

Тема 10. Тестирование и отладка программного средства

Часть 2. Отладка ПС

Виды отладки:

- **Автономная отладка** (последовательное раздельное тестирование различных частей программ, входящих в состав ПС с поиском и исправлением фиксируемых при тестировании ошибок).
- **Комплексная отладка** (тестирование ПС с поиском и исправлением ошибок во всех документах, включая тексты программ ПС, требования к ПС, спецификация качества ПС, описание архитектуры).

(См. рисунок)

Виды отладки программного средства

Автономная отладка

Комплексная отладка

Отладка каждого программного модуля
и его окружения (сопряженных модулей)

Тестирование
ПС в целом

Один отладочный
модуль – головной
в тестируемой программе

Множество отладочных
модулей – отладочных
имитаторов (заглушек)

Восходящее тестирование

Нисходящее тестирование

Виды отладки программного средства

Автономная отладка

Комплексная отладка

Отладка каждого программного модуля и его окружения (сопряженных модулей)

Тестирование ПС в целом

Один отладочный модуль – головной в тестируемой программе

Множество отладочных модулей – отладочных имитаторов (заглушек)

Восходящее тестирование

Нисходящее тестирование

Отладочный модуль готовит состояние информационной среды необходимое для тестирования отлаживаемого модуля, осуществляет обращение к отлаживаемому модулю и после окончания его работы выдает необходимые сообщения

См. следующий рисунок

Отладочные модули (драйверы) для отлаживаемых кластеров

Модули нижнего уровня объединяются в кластеры, отлаживаемые в первую очередь

Модули объединяются в кластеры по функциям, которые они выполняют, для возможности протестировать их с помощью одного набора тестов

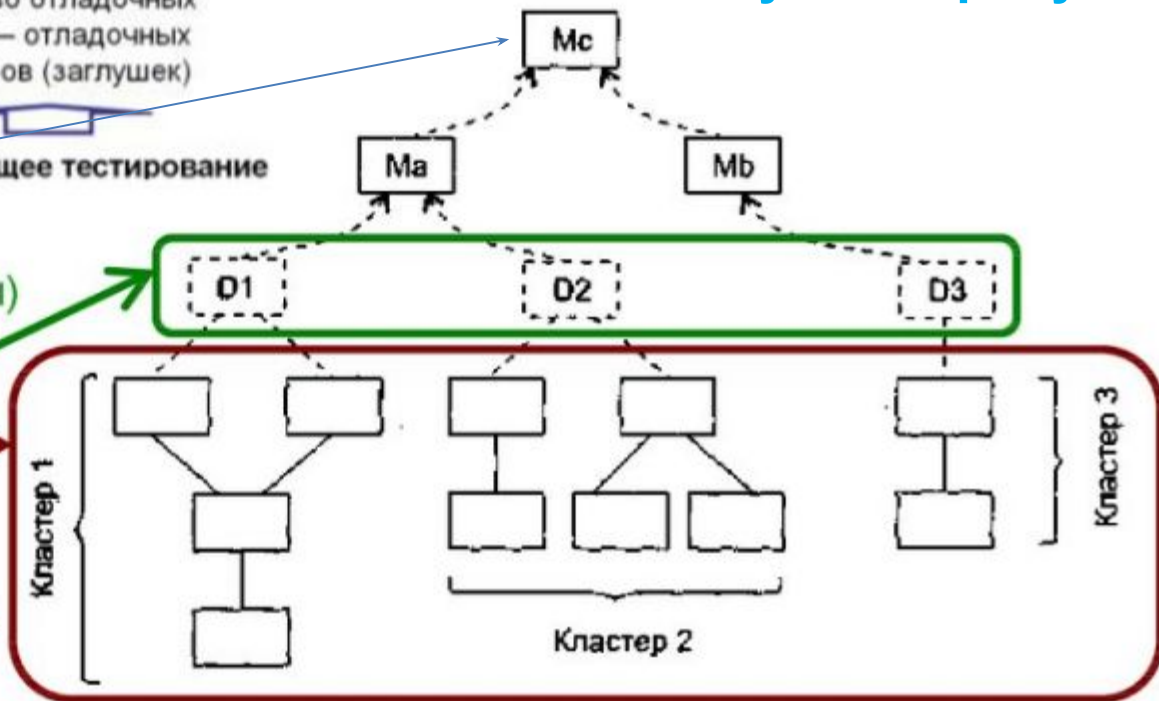
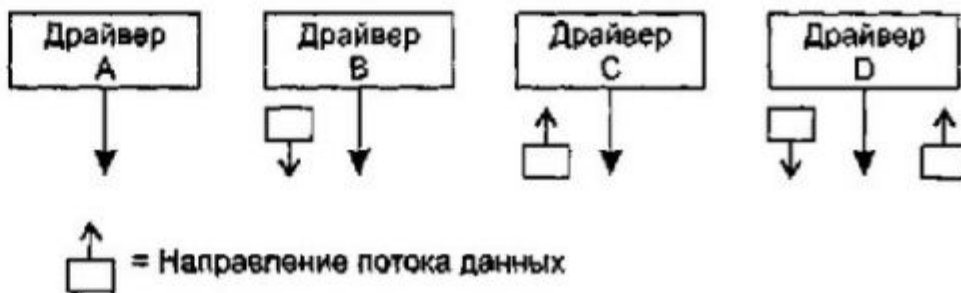


Рис. 8.6. Восходящая интеграция системы



Информационная среда отлаживаемых модулей

Отладочные модули (драйверы) для отлаживаемых кластеров

Модули нижнего уровня объединяются в кластеры, отлаживаемые в первую очередь

Модули объединяются в кластеры по функциям, которые они выполняют, для возможности протестировать их с помощью одного набора тестов

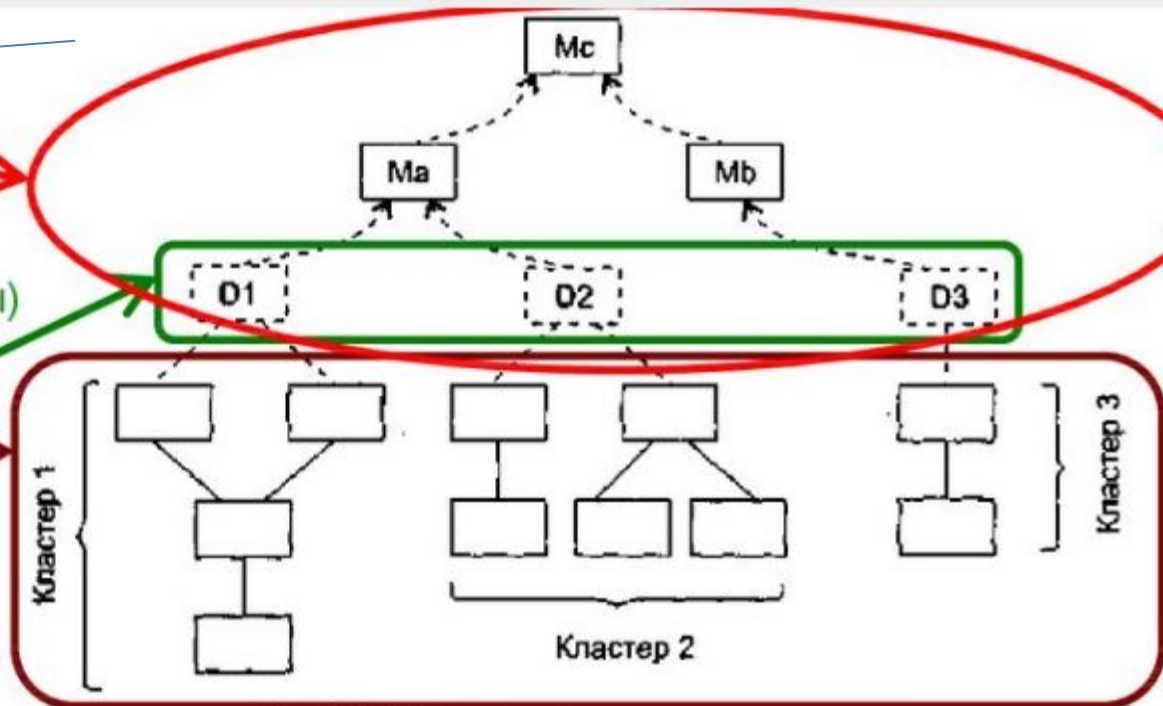


Рис. 8.6. Восходящая интеграция системы



Отладочный модуль готовит состояние информационной среды необходимое для тестирования отлаживаемого модуля, осуществляет обращение к отлаживаемому модулю и после окончания его работы выдает необходимые сообщения

Очередность
тестирования
модулей

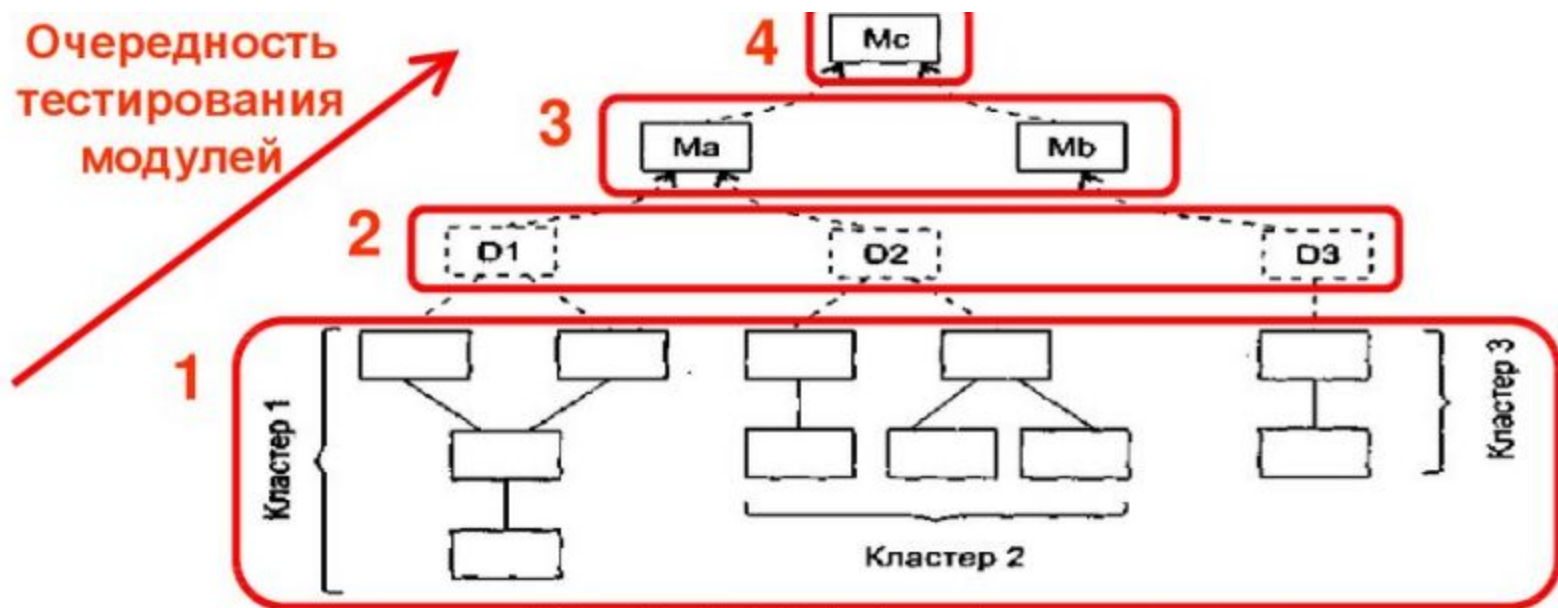
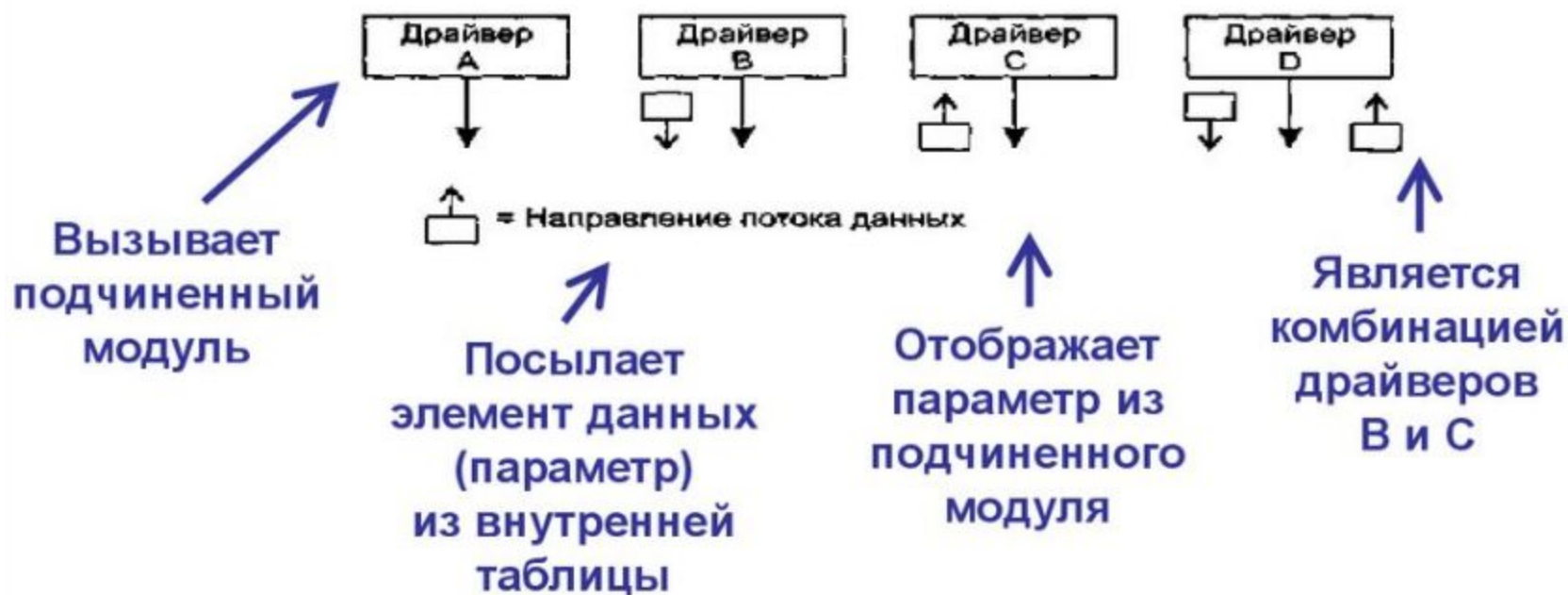


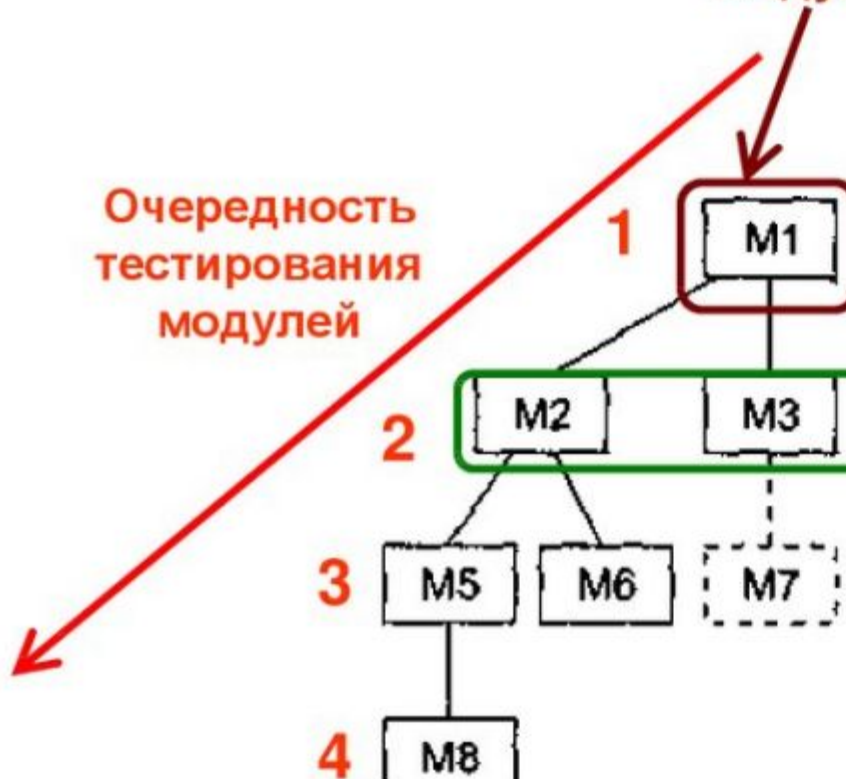
Рис. 8.6. Восходящая интеграция системы



Отлаживаемый модуль



Очередность тестирования модулей



Один отладочный модуль – головной в тестируемой программе

Всегда тестирование



Отладочные имитаторы (заглушки) – имитируют работу неотлаженных модулей

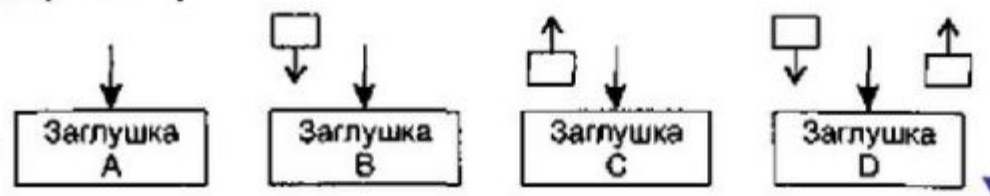
Отображает трассируемое сообщение

↑ = Направление потока данных

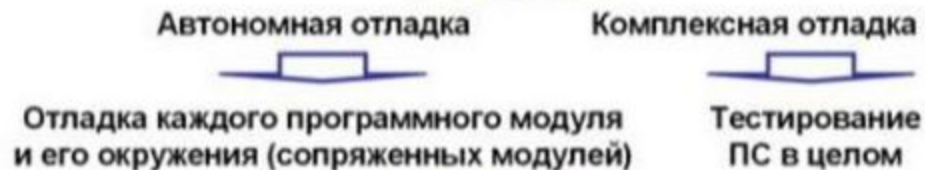
Отображает проходящий параметр

Возвращает величину из таблицы

Выполняет табличный поиск по ключу (входному параметру) и возвращает связанный с ним выходной параметр



Виды отладки программного средства



На практике часто применяется смешанная форма отладки ПС

Один отладочный модуль – головной в тестируемой программе

Восходящее тестирование

Достоинства:

- простота подготовки тестов;
- можно составить полный план тестирования перед его началом.

Недостатками:

- тестовые данные готовятся, как правило, в форме, непонятной пользователю;
- большой объем отладочного программирования;
- необходимость специального тестирования сопряжения модулей.

Множество отладочных модулей – отладочных имитаторов (заглушек)

Нисходящее тестирование

Достоинства:

- большинство тестов готовится в форме, рассчитанной на пользователя;
- во многих случаях относительно небольшой объем отладочного программирования;
- отпадает необходимость тестирования сопряжения модулей.

Недостатки:

- невозможность или сильное затруднение подготовки полного плана перед началом тестирования.

Автономная отладка ПС

При автономной отладке ПС каждый модуль тестируется в некотором программном окружении, состоящем из других модулей, часть которых является модулями отлаживаемой программы, которые уже отлажены, а часть - модулями, управляющими отладкой (*отладочными модулями*).

Таким образом, при автономной отладке тестируется всегда некоторая программа (*тестируемая программа*), построенная специально для тестирования отлаживаемого модуля. Эта программа лишь частично совпадает с отлаживаемой программой, кроме случая, когда отлаживается последний модуль отлаживаемой программы.

В процессе автономной отладки ПС производится наращивание тестируемой программы отлаженными модулями: при переходе к отладке следующего модуля в его программное окружение добавляется последний отлаженный модуль. Такой процесс наращивания программного окружения отлаженными модулями называется *интеграцией* программы.

При **восходящем тестировании** (см. лекцию 7) это окружение будет содержать **только один отладочный модуль** (кроме случая, когда отлаживается последний модуль отлаживаемой программы), который будет головным в тестируемой программе.

Такой отладочный модуль называют **ведущим** (или драйвером).

Ведущий отладочный модуль подготавливает информационную среду для тестирования отлаживаемого модуля, осуществляет обращение к отлаживаемому модулю и после окончания его работы выдает необходимые сообщения.

При отладке одного модуля для разных тестов могут составляться разные ведущие отладочные модули.

При **нисходящем тестировании** (см. лекцию 7) окружение отлаживаемого модуля в качестве отладочных модулей содержит *отладочные имитаторы* (заглушки) некоторых еще не отлаженных модулей.

К таким модулям относятся, прежде всего, все модули, к которым может обращаться отлаживаемый модуль, а также еще не отлаженные модули, к которым могут обращаться уже отлаженные модули (включенные в это окружение). Некоторые из этих имитаторов при отладке одного модуля могут изменяться для разных тестов.

Этапы автономного тестирования

– **Шаг 1 – На основании спецификации отлаживаемого модуля, подготовка тестов для каждой:**

- а) возможной ситуации,
- б) границы областей допустимых значений всех входных данных,
- в) областей изменения данных, недопустимых значений всех входных данных и каждого недопустимого условия.

Тестирование
“черного ящика”

– **Шаг 2 – Проверка текста модуля, с целью убедиться, что каждое направление любого разветвления будет пройдено хотя бы на одном тесте. Добавление недостающих тестов.**

– **Шаг 3 – Проверка текста модуля, с целью убедиться, что для каждого цикла – повторения, существуют тесты, обеспечивающие, по крайней мере, три следующие ситуации:**

- а) тело цикла не выполняется ни разу;
- б) тело цикла выполняется один раз
- в) тело цикла выполняется максимальное число раз.

Добавление недостающих тестов.

Тестирование
“белого ящика”

– **Шаг 4 – Проверка текста модуля, с целью убедиться, что существуют тесты, проверяющие чувствительность к отдельным особым значениям входных данных.**

Добавление недостающих тестов.

Тестирование
“черного ящика”

Комплексная отладка ПС

При комплексной отладке **тестируется ПС в целом**, причем тесты готовятся по каждому из документов в порядке обратном их разработке.

Исключение – тестирование документации по применению, которое разрабатывается по внешнему описанию параллельно с разработкой текстов программ.

Виды отладки программного средства



- 1 Поиск несоответствий между описанием архитектуры, структурой и взаимодействием программных модулей и подсистем ПС.
- 2 Поиск расхождений между функциональной спецификацией и функциями программных модулей и подсистем ПС
- 3 Поиск нарушений требований качества, сформулированных в спецификации качества ПС
- 4 Поиск несоответствий документации заданным требованиям, ее несогласованности с содержанием и функциями ПО
- 5 Поиск несоответствий ПО установленным требованиям самим заказчиком

Заповеди отладки программного средства

Заповедь 1. Считайте тестирование ключевой задачей разработки ПС, поручайте его самым квалифицированным и одаренным программистам; нежелательно тестировать свою собственную программу.

Заповедь 2. Хорош тот тест, для которого высока вероятность обнаружить ошибку, а не тот, который демонстрирует правильную работу программы.

Заповедь 3. Готовьте тесты как для правильных, так и для неправильных данных.

Заповедь 4. Документируйте пропуск тестов через компьютер; детально изучайте результаты каждого теста; избегайте тестов, пропуск которых нельзя повторить.

Заповедь 5. Каждый модуль подключайте к программе только один раз; никогда не изменяйте программу, чтобы облегчить ее тестирование.

Заповедь 6. Пропускайте заново все тесты, связанные с проверкой работы какой-либо программы ПС или ее взаимодействия с другими программами, если в нее были внесены изменения (например, в результате устранения ошибки).

СРЕДСТВА АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ

HP QTP

AutomatedQA TestComplete

Rational Functional Tester

Borland SilkTest

Microsoft VS

Selenium

Watir

Canoo WebTest

Jemmy

TestPartner

Worksoft Certify

и др.

Подготовить презентацию по средствам автоматизации тестирования

<https://habr.com/ru/post/481294/>

Ссылка для примера!

Почитать

<https://habr.com/ru/company/otus/blog/443418/>

<http://juice-health.ru/program/software-testing/495-software-testing-methods>