

CS 240 Laboratory 6

More on GNU Debugger (gdb)

Tutorials and manuals:

<http://wellesleycs240.bitbucket.org/tools.html>

Commands

Can be shortened to a single letter, or repeated by entering <return> at the prompt):

- Compile C program with **-g** option to create debugging information:

```
$ gcc -g -o testprog testprog.c
```

- Run the program under **gdb**

```
$ gdb testprog
```

```
(gdb) run
```

- Set breakpoints

```
(gdb) break main
```

- Step/next statement by statement through your program

```
(gdb) step
```

```
(gdb) next
```

- Display/print code or values of variables and arguments

(gdb) **list**

(gdb) **print x**

(gdb) **info locals**

(gdb) **info args**

- Examine function calling context/backtraces

(gdb) **backtrace**

(gdb) **up**

(gdb) **down**

- (gdb) **cont** -- to continue execution of program
- (gdb) **quit** or **Ctrl-d** -- to exit.

We would also like to be able to do the following, especially for the next assignment:

- To find a bug:
 1. Set breakpoints at the start of every function
 2. Restart the program and step line-by-line until you locate the problem exactly.
 3. If program is stuck (infinite loop) **Ctrl-c** terminates the action of any GDB command that is in progress and returns to GDB command level. It is safe to type the interrupt character at any time because GDB does not allow it to take effect until a time when it is safe.

- Execute statements/expressions during execution to tweak program execution state

(gdb) set var i = 2

- Display/print binary and hexadecimal representation of variables and arguments

(gdb) print /x result -- uses hex representation

(gdb) print /t result -- uses binary representation

