Использование отладчика gdb

А. Г. Фенстер, http://info.fenster.name

6 февраля 2009 г.

Отладчиком называется программа, позволяющая выполнять вашу программу построчно и при этом контролировать значения переменных и состояние памяти. Я считаю, что при написании учебных программ вполне можно обойтись и без такого специального софта: самый полезный способ отладки — вывод текстовых сообщений и значений переменных на экран при помощи функции `printf`, а отладчик — это вещь, которая доступна не всегда и не везде. Однако, по просьбам студентов я кратко опишу работу в установленном на нашем сервере отладчике `gdb` (GNU Debugger).

Подготовка программы к отладке

Отладчик при работе использует как бинарный код (исполняемый файл), так и исходный код программы. Необходимо, чтобы бинарный код содержал информацию о номерах соответствующих строк кода на C и об именах переменных, используемых в программе — так называемую *отладочную информацию*. Для включения этой информации в исполняемый файл нужно добавить ключ `-g` в строку компиляции: если компилируется файл `file.c`, команда будет выглядеть так:

```
make file CFLAGS=-g
```

либо так:

```
cc -o file -g file.c
```

Обратите внимание, что в вызове `make` ключ `-g` устанавливается в переменную `CFLAGS` (ключи, передаваемые компилятору C), а не `LDFLAGS` (ключи,

передаваемые линковщику). Для программы, состоящей из одного файла, это не имеет значения, но важно для больших программ.

После такой компиляции исполняемый файл `file` будет содержать необходимую для отладчика информацию.

Запуск отладчика

Для начала отладки нужно запустить программу `gdb`, указав ей параметром **бинарный** файл — программу, которую отлаживать:

```
gdb file
```

На экране появятся текст (информация об отладчике) и приглашение:

```
(gdb)
```

Сама ваша программа в настоящий момент не запущена. Её можно запустить командой `run`. Если программа завершится успешно, появится сообщение `Program exited normally.` В противном случае (скажем, если произошла ошибка сегментирования `Segmentation fault`) будет показано, какая именно строчка вызвала ошибку:

```
Program received signal SIGSEGV, Segmentation fault.
0x08048350 in f (i=1023) at file.c:7
7               printf("%d\n", A[i]);
```

Можно также узнать последовательность вызовов функций, которая привела к данному состоянию («стек вызовов»). Для этого используется команда `where`:

```
(gdb) where
#0  0x08048350 in f (i=1023) at file.c:7
#1  0x08048390 in main () at file.c:13
```

Наконец, можно вывести значения любой из активных в текущий момент переменных при помощи команды `print` (или просто `p`). Например, для вывода значения переменной `i` напишите:

```
(gdb) p i
$1 = 1023
```

Если необходимо указать при запуске программы параметры командной строки, это можно сделать командой `set args`.

Пошаговый прогон программы

Одной из наиболее полезных возможностей, предоставляемых отладчиком, является пошаговый прогон программы, т. е. выполнение инструкции за инструкцией в интерактивном режиме, по команде пользователя. В `gdb` есть несколько команд, используемых для того, чтобы сделать такой прогон. Во-первых, необходимо установить *точку остановки* — позицию в коде программы, в которой будет приостановлено выполнение. Точка остановки может быть связана с номером строки в файле с исходным кодом или с некоторой функцией. Для установки такой точки используется команда `break`. Вот так можно заставить отладчик остановить программу при входе в функцию `f`:

```
(gdb) break f
Breakpoint 1 at 0x804837a: file file.c, line 7.
```

Следующая команда установит точку остановки на 9-ю строку кода:

```
(gdb) break 9
Breakpoint 2 at 0x8048396: file file.c, line 9.
```

Для удаления точки остановки можно использовать команды `clear` (с теми же аргументами, что и `break`) и `delete` (указав ей номер точки остановки).

Если в программе указаны точки остановки, выполнение программы остановится при достижении любой из них, и на экране будет напечатана следующая строка кода, которая будет исполнена. Далее можно применить, например, одну из следующих команд:

- `step` или `s` — выполнение одной строки (той, что напечатана на экране). Если в этой строке есть вызов процедуры или функции, она также будет проходиться в пошаговом режиме.

- `next` или `n` — то же самое, но вызовы процедур и функций будут проходиться сразу, за один шаг.

- `continue` или `c` — продолжить выполнение программы (закончить пошаговую отладку). Выполнение остановится в случае завершения программы или достижения очередной точки остановки.

Если вы не вводите никакой команды, а просто нажимаете `Enter`, будет повторно выполнена предыдущая команда.

Также в моменты остановки выполнения можно использовать команду `print` (`p`) для вывода значений переменных в данный момент.

Прочие команды

Для выхода из `gdb` нажмите `Ctrl+D` в командной строке или введите команду `quit`. Для того, чтобы прочитать справку по командам, введите команду `help`.