

**Лабораторная работа №10 по курсу «Архитектура ЭВМ, системное программное обеспечение» на 2011/12 учебный год.  
Отладчик системы программирования ОС UNIX**

*«Использование отладчика — последнее средство, перед тем, как бросить системное программирование и заняться чем-нибудь полегче, например, системным анализом!»*

Э. Йодан

В ходе выполнения работы необходимо проделать и запротолировать как минимум **все** описанные действия над некоторой программой на изучаемом языке программирования, в том числе и со специально внесёнными ошибками времени выполнения. Входной контроль знаний по работе проводится по вопросам к зачёту №4 [1].

Отладчик `gdb` [4] представляет собой интерактивное средство отладки программ, разрабатываемых в системе программирования GNU Compiler Collection. Отладка может производиться *путем пошагового интерпретативного выполнения* специальным образом скомпилированной программы, позволяющего осуществлять трассировку работы программы с контролем и изменением значений переменных, установку контрольных точек останова *либо с помощью обработки посмертного дампа программы (core)*. Строчный интерфейс отладчика позволяет вести отладку с помощью текстового терминала, а также облегчает удалённую отладку.

Отладка программы производится в интерактивном режиме путем ввода команд с терминала. Запуск отладчика производится из командной строки командой `gdb` с указанием имени отлаживаемой программы (выполнимого файла в машинных кодах) и, возможно, файла с посмертным дампом `core`. Отлаживаемая программа перед отладкой должна быть обязательно **скомпилирована с ключом компилятора `-g`**.

Основным назначением интерактивного отладчика является поиск динамических ошибок в программе, таких, которые не удаётся найти без её выполнения [2]. Существует два различных способа использования отладчика:

1. *Анализ дампа аварийного завершения (core)*. В случае фатальной ошибки во время выполнения программа аварийно завершается, а содержимое её области оперативной памяти дампируется (сохраняется) в файл `core`. Для отладки такой программы нужно применять соответствующий отладчик к файлам `core` и `a.out` `gdb -c a.out.core a.out` (в зависимости от используемой системы программирования, `a.out` – имя исполняемой программы). После запуска отладчик попадает в тот блок программы, в котором произошла ошибка, и позволяет просмотреть значения переменных, стек вызова процедур/функций, а также произвести откат с тем, чтобы *начерно* пройти место аварии со скорректированными значениями переменных.
2. *Динамическая отладка (без core)* представляет собой пошаговое выполнение программы на тестовых данных. Если автору программы известен набор входных данных, приводящий к ошибке, то отладчик применяется только к оригинальной выполняемой программе: `gdb a.out`. Для пошагового выполнения программы можно установить точку останова на первой строке основной программы, или несколькими строками выше места предполагаемой ошибки (команда `break <номер-строки>`). Далее необходимо запустить программу командой `run`, и после остановки на заданной строке программы использовать команды `next/step` для пошагового выполнения отлаживаемой программы. Для вывода текста программы на экран (например, чтобы определить номер первой выполняемой строки основной программы) служит команда `list`. В процессе пошаговой интерпретации можно отображать текущие значения скалярных и структурных переменных командой `print`.

**Основные команды отладчика (подчёркнуты минимальные сокращения):**

| <b><u>gdb</u></b>  | <b>Описание</b>  |
|--|--|
| <b><u>help</u></b> [ <i>&lt;раздел&gt;</i> ]   | подсказка по разделу помощи отладчика. <b>help</b> без параметров выводит список разделов  |
| <b><u>list</u></b> [ <i>&lt;имя функции/процедуры/файла&gt;:</i> ] [ <i>номер строки</i> ] | распечатка текста функции/процедуры/файла или всей программы, начиная с указанной строки. По умолчанию распечатываются следующие 10 строк программы. Распечатываемый файл становится текущим файлом исходного текста отлаживаемой программы! |
| <b><u>break</u></b> <i>&lt;номер строки/имя функции&gt;</i>                                | задание точки останова на указанной строке/функции текущего исходного файла программы  |
| <b><u>run</u></b> [ <i>&lt;параметры&gt;</i> ]   | запуск программы на выполнение. Могут указываться необязательные параметры командной строки и операции перенаправления ввода-вывода. <b>gdb</b> запоминает параметры и подставляет их при дальнейших вызовах <b>run</b>                      |
| <b><u>set args</u></b> [ <i>&lt;параметры&gt;</i> ]  | предварительная установка параметров командной строки  |
| <b><u>print</u></b> <i>&lt;выражение&gt;</i>   | печать значения выражения, которое может включать и переменные, и вызовы функций программы   |
| <b><u>next</u></b> [ <i>&lt;n&gt;</i> ]  | выполнение очередной строки программы при пошаговой трассировке (процедуры и функции не трассируются, а выполняются за один такт). Необязательный параметр <i>n</i> указывает число строк программы для выполнения (по умолчанию — 1)        |
| <b><u>step</u></b> [ <i>&lt;n&gt;</i> ]  | выполнение очередной строки программы (с трассировкой вызовов функций/процедур). Перед выполнением <i>next/step</i> программа должна быть запущена командой <b>run</b>   |
| <b><u>set var</u></b> <i>&lt;имя&gt;</i> = <i>&lt;выражение&gt;</i>                        | присваивание значения переменной   |
| <b><u>ptype</u></b> <i>&lt;имя-переменной&gt;</i>  | распечатка определения типа переменной (на языке программирования).  |
| <b><u>backtrace</u></b> или <b><u>bt</u></b>   | распечатка содержимого стека вызовов   |
| <b><u>continue</u></b>   | продолжение выполнения программы после остановки   |
| <b><u>quit</u></b>   | выход из отладчика   |

Нажатие клавиши [Enter] в пустой командной строке отладчика вызывает повторение предыдущей команды отладчика. Обычно используется для многократного выполнения команд **next/step**.

Следует избегать команд **next/step**, ведущих в языковую среду, т.к. путешествие по кодам библиотек компилятора, например функции `printf`, бесполезно для начинающего программиста. Соответствующие исходные тексты системных программ могут быть недоступны в момент отладки, либо они могут быть реализованы на другом языке программирования. Кроме того, такие программы, как правило, компилируются без ключа `-g`.

Более подробные сведения по отладчику `gdb` версии 7.0 приведены в полном документе [4], во встроенной документации отладчика и могут быть получены по команде **man gdb**. Русский перевод документа по 5-й версии `gdb` смотрите в [3] ([www.mitya.pp.ru/gdb](http://www.mitya.pp.ru/gdb)), 264 стр. в формате pdf).

*Литература*

1. Жоголев Е.А. Лекции по технологии программирования. Лекция 10. Тестирование и отладка программного средства. [19]
2. Бек Л. Введение в системное программирование. [9], с. 423-429. В приложении – классическая статья про отладчики вообще.
3. Столмен Р. и др. Отладка с помощью **gdb**. Восьмая редакция, для **gdb** версии 5.0 (Март 2000) / Пер. с англ. Д. Сиваченко ([mitya@cavia.pp.ru](mailto:mitya@cavia.pp.ru)) под ред. О. Тихонова.
4. Debugging with gdb. <http://www.gnu.org/software/gdb/>.

Составители: проф. Зайцев В.Е., ст. преп. Лебедев А.В., ст.преп. Сеницкий П.А., доц. Сошников Д.В., асс. Перетягин И.А., прогн. Измайлов А.А. и Миронов Е.С.