

Визуальная отладка с помощью gdbgui

В.А. Галатенко¹, К.А. Костюхин², А.А. Фролова³

¹Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», Москва, РФ, galat@niisi.ras.ru;

²Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», Москва, РФ, kost@niisi.ras.ru;

³Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», Москва, РФ, frolova-a@niisi.ras.ru

Аннотация. Интерактивный отладчик gdb является основным средством отладки программ, разрабатываемых на инструментальной платформе Linux. Предоставляя широкие возможности для отладки, gdb имеет один (по мнению достаточно большого числа разработчиков) существенный недостаток – интерфейс командной строки. Многие разработчики по-прежнему предпочитают более удобные графические интерфейсы. В этой статье будет рассмотрена перспективная графическая оболочка для gdb, являющаяся браузерным клиентом, за счет чего у пользователей появляется возможность платформеннонезависимой отладки.

Ключевые слова: отладка, gdb, графическая оболочка, GUI

1. Введение

Отладчик gdb [1], предоставляющий пользователю интерфейс командной строки, используется при разработке программ в UNIX-подобных системах уже давно. Все это время для него создавалось и создается большое количество графических оболочек (так называемых фронтендов), взаимодействующих с gdb через специальный «машинный» интерфейс (gdb/MI [2]).

В этой статье речь пойдет о сравнительно новом интерфейсе отладчика gdb – веб-интерфейсе под названием gdbgui [3]. Само слово веб-интерфейс говорит о том, что для отладки можно использовать любой современный браузер, поддерживающий в данном случае typescript. На рис. 1 представлен внешний вид gdbgui.

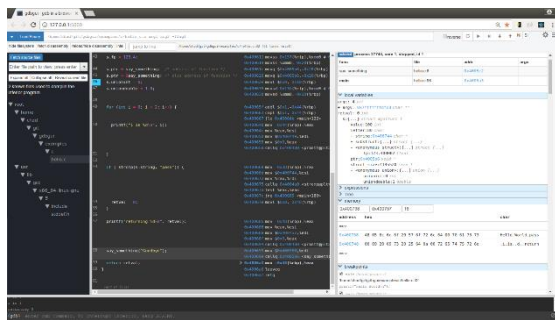


Рис. 1. Внешний вид gdbgui

2. Архитектура gdbgui

Технически gdbgui представляет собой веб-сервер, разворачиваемый на инструментальной ЭВМ, где располагается отладчик gdb. По умолчанию он использует порт 5000.

При установке соединения gdbgui запускает

отладчик (по умолчанию командой gdb) и затем начинает сеанс отладки с использованием текстового интерфейса gdb/MI. По умолчанию (см. ниже) gdbgui также запускает локальный веб-браузер, с помощью которого предполагается вести отладку.

3. Установка gdbgui

Коммуникационная часть gdbgui написана на Python 3, в то время, как команды отладки – на typescript. Это удобно, поскольку дает возможность использовать gdbgui в разных операционных системах. В рамках данной статьи мы ограничимся описанием установки и функционирования gdbgui под управлением ОС семейства Linux.

Для установки gdbgui рекомендуется использовать команду pipx (на момент запуска этих команд python3 уже должен быть установлен в системе).

1. Установка pipx

`python3 -m pip install --user pipx`

2. Добавление нового каталога с выполняемыми программами в окружение пользователя

`python3 -m userpath append ~/.local/bin`

3. Установка gdbgui

`pipx install gdbgui`

Отметим, что gdbgui можно запускать и без установки командой

`pipx run gdbgui`

4. Отладка с помощью gdbgui

4.1. Параметры запуска gdbgui

Список всех параметров запуска можно посмотреть при помощи команды `gdbgui --help`.

```
$ gdbgui --help
usage: gdbgui [-h] [-g gdb_CMD] [-p PORT]
[--host HOST] [-r] [--auth-file AUTH_FILE]
[--user USER] [--password PASSWORD]
[--key KEY] [--cert CERT]
[--remap-sources REMAP_SOURCES]
[--project PROJECT] [-v] [-n] [-b BROWSER]
[--debug] [--args ...] [debug_program]
```

Ниже приводится их краткое описание.

`debug_program` - Отлаживаемый исполняемый файл и (опционально) его параметры командной строки. Чтобы передать параметры в отлаживаемый файл, их необходимо заключить в одинарные кавычки или использовать параметр `--args`.

Пример: `gdbgui ./mybinary [другие параметры gdbgui]`

`gdbgui './mybinary --arg1 --arg2' [другие параметры gdbgui]`

`-g <команда для запуска gdb>`, `--gdb-cmd <команда для запуска gdb>`

`<команда для запуска gdb>` может содержать параметры, с которыми отладчик должен быть запущен, например, `'/path/to/gdb --command=FILE -ix'`.

По умолчанию запускается просто `gdb`.

`-p <номер порта>`, `--port <номер порта>`

Порт, через который `gdbgui` будет работать с удаленным пользователем. По умолчанию 5000.

`--host <ip адрес или имя хоста>`

Адрес или имя хоста, на котором будет работать `gdbgui`. По умолчанию 127.0.0.1.

`-r`, `--remote`

Ключ, указывающий, что ожидается удаленное соединение с `gdbgui` с другого хоста, поэтому локальный веб-браузер не запускается (отменить запуск локального веб-браузера можно и с помощью ключа `-n`).

`--auth-file <файл с учетными данными>`

Если указан этот ключ, то после установки соединения с веб-браузером, будет запрошена учетная информация (логин, пароль). Учетные данные для сверки (логин и пароль) должны находиться в указанном файле в текстовом виде, разделенные переводом строки.

`--cert <SSL сертификат>`

Включает авторизацию по SSL. В качестве аргумента указывается SSL сертификат. Сгенерировать сертификат можно следующей командой:

```
openssl req -newkey rsa:2048 -nodes -keyout
host.key -x509 -days 365 -out host.cert
```

`--project <Каталог исходных текстов проекта>`

При старте `gdbgui` строит дерево подкаталогов исходных текстов проекта из каталога, переданного в качестве аргумента.

`-v`, `--version`

Печатает текущую версию `gdbgui`.

`-b <веб-браузер>`, `--browser <веб-браузер>`

Локально запускает указанный веб-браузер вместо прописанного в системе по умолчанию.

`--debug`

Запуск `gdbgui` в режиме отладки. Если указан этот ключ, то в отдельном окне будут отображаться вызовы и результаты их выполнения `gdb/MI` интерфейса.

4.2. Обзор графического интерфейса

Проиллюстрируем функционал программы `gdbgui` на примере с рисунка 1.

В верхней части страницы находится поле «*Load Binary*», которое можно использовать для загрузки исполняемых файлов программы, а также передавать им аргументы, как в командной строке (рис. 2).

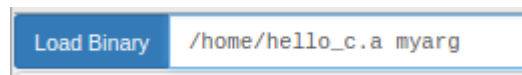


Рис. 2. Поле для загрузки файлов `gdbgui`

Там же справа располагаются кнопки управления ходом выполнения отлаживаемой программы: запустить/перезапустить, продолжить/остановить, продолжить по шагам (по строкам исходного кода) с заходом в вызываемую функцию или нет, выполнить одну команду процессора с заходом в вызываемую функцию или нет (рис. 3).



Рис. 3. Кнопки управления `gdbgui`

Для некоторых из этих элементов управления также есть сочетание клавиш.

Запуск программы: `г`

Продолжение выполнения: `с`

Шаг без захода в функцию: `п` или стрелка вправо.

Шаг с заходом в функцию: `с` или стрелка вниз.

В нижней части страницы находится традиционный интерфейс командной строки `gdb` (рис. 4).

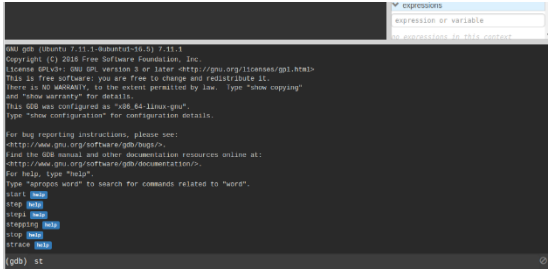


Рис. 4. Консоль gdb

Им можно воспользоваться для ввода команд отладчика, не имеющих своего отдельного графического элемента управления. Например, задать специфическую цель отладки командой `target`.

После загрузки исполняемого файла становится доступен раздел в середине страницы, в котором отображается исходный код программы (рис. 5) с расставленными точками останова.

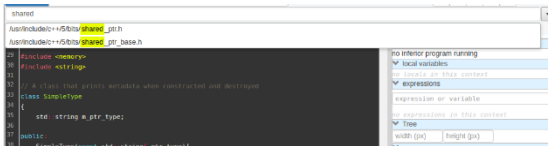


Рис. 5. Исходный текст программы

При необходимости можно здесь же смотреть дизассемблированный код программы (рис. 6).

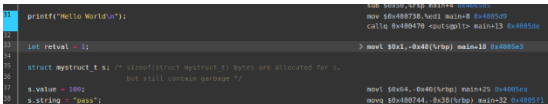


Рис. 6. Исходный и дизассемблированный текст

Если исходный текст недоступен, то отображается только дизассемблированный код (рис. 7).

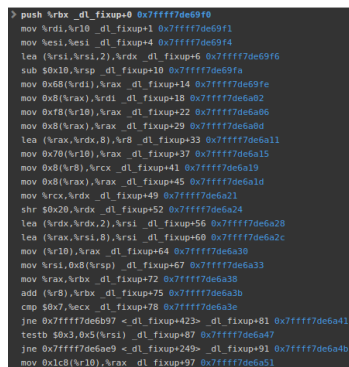


Рис. 7. Дизассемблированный код

Слева располагается выпадающий список всех исходных файлов программы, доступных для отладки (рис. 8).

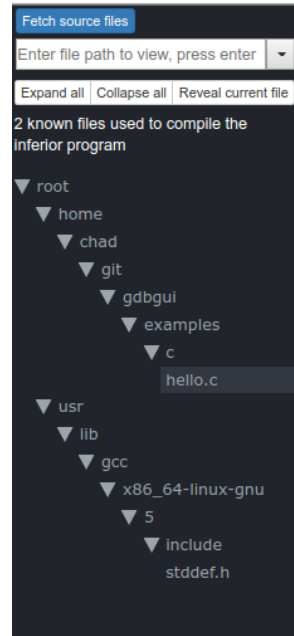


Рис. 8. Список исходных файлов

С правой стороны также находится несколько разделов.

Раздел «Threads» содержит информацию об отлаживаемых потоках и их стеке вызовов. Там же можно выбрать нужный поток для отладки.

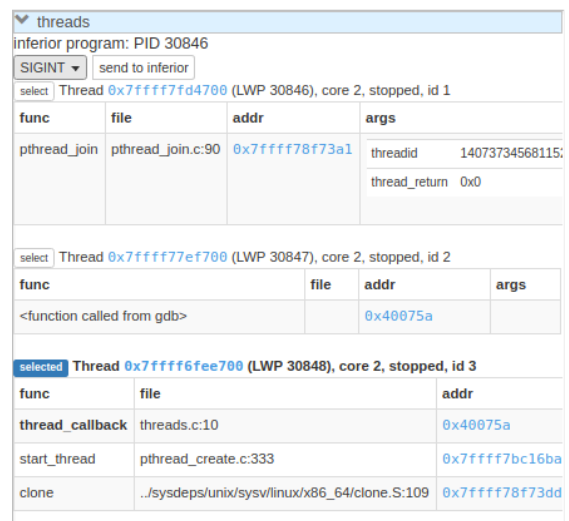


Рис. 9. Список потоков

Под ним располагается раздел с информацией о локальных переменных (рис. 10).



Рис. 10. Список локальных переменных

Также текущее значение переменной можно посмотреть, просто наведя на ее имя мышку (рис. 11).

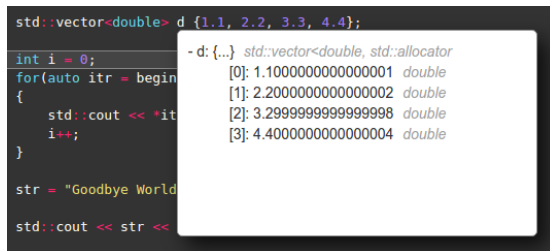


Рис. 11. Просмотр значения переменной

В разделе «Expressions», можно задать выражения, которые будут вычисляться при каждой остановке программы (рис. 12).

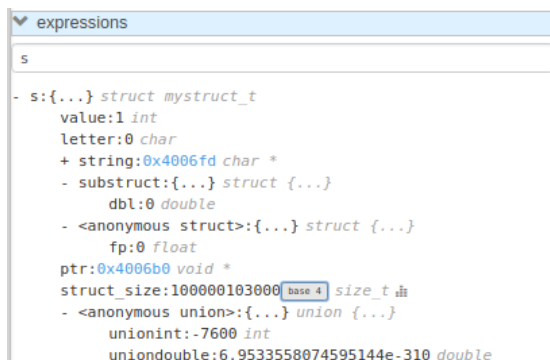


Рис. 12. Выражения

Вычисленные значения выражений могут отображаться на координатной сетке, позволяющей проследить историю изменений заданного выражения (рис. 13).

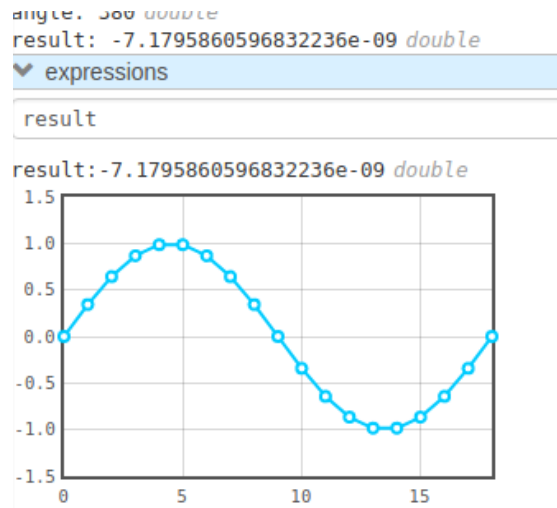


Рис. 13. Изменение значения выражения

Ниж располагается раздел, в котором можно посмотреть содержимое памяти отлаживаемой программы (рис. 14). Следует обратить внимание, что все шестнадцатеричные адреса gdbgui делает гиперссылками для удобства изучения памяти.

memory			
0x400738	0x400758	10	
address	hex		char
more			
0x400738	48 65 6c 6c 6f 20 57 6f 72 6c		Hello.Worl
0x400742	64 00 70 61 73 73 00 69 20 69		d.pass.i.i
0x40074c	73 20 25 64 0a 00 72 65 74 75		s..d..retu
0x400756	72 6e 69 6e 67 20 25 64 0a 00		rning..d..
0x400760	47 6f 6f 64 62 79 65 00 66 66		Goodbye.ff
0x40076a	66 66 66 be 81 40 cd cc f6 42		fff.....B
0x400774	00 00 00		...

Рис. 14. Содержимое памяти

Далее находится раздел с регистрами текущего отлаживаемого потока (рис. 15). Все измененные с момента предыдущей остановки программы регистры отображаются желтым цветом.

registers		
name	value (hex)	value (decimal)
rax	0xd	13
rbx	0x0	0
rcx	0x1000	4096
rdx	0x7fff7dd3780	140737351858048
rsi	0x400688	4195976
rdi	0x7fff7dd2620	140737351853600
rbp	0x7ffffe000	140737488347136
rsp	0x7ffffd4d0	140737488347088
r8	0x602000	6299648
r9	0xd	13
r10	0x7fff7dd1b78	140737351850872

Рис. 15. Регистры программы

На отдельной вкладке располагается панель управления сеансами gdbgui (так называемая dashboard, рис. 16).

active gdb processes managed by gdbgui

gdb path	command	gdb pid	number of connected browser tabs		
/usr/bin/gdb	/usr/bin/gdb --interpreter=mi2	4623	1	View	Kill process
/usr/bin/gdb	/usr/bin/gdb --interpreter=mi2	4596	2	View	Kill process
/usr/bin/gdb	/usr/bin/gdb --interpreter=mi2	4619	1	View	Kill process
/usr/bin/gdb	/usr/bin/gdb --interpreter=mi2	4617	4	View	Kill process

[+ open tab with new gdb process](#)

Рис. 16. Панель управления сеансами отладки

Разработчик имеет возможность переключаться между разными сеансами отладки, участвуя в них или активно (только один пользователь может вести сеанс отладки), или пассивно в качестве наблюдателя.

5. Заключение

Авторами была произведена доработка текущей версии gdbgui, исправлен ряд ошибок, касающийся, в частности, отображения регистров отлаживаемой программы. По результатам тестирования gdbgui его можно рекомендовать к

использованию вместе с любым отладчиком, поддерживающим gdb/MI интерфейс.

Одним из главных плюсов можно выделить его независимость от инструментальной системы, на которой выполняется отладчик gdb, что несомненно облегчает удаленную отладку. Клиентом отладки может выступить любое устройство, имеющее веб-браузер с поддержкой typescript. Также следует упомянуть простоту кастомизации gdbgui, что позволяет его адаптировать под конкретные задачи разработки и отладки сложных систем. Кроме того, gdbgui дает разработчику возможность устроить совместный сеанс отладки с другими даже удаленными разработчиками, что увеличивает вероятность оперативного обнаружения и устранения ошибки.

«Публикация выполнена в рамках государственного задания по проведению фундаментальных исследований по теме «Исследование и реализация программной платформы для перспективных многоядерных процессоров» (FNEF-2022-002).»

Visual Debugging with gdbgui

Vladimir Galatenko, Konstantin Kostiukhin, Anastasiia Frolova

Abstract. The gdb interactive debugger is the main debugging tool for programs developed on the Linux platform. Providing many opportunities for debugging, gdb has one (according to a large number of developers) significant drawback – the command-line interface. Many developers still prefer more convenient graphical interfaces. In this article, we will consider a promising graphical frontend for gdb, which is a browser client, due to which users have the opportunity for platform-independent debugging.

Keywords: debugging, gdb, frontend, GUI

Литература

1. GDB: The GNU Project Debugger, <https://www.sourceware.org/gdb>
2. Галатенко В.А., Костюхин К.А. Машинно-ориентированный текстовый интерфейс отладчика GDB // Труды научно-исследовательского института системных исследований Российской академии наук, 2022, Т. 12, № 4, стр. 38–42.
3. GDBGUI: A browser-based frontend to gdb (gnu debugger), <https://www.gdbgui.com>