Software Testing

Linux Debugging

Linux Debugging

- Uses the gdb debugger
- You must compile with the –g option to include debugging information
  - `gcc -Wall -g -o broken1 broken1.c`
- Gdb is a command line debugger
- To start:
  - `gdb myExecutable`

Gdb Commands

Command	Explanation
break *location*	Sets a breakpoint at the location.could be a file:line or a function
continue	Resume execution and stop at the next
delete *breaknum*	Delete the indicated breakpoint. The numbers come from info breakpoint.
info breakpoints	Displays a list of active breakpoints along with numbers which can be used to delete them.
finish	Continues execution until the end of the current function and then breaks.
print *var*	Display the value of the provided variable or expression.
step	Execute the current line and stop before the next line is executed.
watch *var*	The program whenever the value of the variable is changed and print out the old and new value of the variable.
where	Display a stack trace showing you where you are in the program.

Sample Output

```
(gdb) break broken1.c:9
Breakpoint 1 at 0x8001198: file broken1.c, line 9.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/rob/broken1

Breakpoint 1, function1 (v=0x7ffffffee490, size=15) at broken1.c:9
9 for(i = 0; i < size; i++)
(gdb) print i
$1 = 0
(gdb) step
11 v[i] = i;
(gdb) step
9 for(i = 0; i < size; i++)
(gdb) print i
$2 = 0
(gdb) step
11 v[i] = i;
(gdb) print i
$3 = 1
(gdb)
```