(http://peeterjoot.com) gdb TUI mode debugging

June 9, 2016 C/C++ development and debugging. (http://peeterjoot.com/category/cc-development-and-debugging/) debugging (http://peeterjoot.com/tag/debugging/), gdb (http://peeterjoot.com/tag/gdb/), putty (http://peeterjoot.com/tag/putty/), tui (http://peeterjoot.com/tag/tui/)

Facebook

Twitter

LinkedIn

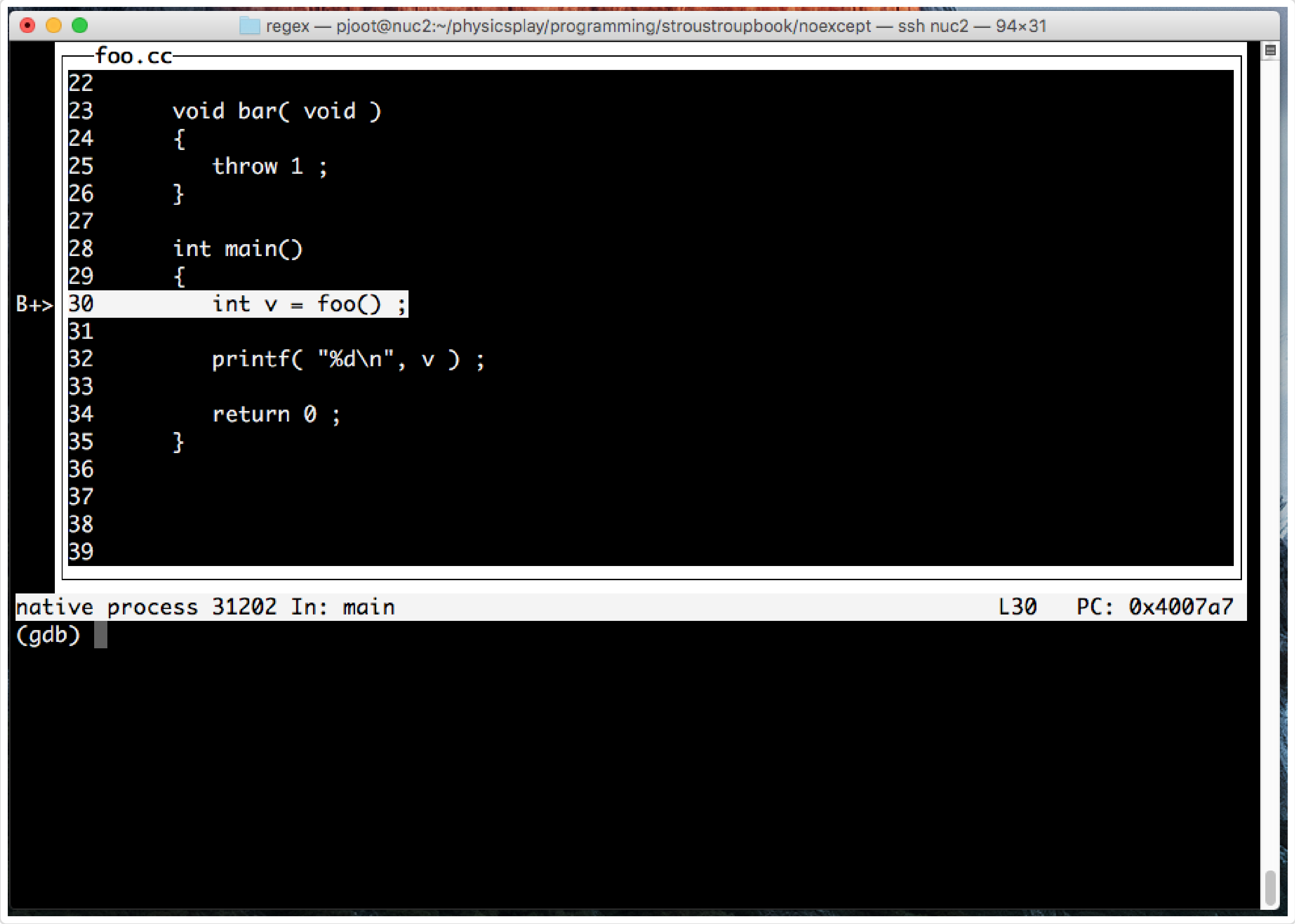
I recently watched Greg Law's "I'll change your opinion on GDB" talk from CPPCON 2015 (https://www.youtube.com/watch?v=PorfLSr3DDI), where he gave a nice example of how to use the gdb TUI (text user interface) mode. I'd recently encountered the gdb –tui option, but thought that you had to have the foresight to remember to invoke gdb with –tui (which I usually didn't, but also didn't always want to since it's redraw was flaky in a putty terminal session).

It turns out that there are command line keystrokes to enable TUI dynamically when you want it (or turn it off when you don't). That makes this TUI a much nicer feature!

The main key commands of interest (https://sourceware.org/gdb/onlinedocs/gdb/TUI-Keys.html) are:

– C-x a, turn on (or off) TUI mode.
– C-x 2. Use two windows, or cycle to the next layout.
– C-x o. Change active window.
– C-L. Redraw screen.
– C-x p. command history (since up and down move the cursor)

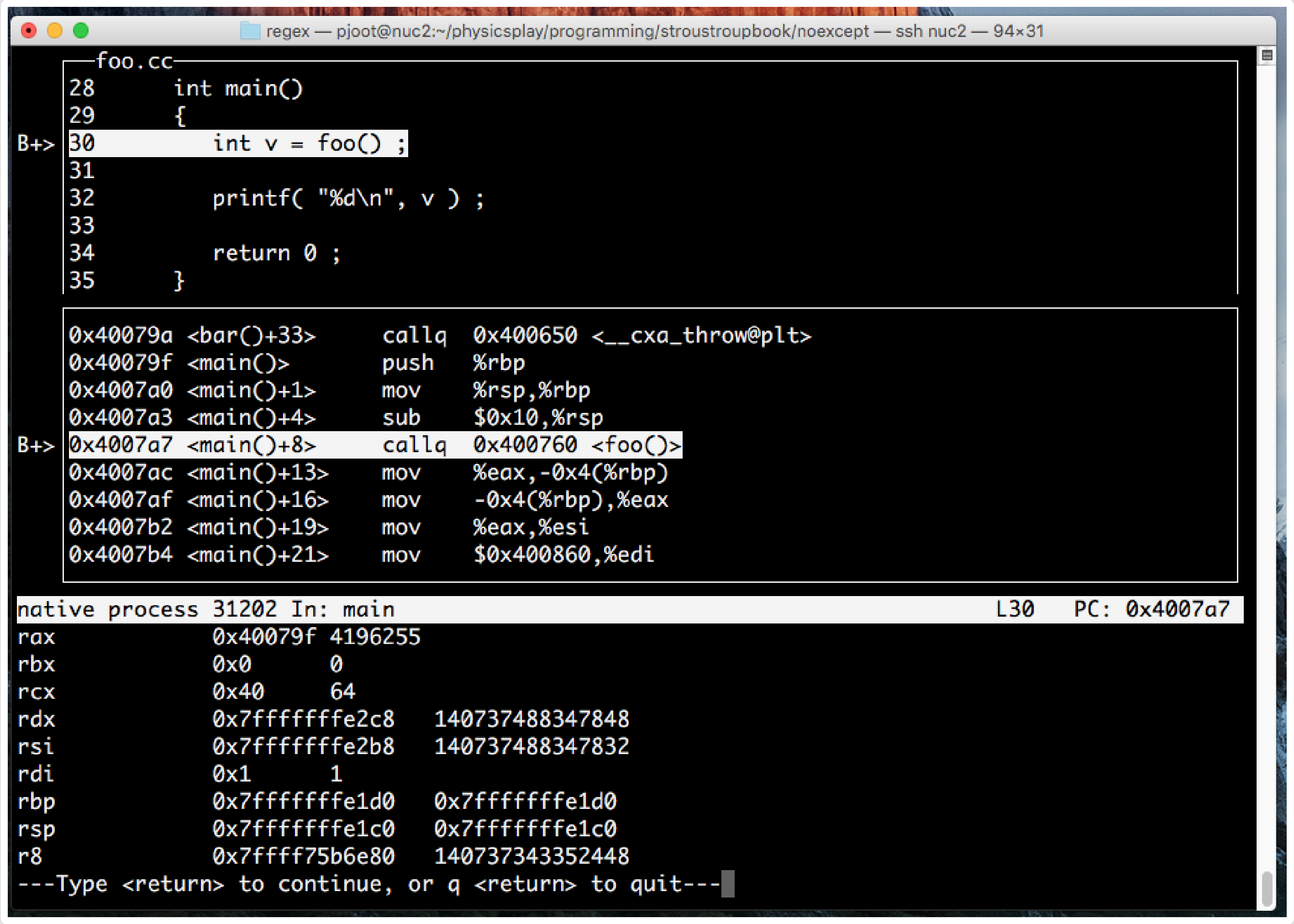
Here's an example how things look after turning on TUI (C-x a):


regex — pjoot@nuc2:~/physicsplay/programming/stroustroupbook/noexcept — ssh nuc2 — 94×31

```
    foo.cc
    22
    23      void bar( void )
    24      {
    25          throw 1 ;
    26      }
    27
    28      int main()
    29      {
B+> 30          int v = foo() ;
    31
    32          printf( "%d\n", v ) ;
    33
    34          return 0 ;
    35      }
    36
    37
    38
    39
native process 31202 In: main                                    L30   PC: 0x4007a7
(gdb)
```

(http://peeterjoot.com/wp-content/uploads/2016/06/Screen-Shot-2016-06-09-at-7.42.59-PM.png)

After turning on two window mode (C-X 2):

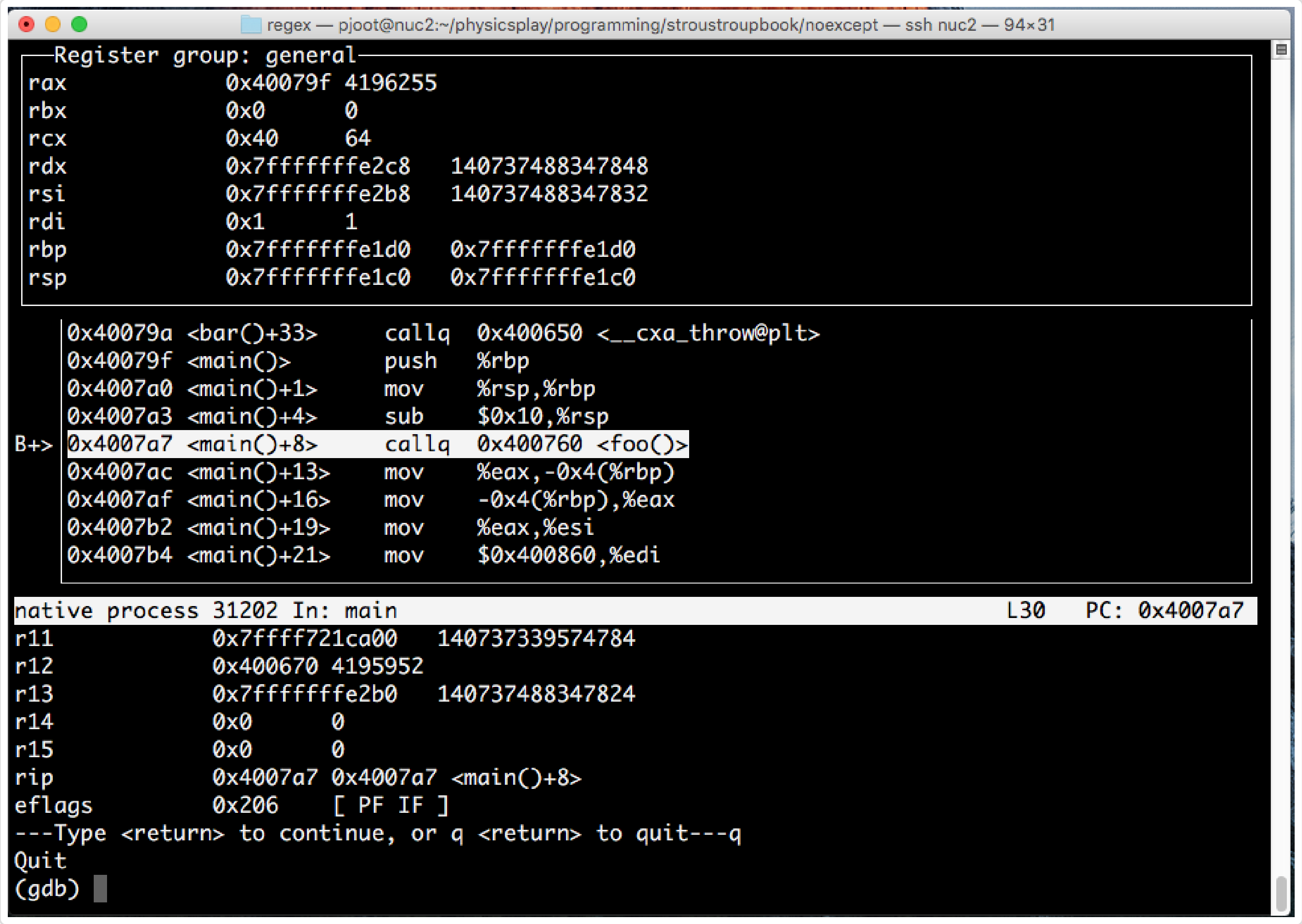

regex — pjoot@nuc2:~/physicsplay/programming/stroustroupbook/noexcept — ssh nuc2 — 94×31

```
    ┌─foo.cc─────────────────────────────────────────────────────────────────┐
    │28      int main()
    │29      {
B+> │30          int v = foo() ;
    │31
    │32          printf( "%d\n", v ) ;
    │33
    │34          return 0 ;
    │35      }

    ┌────────────────────────────────────────────────────────────────────────┐
    │0x40079a <bar()+33>     callq  0x400650 <__cxa_throw@plt>               │
    │0x40079f <main()>       push   %rbp                                     │
    │0x4007a0 <main()+1>     mov    %rsp,%rbp                                │
    │0x4007a3 <main()+4>     sub    $0x10,%rsp                               │
B+> │0x4007a7 <main()+8>     callq  0x400760 <foo()>                         │
    │0x4007ac <main()+13>    mov    %eax,-0x4(%rbp)                          │
    │0x4007af <main()+16>    mov    -0x4(%rbp),%eax                          │
    │0x4007b2 <main()+19>    mov    %eax,%esi                                │
    │0x4007b4 <main()+21>    mov    $0x400860,%edi                           │
    └────────────────────────────────────────────────────────────────────────┘
native process 31202 In: main                                 L30   PC: 0x4007a7
rax            0x40079f 4196255
rbx            0x0      0
rcx            0x40     64
rdx            0x7fffffffe2c8   140737488347848
rsi            0x7fffffffe2b8   140737488347832
rdi            0x1      1
rbp            0x7fffffffe1d0   0x7fffffffe1d0
rsp            0x7fffffffe1c0   0x7fffffffe1c0
r8             0x7ffff75b6e80   140737343352448
---Type <return> to continue, or q <return> to quit---
```

(http://peeterjoot.com/wp-content/uploads/2016/06/Screen-Shot-2016-06-09-at-7.44.05-PM.png)

This has a source view, an assembly view, and the output of ‘info registers’. You can actually get a register window by cycling through the window layouts once (C-x 2 again) :


regex — pjoot@nuc2:~/physicsplay/programming/stroustroupbook/noexcept — ssh nuc2 — 94×31

```
┌Register group: general─────────────────────────────────────────────────────────────────┐
│rax            0x40079f 4196255                                                          │
│rbx            0x0      0                                                                │
│rcx            0x40     64                                                               │
│rdx            0x7fffffffe2c8   140737488347848                                          │
│rsi            0x7fffffffe2b8   140737488347832                                          │
│rdi            0x1      1                                                                │
│rbp            0x7fffffffe1d0   0x7fffffffe1d0                                           │
│rsp            0x7fffffffe1c0   0x7fffffffe1c0                                           │
└─────────────────────────────────────────────────────────────────────────────────────────┘
   │0x40079a <bar()+33>     callq  0x400650 <__cxa_throw@plt>                              │
   │0x40079f <main()>       push   %rbp                                                    │
   │0x4007a0 <main()+1>     mov    %rsp,%rbp                                               │
   │0x4007a3 <main()+4>     sub    $0x10,%rsp                                              │
B+>│0x4007a7 <main()+8>     callq  0x400760 <foo()>                                        │
   │0x4007ac <main()+13>    mov    %eax,-0x4(%rbp)                                         │
   │0x4007af <main()+16>    mov    -0x4(%rbp),%eax                                         │
   │0x4007b2 <main()+19>    mov    %eax,%esi                                               │
   │0x4007b4 <main()+21>    mov    $0x400860,%edi                                          │
   └───────────────────────────────────────────────────────────────────────────────────────┘
native process 31202 In: main                                            L30   PC: 0x4007a7
r11            0x7ffff721ca00   140737339574784
r12            0x400670 4195952
r13            0x7fffffffe2b0   140737488347824
r14            0x0      0
r15            0x0      0
rip            0x4007a7 0x4007a7 <main()+8>
eflags         0x206    [ PF IF ]
---Type <return> to continue, or q <return> to quit---q
Quit
(gdb) █
```

(http://peeterjoot.com/wp-content/uploads/2016/06/Screen-Shot-2016-06-09-at-7.44.40-PM.png)

Some of the previous times that I'd tried 'gdb –tui' explicitly, I'd found it hit redraw issues, but didn't really want to detach and reattach the debugger just to change the TUI mode. It sounds like such issues can be dealt with either by turning off TUI mode (C-x a), or using redraw (C-L).

Share this:

- ✉ Email (http://peeterjoot.com/2016/06/09/gdb-tui-mode-debugging/?share=email)
- Facebook (http://peeterjoot.com/2016/06/09/gdb-tui-mode-debugging/?share=facebook)
- LinkedIn (http://peeterjoot.com/2016/06/09/gdb-tui-mode-debugging/?share=linkedin)
- WhatsApp (http://peeterjoot.com/2016/06/09/gdb-tui-mode-debugging/?share=jetpack-whatsapp)
- Reddit (http://peeterjoot.com/2016/06/09/gdb-tui-mode-debugging/?share=reddit)
- Telegram (http://peeterjoot.com/2016/06/09/gdb-tui-mode-debugging/?share=telegram)
- Twitter (http://peeterjoot.com/2016/06/09/gdb-tui-mode-debugging/?share=twitter)

One comment

Peeter Joot's Blog » peeking into relocation of function static in shared library 🏠

(http://peeterjoot.com/2017/02/27/peeking-into-relocation-of-function-static-in-shared-library/) 4 years ago

Reply Preview ? Help

Markdown enabled

Name

E-mail

URL

□ **Notify me of follow-up comments by email.**

□ **Notify me of new posts by email.**

Post Comment

Back to top

© **Peeter Joot's Blog**

Theme by nehalist.io (http://nehalist.io)