

Практическая работа № 4

Тема Выявление первичных и вторичных ошибок

Цель: научиться выявлять первичные и вторичные ошибки программного обеспечения

Теоретические сведения

Под ошибкой в широком смысле слова понимается неправильность, погрешность или неумышленное, невольное искажение объекта или процесса. При этом подразумевается, что известно правильное или неискаженное эталонное состояние объекта, к которому относится ошибка.

Считается, что если программа не выполняет того, что пользователь от нее ожидает, то в ней имеется ошибка.

Важной особенностью процесса выявления ошибок в сложных программах является отсутствие полностью определенной правильной программы-эталона, которому должен соответствовать проверяемый текст.

Поэтому нельзя гарантированно утверждать, что возможно написать программу без ошибок.



Рисунок 1 – Распределение выявленных ошибок

Искажения в тексте программ (первичные ошибки) являются элементами, подлежащими корректировке. Однако непосредственно наличие ошибки обнаруживаются по ее вторичным проявлениям.

Искажение выходных результатов исполнения программ (вторичная ошибка) вызывает необходимость выполнения ряда операций по локализации устранению первичной ошибки (отладка программ).

На этапе отладки программ выявляются и исправляются много ошибок, но не все.

После отладки в течение некоторого времени интенсивность обнаружения ошибок при самом активном тестировании снижается настолько, что разработчики попадают в зону нечувствительности к программным ошибкам и отказам.

При такой интенсивности отказов программ трудно прогнозировать затраты времени, необходимые для обнаружения очередной ошибки.

Создается представление о полном отсутствии ошибок в программе, о невозможности и бесцельности их поиска. Поэтому усилия на отладку сокращаются, и интенсивность обнаружения ошибок еще больше снижается.

Этой предельно низкой интенсивности обнаружения отказов соответствует наработка на обнаруженную ошибку, при которой прекращается улучшение характеристик программного обеспечения на этапах его отладки и испытаний у заказчика.

Ошибку можно отнести к одному из ниже перечисленных классов:

- системные ошибки;
- ошибки в выборе алгоритма;
- алгоритмические ошибки;
- технологические ошибки;
- программные ошибки.

Системные ошибки в большом (сложном) программном обеспечении определяются, прежде всего неполной информацией о реальных процессах, происходящих в источниках и потребителях информации.

На начальных стадиях проектирования программного обеспечения не всегда удается точно сформулировать целевую задачу всей системы и требования к ней. В процессе проектирования целевая функция системы

уточняется и выявляются отклонения от уточненных требований, которые могут квалифицироваться как системные ошибки.

Некачественное определение требований к программе приводит к созданию программы, которая будет правильно решать неверно сформулированную задачу. В таких случаях, как правило, требуется полное перепрограммирование.

Признаком того, что создаваемая для заказчика программа может оказаться не соответствующей его истинным потребностям, служит ощущение неясности задачи. Письменная регистрация требований к программе заставляет заказчика собраться с мыслями и дать достаточно точное определение требований. Всякие устные указания являются заведомо ненадежными и часто приводят к взаимному недопониманию.

При автономной и в начале комплексной отладки программного обеспечения доля найденных системных ошибок в нем невелика (примерно 10%), но она существенно возрастает (до 35—40%) на завершающих этапах комплексной отладки.

В процессе эксплуатации преобладающими являются системные ошибки (примерно 80% всех ошибок).

Ошибки в выборе алгоритма. Часто плохой выбор алгоритма становится очевидным лишь после его опробования. Поэтому все же следует уделять внимание и время выбору алгоритма, с тем, чтобы впоследствии не приходилось переделывать каждую программу. Во избежание выбора некорректных алгоритмов, необходимо хорошо ознакомиться с литературой по своей специальности.

К алгоритмическим ошибкам следует отнести, прежде всего, ошибки, обусловленные некорректной постановкой функциональных задач, когда в спецификациях не полностью оговорены все условия, необходимые для получения правильного результата.

Эти условия формируются и уточняются в значительной части в процессе тестирования и выявления ошибок в результатах функционирования программ.

Также следует отнести ошибки связей модулей и функциональных групп программ. Их можно квалифицировать как ошибки некорректной постановки задачи.

Алгоритмические ошибки проявляются в неполном учете диапазонов изменения переменных, в неправильной оценке точности используемых и получаемых величин, в неправильном учете связи между различными переменными, в неадекватном представлении формализованных условий решения задачи в спецификациях или схемах, подлежащих программированию и т.д.

Эти обстоятельства являются причиной того, что для исправления каждой алгоритмической ошибки приходится изменять иногда целые ветви программного обеспечения, т.е. пока еще существенно больше операторов, чем при исправлении программных ошибок.

Алгоритмические ошибки значительно труднее поддаются обнаружению методами формализованного автоматического контроля. Вот почему необходимо тщательным образом продумывать алгоритм прежде, чем транслировать его в программу.

Некоторые программисты проверяют алгоритм следующим образом. Через несколько дней после составления алгоритма они повторно обращаются к описанию задачи и составляют алгоритм заново. Затем сличают оба варианта. Такой шаг на первый взгляд может показаться пустой тратой времени, однако всякая ошибка на уровне алгоритма может в дальнейшем обернуться катастрофой и повлечь основательный пересмотр программы.

Технологические ошибки— это ошибки документации и фиксирования программ в памяти ЭВМ. Они составляют 5—10 % от общего числа ошибок, обнаруживаемых при отладке. Большинство технологических ошибок

выявляются автоматически формализованными методами (например, транслятором).

Программные ошибки. Языки программирования - это искусственные языки, созданные человеком для описания алгоритмов. Все предложения таких языков строятся по строгим синтаксическим правилам, обеспечивающим однозначное их понимание, что позволяет поручать расшифровку алгоритма ЭВМ, построенного по правилам семантики.

Синтаксис - это набор правил построения из символов алфавита специальных конструкций, с помощью которых можно составлять различные алгоритмы (программы). Эти правила требуют их неукоснительного соблюдения. В противном случае будет нарушен основной принцип - четкая и строгая однозначность в понимании алгоритма.

Семантика языка — это система правил истолкования построений конструкций. Правила семантики конструкций обычно вполне естественны и понятны, но в некоторых случаях их надо специально оговаривать, комментировать.

Таким образом, программы, позволяющие однозначно производить процесс переработки данных, составляются с помощью соединения символов из алфавита в предложения в соответствии с синтаксическими правилами, определяющими язык, с учетом правил семантики.

Выделяют синтаксические и семантические ошибки.

Под **синтаксическими ошибками** понимается нарушение правил записи программ на данном языке программирования. Они выявляются самой машиной, точнее транслятором, вовремя перевода записи алгоритма на язык машины. Исправление их осуществляется просто — достаточно сравнить формат исправляемой конструкции с синтаксисом в справочнике и исправить его.

Семантические (смысловые) ошибки — это применение операторов, которые не дают нужного эффекта (например, a —вместо, $a+v$), ошибка в структуре алгоритма, в логической взаимосвязи его частей, в применении

алгоритма к тем данным, к которым он неприменим и т.д. Правила семантики не формализуемы. Поэтому поиск и устранение семантической ошибки и составляет основу отладки.

Каждая программная ошибка влечет за собой необходимость изменения команд существенно меньше, чем при алгоритмических и системных ошибках.

На этапах комплексной отладки программного обеспечения и эксплуатации удельный вес программных ошибок падает и составляет примерно 15 и 30 % соответственно от общего количества ошибок, выявляемых в единицу времени.

Методические указания

1. Заполните таблицу 4

Таблица 4

Категории тяжести ошибки в программном обеспечении

Номер категории ошибки	Наименование категории тяжести ошибки	Описание последствий проявления ошибки

2. Разделите ошибки на первичные и вторичные
3. Сделайте вывод по проделанной работе

Контрольные вопросы

1. Приведите классификацию ошибок программного обеспечения
2. Какие ошибки могут возникнуть при тестировании программного обеспечения?
3. Перечислите основные пути борьбы с ошибками.