и, соответственно, меньше ошибок будет обнаруживаться на следующих этапах. В связи с этим говорят о языках программирования с защищенным синтаксисом и с незащищенным синтаксисом. К первым, безусловно, можно отнести Pascal, имеющий очень простой и четко определенный синтаксис, хорошо проверяемый при компиляции программы, ко вторым - Си со всеми его модификациями. Чего стоит хотя бы возможность выполнения присваивания в условном операторе в Си, например:

```
if (c = n) x = 0; /* в данном случае не проверяется равенство c и n, а
выполняется присваивание c значения n, после чего результат операции
сравнивается с нулем, если программист хотел выполнить не присваивание, а
сравнение, то эта ошибка будет обнаружена только на этапе выполнения при
получении результатов, отличающихся от ожидаемых */ Ошибки
компоновки. Ошибки компоновки, как следует из названия, 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. К самой непредсказуемой группе относятся ошибки выполнения. Прежде всего они могут иметь разную природу, и

соответственно по-разному проявляться. Часть ошибок обнаруживается и документируется операционной системой. Выделяют четыре способа проявления таких ошибок:

- появление сообщения об ошибке, зафиксированной схемами контроля выполнения машинных команд, например, переполнении разрядной сетки, ситуации «деление на ноль», нарушении адресации и т. п.;
- появление сообщения об ошибке, обнаруженной операционной системой, например, нарушении защиты памяти, попытке записи на устройства, защищенные от записи, отсутствии файла с заданным именем и т. п.;
- «зависание» компьютера, как простое, когда удастся завершить программу без перезагрузки операционной системы, так и «тяжелое», когда для продолжения работы необходима перезагрузка;
- несовпадение полученных результатов с ожидаемыми.

Примечание. Отметим, что, если ошибки этапа выполнения обнаруживает пользователь, то в двух первых случаях, получив соответствующее сообщение, пользователь в зависимости от своего характера, степени необходимости и опыта работы за компьютером, либо попытается понять, что произошло, ища свою вину, либо обратится за помощью, либо постарается никогда больше не иметь дела с этим продуктом. При «зависании» компьютера пользователь может даже не сразу понять, что происходит что-то не то, хотя его печальный опыт и заставляет волноваться каждый раз, когда компьютер не выдает быстрой реакции на введенную команду, что также целесообразно иметь в виду. Также опасны могут быть ситуации, при которых пользователь получает неправильные результаты и использует их в своей работе.

Причины ошибок выполнения очень разнообразны, а потому и локализация может оказаться крайне сложной. Все возможные причины ошибок можно разделить 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 (см. § 2.7) 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 последней группе 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 и т. п.;

ошибки вычислений, например, некорректные вычисления над неарифметическими переменными, некорректное использование целочисленной арифметики, некорректное преобразование типов данных в процессе вычислений, ошибки, связанные с незнанием приоритетов выполнения операций для арифметических и логических выражений, и т. п.;

ошибки межмодульного интерфейса, например, игнорирование системных соглашений, нарушение типов и последовательности при передаче параметров, несоблюдение единства единиц измерения формальных и фактических параметров, нарушение области действия локальных и глобальных переменных;

другие ошибки кодирования, например, неправильная реализация логики программы при кодировании, игнорирование особенностей или ограничений конкретного языка программирования.

Накопление погрешностей результатов числовых вычислений возникает, например, при некорректном отбрасывании дробных цифр чисел, некорректном использовании приближенных методов вычислений,

игнорировании ограничения разрядной сетки представления вещественных чисел в ЭВМ и т. п.

Все указанные выше причины возникновения ошибок следует иметь в виду в процессе отладки. Кроме того, сложность отладки увеличивается также вследствие влияния следующих факторов:

опосредованного проявления ошибок;

возможности взаимовлияния ошибок;

возможности получения внешне одинаковых проявлений разных ошибок;

отсутствия повторяемости проявлений некоторых ошибок от запуска к запуску – так называемые 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.

Методы отладки программного обеспечения

Отладка программы в любом случае предполагает обдумывание и логическое осмысление всей имеющейся информации об ошибке. Большинство ошибок можно обнаружить по косвенным признакам посредством тщательного анализа текстов программ и результатов тестирования без получения дополнительной информации. При этом используют различные методы:

ручного тестирования;

индукции;

дедукции;

обратного прослеживания.

Метод ручного тестирования. Это - самый простой и естественный способ данной группы. При обнаружении ошибки необходимо выполнить тестируемую программу вручную, используя тестовый набор, при работе с которым была обнаружена ошибка.

Метод очень эффективен, но не применим для больших программ, программ со сложными вычислениями и в тех случаях, когда ошибка связана с неверным представлением программиста о выполнении некоторых операций.

Данный метод часто используют как составную часть других методов отладки.

Метод индукции. Метод основан на тщательном анализе симптомов ошибки, которые могут проявляться как неверные результаты вычислений или как сообщение об ошибке. Если компьютер просто «зависает», то фрагмент проявления ошибки вычисляют, исходя из последних полученных результатов и действий пользователя. Полученную таким образом информацию организуют и тщательно изучают, просматривая соответствующий фрагмент программы. В результате этих действий выдвигают гипотезы об ошибках, каждую из которых проверяют. Если гипотеза верна, то детализируют информацию об ошибке, иначе - выдвигают другую гипотезу. Последовательность выполнения отладки методом индукции показана на рис. 10.3 в виде схемы алгоритма.

Самый ответственный этап - выявление симптомов ошибки. Организуя данные об ошибке, целесообразно записать все, что известно о ее проявлениях, причем фиксируют, как ситуации, в которых фрагмент с ошибкой выполняется нормально, так и ситуации, в которых ошибка проявляется. Если в результате изучения данных никаких гипотез не появляется, то необходима дополнительная информация об ошибке. Дополнительную информацию можно получить, например, в результате выполнения схожих тестов.

В процессе доказательства пытаются выяснить, все ли проявления ошибки объясняет данная гипотеза, если не все, то либо гипотеза не верна, либо ошибок несколько.

Метод дедукции. По методу дедукции вначале формируют множество причин, которые могли бы вызвать данное проявление ошибки. Затем анализируя причины, исключают те, которые противоречат имеющимся данным. Если все причины исключены, то следует выполнить дополнительное тестирование исследуемого фрагмента. В противном случае наиболее вероятную гипотезу пытаются доказать. Если гипотеза объясняет

полученные признаки ошибки, то ошибка найдена, иначе - проверяют следующую причину.

Метод обратного прослеживания. Для небольших программ эффективно применение метода обратного прослеживания. Начинают с точки вывода неправильного результата. Для этой точки строится гипотеза о значениях основных переменных, которые могли бы привести к получению имеющегося результата. Далее, исходя из этой гипотезы, делают предложения о значениях переменных в предыдущей точке. Процесс продолжают, пока не обнаружат причину ошибки.