

[Каталог документации](#) / [Раздел "Программирование, языки"](#) / [Оглавление документа](#)

Go to the [first](#), [previous](#), [next](#), [last](#) section, [table of contents](#).

Вход и выход из GDB

Эта глава посвящена тому, как запустить GDB и как из него выйти. Основные принципы:

- введите `'gdb'` для вызова GDB.
- введите `quit` или `C-d` для выхода из него.

Вызов GDB

Вызывайте GDB путем запуска программы `gdb`. Начав работу, GDB считывает команды с терминала до тех пор, пока вы не скажете ему выйти.

Вы также можете запустить `gdb` с различными аргументами и ключами, чтобы в самом начале лучше настроить среду отлаживания.

Ключи командной строки, описанные здесь, предназначены для охвата различных ситуаций; в действительности, в некоторых средах часть этих ключей может быть недоступна.

Чаще всего GDB вызывается с одним аргументом, который определяет исполняемую программу:

```
gdb программа
```

Вы также можете указать при старте как исполняемую программу, так и файл дампа памяти:

```
gdb программа дамп
```

Если вы хотите отладить выполняющийся в данный момент процесс, то вместо этого, вы можете указать вторым аргументом идентификатор этого процесса:

```
gdb программа 1234
```

присоединит GDB к процессу 1234 (если, конечно, у вас нет файла с именем `'1234'`, GDB сначала проверяет наличие файла дампа памяти).

Преимущества, которые можно получить при использовании второго аргумента командной строки, требуют наличия достаточно совершенной операционной системы; если вы используете GDB как удаленный отладчик, присоединенный к компьютеру без операционной системы, там вообще может не быть понятия "процесса", и часто нет никакого способа получить дамп. GDB предупредит вас, если ему не удастся присоединиться к процессу или считать файл дампа памяти.

Вы можете запустить `gdb` без вывода начального сообщения, описывающего отсутствие гарантии на него, указав `-silent`:

```
gdb -silent
```

Кроме того, вы можете контролировать процесс запуска GDB с помощью ключей командной строки. GDB может сам напомнить вам о доступных ключах.

Введите

```
gdb -help
```

чтобы вывести на экран все доступные опции с кратким описанием их использования (сокращенный эквивалент---`gdb -h').

Все заданные вами ключи и параметры командной строки обрабатываются последовательно. Порядок становится важным при использовании ключа `-x'.

Выбор файлов

При запуске, GDB считывает параметры, отличные от ключей, как указатели на исполняемую программу и файл дампа (или идентификатор процесса), точно так же, как если бы эти параметры задавались ключами `-se' и `-с' соответственно. (GDB считает первый параметр, не имеющий соответствующего флага ключа, эквивалентом ключа `-se', за которым следует этот параметр; а второй параметр, не имеющий соответствующего флага ключа, если он есть, эквивалентом ключа `-с', за которым следует этот параметр.)

Если GDB был сконфигурирован без включения поддержки файлов дампа, что имеет место для большинства встроенных целей, то он выразит недовольство вторым аргументом и проигнорирует его.

Многие ключи имеют как длинную, так и краткую формы; в следующем списке приводятся обе. GDB также распознает сокращения длинных форм, не являющиеся двусмысленными. (Вы можете, по желанию, обозначать ключи с помощью `--', а не `-', хотя мы показываем наиболее употребляемый формат.)

`-symbols файл`

`-s файл`

Читать таблицу символов из файла *файл*.

`-exes файл`

`-e файл`

Использовать *файл* как исполняемый для выполнения и исследования данных вместе с дампом памяти, когда это необходимо.

`-se файл`

Читать таблицу символов из файла *файл* и использовать его как исполняемый файл.

`-core файл`

`-с файл`

Использовать *файл* как дамп памяти для исследования.

`-с номер`

Присоединиться к процессу с идентификатором *номер*, также, как по команде `attach` (при условии, что нет файла в формате дампа памяти с именем *номер*; в этом случае `-с' определяет этот файл как дамп для считывания).

`-command файл`

`-x файл`

Выполнить команды GDB из файла *файл*. See section [Командные файлы](#).

`-directory каталог`

`-d каталог`

Добавить *каталог* к путям поиска файлов с исходными текстами.

`-m`

`-mapped`

Предупреждение: этот ключ зависит от возможностей операционной системы, которые реализованы не везде.

Если отображаемые в память файлы поддерживаются в вашей системе через системный вызов `mmap`, вы можете использовать этот ключ, чтобы GDB записывал символы из вашей программы в файл в текущем каталоге, допускающий повторное использование. Если программа, которую вы отлаживаете, называется `'/tmp/fred'`, то отображаемым символьным файлом будет `'/tmp/fred.syms'`. Последующие отладочные сеансы GDB замечают наличие этого файла и могут быстро отобразить в память символьную информацию из него, а не читать таблицу символов из выполняемого файла. Файл `'.syms'` специфичен для рабочей машины, на которой запускается GDB. Он содержит точный образ внутренней символьной таблицы GDB. Он не может быть разделен между несколькими рабочими платформами.

`-r`

`-readnow`

Читать символьную таблицу каждого файла, содержащего таблицу символов, сразу целиком, а не стандартным образом, при котором она считывается постепенно по мере необходимости. Эта команда замедляет запуск, но дальнейшие операции производятся быстрее.

Ключи `-mapped` и `-readnow` обычно используются вместе, чтобы построить файл `'.syms'`, который содержит полную информацию о символах. (See section [Команды для задания файлов](#), для информации о файлах `'.syms'`.) Вот простой вызов GDB, не делающий ничего, кроме построения файла `'.syms'` для использования в будущем:

```
gdb -batch -nx -mapped -readnow имя-программы
```

Выбор режимов

Вы можете вызывать GDB в различных альтернативных режимах---например, в пакетном или в "тихом" режиме.

`-nx`

`-n`

Не выполнять команды ни из каких файлов инициализации (обычно называемых `'.gdbinit'`, или `'gdb.ini'` на PC). В нормальном режиме, GDB выполняет команды из этих файлов после обработки всех командных ключей и параметров. See section [Командные файлы](#).

`-quiet`

`-silent`

`-q`

"Тихий". Не печатать вводное сообщение и информацию об авторских правах. Эти сообщения также подавляются в пакетном режиме.

`-batch`

Выполняться в пакетном режиме. Выйти со значением 0 после обработки всех командных файлов, заданных ключем `'-x'` (и всех команд из инициализационных файлов, если это не запрещено ключем `'-n'`). Выйти с ненулевым значением, если во время выполнения команд GDB из командных файлов произойдет ошибка. Пакетный режим может быть полезен при вызове GDB как фильтра; например, чтобы загрузить программу и запустить ее на другом компьютере. Для того, чтобы сделать это более удобным, сообщение

Program exited normally.

(которое обычно выдается при завершении программы, выполняемой под управлением GDB), при выполнении в пакетном режиме не выдается.

-nowindows

-nw

"Без окон". Если GDB имеет встроенный графический интерфейс пользователя (GUI), то этот ключ велит GDB использовать только интерфейс командной строки. Если GUI недоступен, этот ключ не оказывает никакого действия.

-windows

-w

Если GDB включает GUI, этот ключ требует использовать его, если только возможно.

-cd *каталог*

Запустить GDB, используя в качестве рабочего каталога *каталог*, вместо текущего.

-fullname

-f

GNU Emacs устанавливает этот ключ, когда вызывает GDB как подпроцесс. Это велит GDB выводить полное имя файла и номер строки в стандартном, распознаваемом стиле всякий раз, когда отображается кадр стека (что включает каждую остановку вашей программы). Этот распознаваемый формат выглядит как два знака '\032', за которыми следует имя файла, номер строки и символьная позиция, разделенные двоеточиями, и знак новой строки. Программа интерфейса Emacs-GDB использует два знака '\032' как сигнал для отображения исходного текста для кадра.

-epoch

Интерфейс Emacs-GDB Epoch устанавливает этот ключ, когда вызывает GDB как подпроцесс. Это велит GDB изменить свои подпрограммы печати так, чтобы позволить Epoch отображать значения выражений в отдельном окне.

-annotate *уровень*

Этот ключ устанавливает *уровень примечаний* внутри GDB. Его эффект аналогичен использованию 'set annotate *уровень*' (see section [Примечания GDB](#)). Уровень примечаний контролирует, какое количество информации GDB выводит вместе с приглашением, значениями выражений, строками исходного текста и другими типами вывода. Уровень 0 является обычным, уровень 1 используется, когда GDB выполняется как подпроцесс GNU Emacs, уровень 2 выводит максимальное количество примечаний и подходит для программ, которые управляют GDB.

-async

Использовать асинхронный цикл событий для интерфейса командной строки. GDB обрабатывает все события, такие как ввод пользователя с клавиатуры, через специальный цикл событий. Это позволяет GDB принимать и обрабатывать команды пользователя параллельно с выполнением отлаживаемого процесса(4), так что вы не должны ждать возвращения управления GDB, прежде чем ввести следующую команду. (*Замечание:* в версии 5.0, асинхронное выполнение на целевой системе еще не поддерживается, так что режим '-async' еще реализован не полностью.) Когда стандартный ввод соединен с терминальным устройством, GDB по умолчанию использует асинхронный цикл событий, если это не отключено ключом '-noasync'.

-noasync

Отключить асинхронный цикл событий для интерфейса командной строки.

-baud *бод-в-сек*

-b *бод-в-сек*

Устанавливает скорость линии (скорость в бодах, или в битах в секунду) любого последовательного интерфейса, используемого GDB для удаленной отладки.

-tty *устройство*

-t *устройство*

Запуститься, используя *устройство* для стандартного ввода и вывода вашей программы.

-interpreter *интерп*

Использовать интерпретатор *интерп* к качестве интерфейса с управляющей программой или устройством. Подразумевается, что этот ключ должен устанавливаться программами, которые взаимодействуют с GDB, используя его как выходной буфер. Например, '--interpreter=mi' велит GDB использовать *интерфейс gdbmi* (see section [Интерфейс GDB/MI](#)).

-write

Открыть выполняемый файл и файл дампа памяти как для чтения, так и для записи. Это эквивалентно команде GDB 'set write on' (see section [Внесение изменений в программу](#)).

-statistics

Этот ключ велит GDB печатать статистику о времени и использовании памяти после завершения каждой команды и возврата к приглашению.

-version

Этот ключ велит GDB напечатать номер своей версии и объявление об отсутствии гарантий, и затем завершиться.

Выход из GDB

quit [*выражение*]**q**

Чтобы выйти из GDB, используйте команду quit (сокращенно q), или введите знак конца файла (обычно C-d). Если вы не укажете *выражение*, GDB закончит работу нормально; в противном случае, он использует результат *выражения* как код ошибки.

Прерывание (часто C-c) не приводит к выходу из GDB, а завершает любую выполняющуюся команду и возвращает вас на командный уровень. Вы можете безопасно пользоваться прерыванием в любое время, потому что GDB не позволяет ему вступить в силу до того, как это станет безопасным.

Если вы использовали GDB для управления присоединенным процессом или устройством, вы можете освободить его командой detach (see section [Отладка запущенного ранее процесса](#)).

Команды оболочки

Если вам случайно потребовалось выполнить команды оболочки в течение сеанса отладки, нет смысла приостанавливать или покидать GDB; вам достаточно воспользоваться командой shell.

shell *командная строка*

Вызвать стандартную оболочку для выполнения *командной строки*. Переменная окружения SHELL, если она существует, определяет, какую оболочку запустить. В противном случае, GDB использует оболочку по умолчанию ('/bin/sh' в системах Unix, 'COMMAND.COM' в MS-DOS, и так далее).

В средах разработки часто бывает необходимо воспользоваться утилитой make. Для этой цели вам не обязательно пользоваться командой shell в GDB:

make *make-arg*

Выполнить программу make с указанными аргументами. Это эквивалентно 'shell make *make-arg*'.

Go to the [first](#), [previous](#), [next](#), [last](#) section, [table of contents](#).

Спонсоры:



При поддержке
inferno solutions*

Хостинг:



Hoster.ru
хостинг провайдер

[Закладки на сайте](#)
[Проследить за страницей](#)

Created 1996–2022 by [Maxim Chirkov](#)
[Добавить](#), [Поддержать](#), [Вебмастеру](#)
