

Надежность программных продуктов

Средства и методы повышения надежности

Принципы и методы обеспечения надежности программ, в соответствии с их целью, можно разделить на четыре группы:

- Предупреждение ошибок,
- Обнаружение ошибок
- Исправление ошибок
- Обеспечение устойчивости к ошибкам

К первой группе относятся принципы и методы, позволяющие минимизировать или вообще исключить ошибки.

Методы второй группы сосредоточивают внимание на функциях самого программного обеспечения, помогающих выявлять ошибки.

К третьей группе относятся функции программного обеспечения, предназначенные для исправления ошибок или их последствий.

Устойчивость к ошибкам (четвертая группа) – это мера способности системы программного обеспечения продолжать функционирование при наличии ошибок.



Схема обеспечения надежного ПС

К действиям, направленным на минимизацию ошибок и сбоев пп можно отнести:

1. предотвращение ошибок за счет структурного программирования;
2. сокрытие информации или дозированный доступ к данным со стороны программных средств и объектов в объектно-ориентированном программировании;
3. отладку;
4. устойчивость к сбоям;
5. обработку исключительных ситуаций (перехват ошибок, например, деление на ноль) и локализацию ошибок и сбоев;
6. восстановление программы после сбоя

Существует стандартное утверждение, что количество найденных ошибок в программе зависит от интенсивности использования программы. Причем интенсивность использования в свою очередь зависит от количества пользователей программы. Чем большее количество пользователей осуществляет работу с программным продуктом, тем быстрее будут обнаружены возможные программные ошибки. С течением времени процент найденных ошибок уменьшается, причем самое значительное уменьшение отмечается для программных продуктов с высокой степенью интенсивности использования.

В диаграмме, представленной ниже, рассмотрены статистические данные по частоте и видах наиболее часто регистрируемых ошибок программных продуктов.

Из диаграммы видно, что максимальный процент найденных ошибок связан с неполной или ошибочной спецификацией программного продукта.

При этом анализ данных ошибок выявил то, что в данную категорию ошибок максимальный вклад вносит ошибочность, неточность или неполнота исходных данных (52%). На втором месте находятся ошибки составления и оформления спецификации (15%), а на третьем двусмысленность описанных требований (13%).

Анализ статистики выявленных ошибок позволяет улучшить качество таких важных процессов разработки пп, как *валидация и верификации ПО*.

Следующим методом обеспечения надежности по является **метод обнаружения ошибок**, который базируется на стратегии включения средств обнаружения ошибок в само программное обеспечение. Большинство методов направлено по возможности на незамедлительное обнаружение сбоев.

Немедленное обнаружение имеет два преимущества:

- возможность минимизировать влияние ошибки
- возможность минимизировать затраты на поиск, анализ и исправление ошибки.

Меры по обнаружению ошибок можно разбить на две подгруппы:

1. Пассивные
2. Активные

1. *Пассивные* опираются на попытки обнаружить симптомы ошибки в процессе обычной работы программного обеспечения .

Меры по обнаружению ошибок могут быть приняты на нескольких структурных уровнях программной системы, при этом данные меры могут быть одинаково успешно применимы на любом уровне.

Стратегия этих мер заключается в совместном применении определенных приемов, каждый из которых способствует выявлению ошибок:

1. *Взаимное недоверие*. Каждый из компонентов должен предполагать, что все другие содержат ошибки. Когда он получает какие-нибудь данные от другого компонента или из источника вне системы, он должен предполагать, что данные могут быть неправильными, и пытаться найти в них ошибки.

2. *Немедленное обнаружение*. Ошибки необходимо обнаружить как можно раньше. Это не только ограничивает наносимый ими ущерб, но и значительно упрощает задачу отладки.

3. *Избыточность*. Все средства обнаружения ошибок основаны на некоторой форме избыточности (явной или неявной).

Меры по обнаружению ошибок, должны быть согласованы для всей системы. Предпринимаемые после обнаружения ошибки в ПО действия, должны быть единообразными для всех компонентов системы.

Среди действий, которые необходимо предпринять если ошибка обнаружена могут быть:

- немедленное завершение программы
- метод регистрации ошибок, когда описание симптомов ошибки и «моментальный снимок» состояния системы сохраняются во внешнем файле, после чего система может продолжать работу. Этот файл позднее будет изучен обслуживающим персоналом.

2. *Активное обнаружение ошибок.*

Не все ошибки можно выявить пассивными методами, поскольку эти методы обнаруживают ошибку лишь тогда, когда ее симптомы подвергаются соответствующей проверке. Существуют специальные программные средства для активного поиска признаков ошибок в системе. Такие средства называются средствами активного обнаружения ошибок.

Активные средства обнаружения ошибок обычно объединяются в диагностический монитор: параллельный процесс, который периодически анализирует состояние системы с целью обнаружить ошибку.

Диагностический монитор можно реализовать как периодически выполняемую задачу (например, она планируется на каждый час) либо как задачу с низким приоритетом, которая планируется для выполнения в то время, когда система переходит в состояние ожидания.

Выполняемые монитором конкретные проверки зависят от специфики системы. Например, обследование основной памяти, чтобы обнаружить блоки памяти, не выделенные ни одной из выполняемых задач и не включенные в системный список свободной памяти. Проверка необычных ситуаций: например, процесс не планировался для выполнения в течение некоторого разумного интервала времени. Монитор может осуществлять поиск «затерявшихся» внутри системы сообщений или операций ввода-вывода, которые необычно долгое время остаются незавершенными, участков памяти на диске, которые не помечены как выделенные и не включены в список свободной памяти, а также различного рода странностей в файлах данных и т.п.

Также в определенных обстоятельствах монитор может выполнять диагностические тесты системы. Он может вызывать определенные системные функции, сравнивая их результат с заранее определенным и проверяя, насколько разумно время выполнения. Монитор может также периодически предъявлять системе «пустые» или «легкие» задания, чтобы убедиться, что система функционирует.