

CMSC 216

Introduction to Computer Systems
GDB/Miscellaneous

Project #3a

- Let's discuss hashing
 - Close addressing with linked list chaining to handle collisions
- What you need to implement for this project

GDB/Emacs

- You can access gdb in a non-graphical way (see below reference)
 - <http://www.cs.umd.edu/class/summer2011/cmsc216/resources/gdb-emacs.htm>
- You can also access it via the provided GUI (see next slides)

GDB/Emacs (via GUI)

- Remember you need to run Xming
- You need to compile your program with `-g` option
 - `gcc -g prog.c`
- Start emacs with file to debug
 - `emacs prog.c`
- Select Tools→Debugger(GDB)
 - You will notice at the bottom
 - Run gdb (like this): `gdb -annotate=3 a.out`
- Press enter
- Now you are in the debugger
- Select Gud→GDB-UI-→Display Other Windows
 - Now you will see panes for gdb commands (top left), code (middle), stack (bottom left) and breakpoints (bottom right)
- Options at the top allows you to run the program (GO), step (e.g., { }) and move between stacks
 - You can also perform these operations and more using the (gdb) command line (see next slide)

Some GDB Commands

- **Setting break points**
 - (gdb) b main
 - Sets breakpoint in function main
 - (gdb) b 18
 - Sets breakpoint in line 18
 - Notice you can also set/clear a breakpoint by clicking the gray area to the left of a statement
- **Running program**
 - (gdb) r
 - You need to have a breakpoint to examine data
- **Continue running program after breakpoint** (will stop in next breakpoint or finishes program execution)
 - (gdb) c
- **Execute next statement** (for a function call it will execute function without stepping in)
 - (gdb) n

Some GDB Commands

- **Stepping into a function**
 - (gdb) sbt
- **Displaying stack trace**
 - (gdb) bt
 - You can also see the stack trace in the bottom left pane
- **Changing stack frame**
 - Click on the appropriate entry in the bottom left pane
- **Printing values**
 - (gdb) p y /* y is a variable */
 - (gdb) p/x y /* printing y as hexadecimal */
 - Notice you can see value on top right pane
- **Finish function**
 - (gdb) fin /* continue until current function has returned */

Miscellaneous

- Reviewing pointers
- Reviewing dynamic memory allocation
- String literals and modifications
- Quick overview of two-dimensional arrays
- What is core dump?
- atoi function
- Why you should not use gets