



## Debugging

---

CSCE 310J  
Jason Stark  
November 4, 2004



## Methods of “Debugging”

---

- cout/printf
- Assertions
- Debuggers



## Debuggers

---

- A debugger allows a programmer to see what goes on inside a running program
  1. Start the program in a controlled environment
  2. Stop the program at any point and examine its current state
  3. Find out what caused the program to crash and examine its state at the time of the crash
  4. Change variables within the running program to help find and eliminate bugs



## Terminology

---

- Breakpoint
  - Set a specific point in the code where execution should pause
- Watchpoint
  - Execution pauses when a specific variable is changed and/or read
- Catchpoint
  - Pauses execution when a specific event occurs (e.g. exception, library loaded)



## GDB: The GNU Debugger

---

- Available on csce.unl.edu and most Unix or Linux environments
- Probably the most complete debugger and has the most useful features
- Interfaces
  - Command-line
  - Emacs interface
  - KDevelop
  - Others



## Prerequisites

---

- **IMPORTANT:** Add debugging symbols when compiling

```
# g++ -ggdb -o exec main.cpp
```

  - Debugger is not very helpful without this flag
- Debugging your program

```
# gdb exec
```



## GDB: Tutorial

---

- Login to csce.unl.edu
- Copy example files
  - # cp -R /home/misc/jstark/gdb .
- Enter new gdb directory
  - # cd gdb
- Build and debug broken.c
  - # make broken
    - Makefile includes debug flag
  - # gdb broken



## GDB: Tutorial

---

- List of commands for "broken" tutorial:

```
break main          step 2
run                 continue
break length        undisplay
break length2       list
watch a             continue
info break          display/s z
list                continue (do this a few times)
continue            backtrace
step                continue [repeat until watchpoint hit]
backtrace           list
print l             continue
display l           continue
display/c z[1]     print c
info display        print *c
step                continue
step                quit
```



## References

---

- CSCE 310J Webpage  
<http://www.cse.unl.edu/~goddard/Courses/CSCE310J/Assignments.html>
- The GNU Debugger  
<http://www.gnu.org/software/gdb/gdb.html>
- GDB tutorial by Andrew Gilpin  
<http://www-2.cs.cmu.edu/~gilpin/tutorial/>