Summer 2016 CS61C GDB Shortcut Manual

Alex Jing

June 19, 2016

Credits to cs162 hw0 notes.

1 GDB

- run, r
  Start program execution from the beginning of the program. Also allows argument passing and basic I/O redirection basic I/O redirection
- quit, q
  exit GDB
- kill
  Stop program execution
- break, break x if condition
  Suspend program at specified function (e.g. break strcpy) or line number (e.g. break file.c:80).
- info -b
  Give a list view of all the currently set breakpoints
- clear
  The clear command will remove the current breakpoint.
- delete
  Use this command without any arguments will delete all break points. You can also give it arguments.(e.g. break breakpoint1)
- step, s
  If the current line of code contains a function call, GDB will **step into** the body of the called function. Otherwise, GDB will execute the current line of code and stop at the next line.

- next, n
  Execute the current line of code and stop at the next line. Do not use this command if you know your bug is somewhere in the next line, use step to locate where exactly the bug is.

- up
  The command up can be used to examine the contents of other active frames, by moving the focus up the stack, that is to say from callee to caller, one frame at a time. This is very useful to debug segfault when you have many files and you do not know where the segfault is. (break at the offending line, use up to uncover what's going on).

- down
  The inverse of the previous instruction.

- continue, c
  Continue execution (until the next breakpoint)

- finish
  Continue to end of the current function.

- print, p
  Print value stored in variable. You can access all the variables in the current scope. Use this command just like inline print statements. Remember you can print out an expression.(e.g. print struct1->value or print *str)

- display
  Like print, but will print out the value every step. Use this command to save some time if you want to monitor the changes to a variable.

More info at **HERE**