

# COP4530 Recitation Fall 2012

## Week 3

---

### Objective

1. Debugging with DDD / GDB
2. Valgrind

### Debugging with DDD / GDB

#### A. Initial Set-up

If you want to use DDD on linprog you must establish an X11 connection. Here are your options for X servers:

##### Windows

1. If you use OpenSSH, you can install Xming; you will then have to enable X11 connections in the settings
2. If you use Cygwin, there is an X version available

##### Mac OS X

1. Before Mountain Lion, install the developer tools (XCode) and you should have X11, which is terminal + X server
2. Mountain Lion, X11 is no longer maintained by Apple but a replacement called XQuartz, maintained by independent developers, is available and suggested by Apple

##### Linux/BSD

1. You should have an X server, if not there are probably instructions for your specific distro available online

Once you are able to start a X11 connection ssh into linprog with the '-X' flag (or if you use OpenSSH, follow your normal login process).

#### B. Data Display Debugger (DDD)

DDD is a graphical user interface (GUI) for various debugging tools. In this course we will be using DDD as a GUI for the Gnu Project Debugger (GDB).

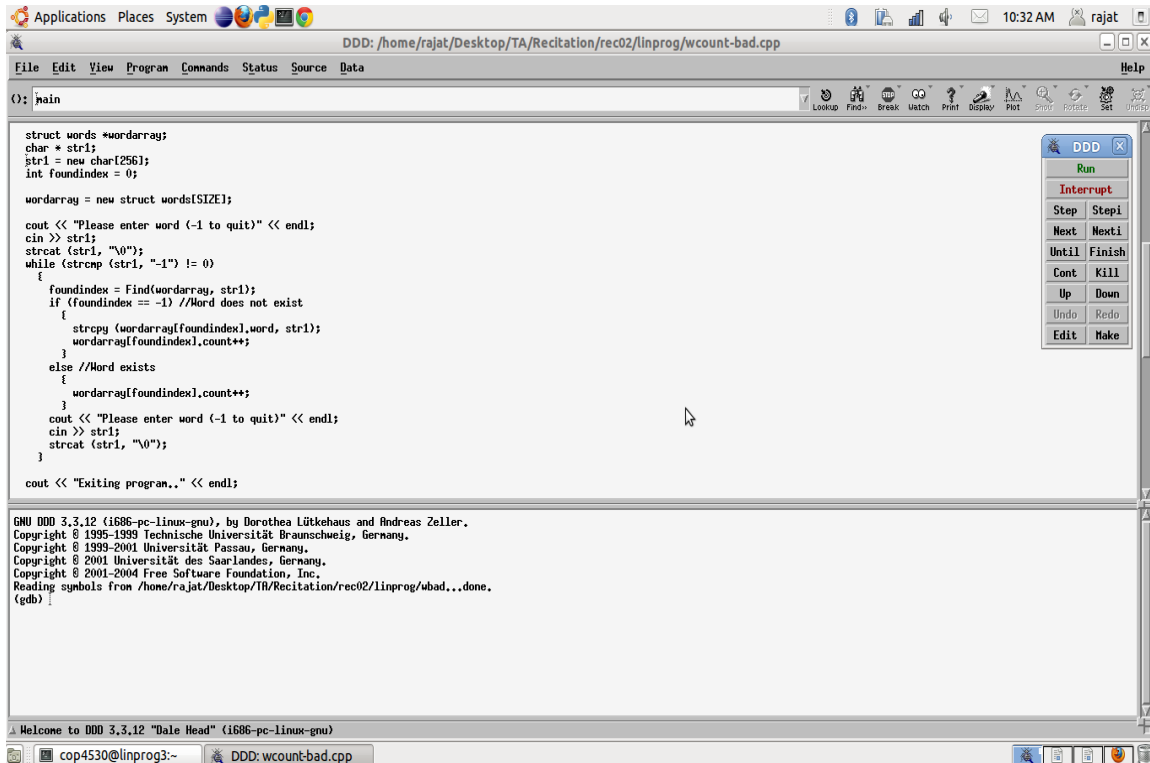
When you compile your programs use the '-g' flag and remember to not use any optimization options. For example:

```
g++ myprogram.cpp -Wall -pedantic -g -o myprogram.x
```

Start DDD using this command:

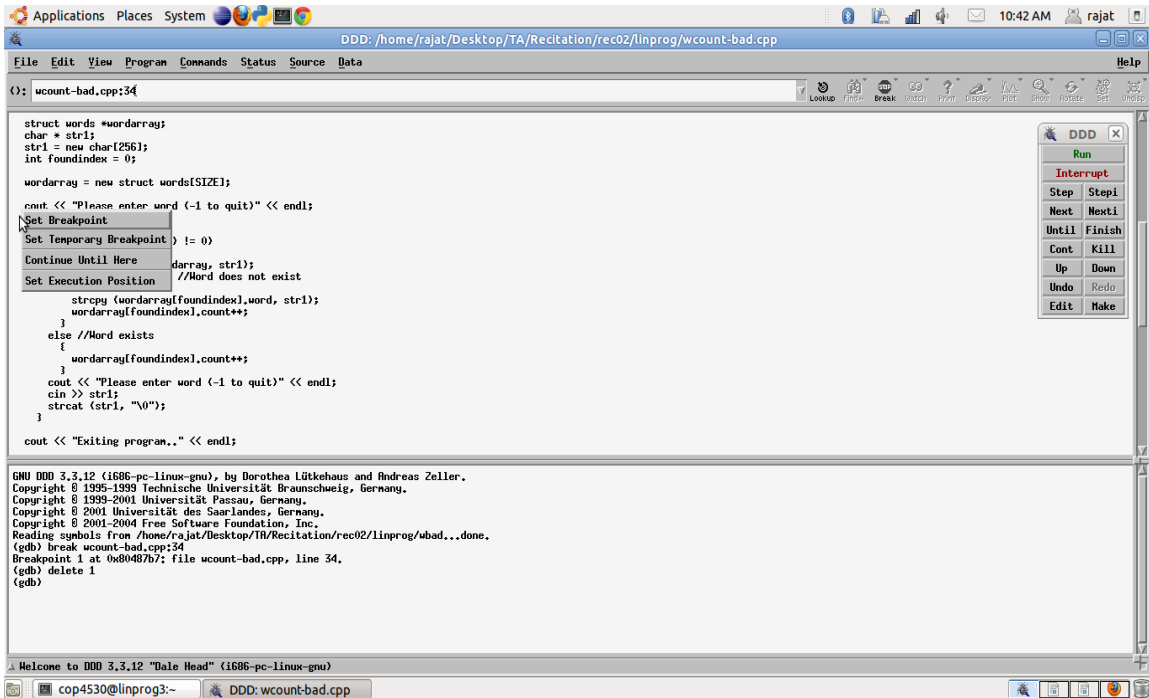
```
ddd myprogram.x
```

You should get a screen looking like this:

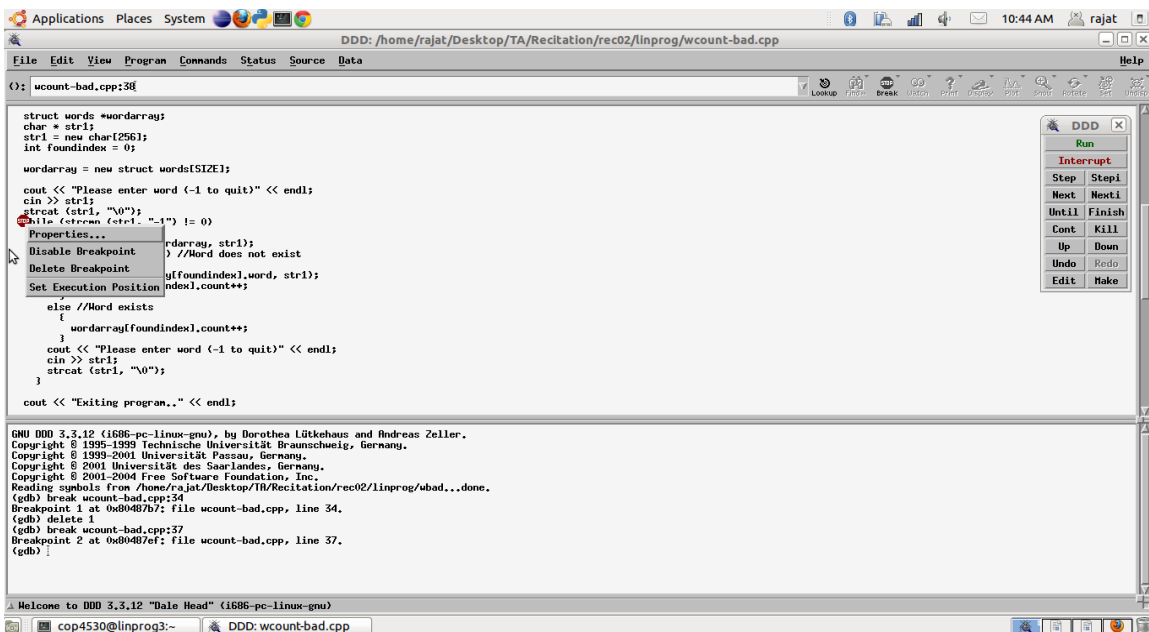


A breakpoint is a signal that tells the debugger to temporarily suspend execution of your program at a certain point. When execution is suspended at a breakpoint, your program is said to be in break mode. Entering break mode does not stop or end the execution of your program; execution can be resumed at any time.

To set the breakpoint take the mouse pointer to the beginning of the line before which you want to set a breakpoint. Then do a right-click, which will provide you with the menu to set a breakpoint.



To clear the breakpoint you can do a right click on that breakpoint and use the menu.

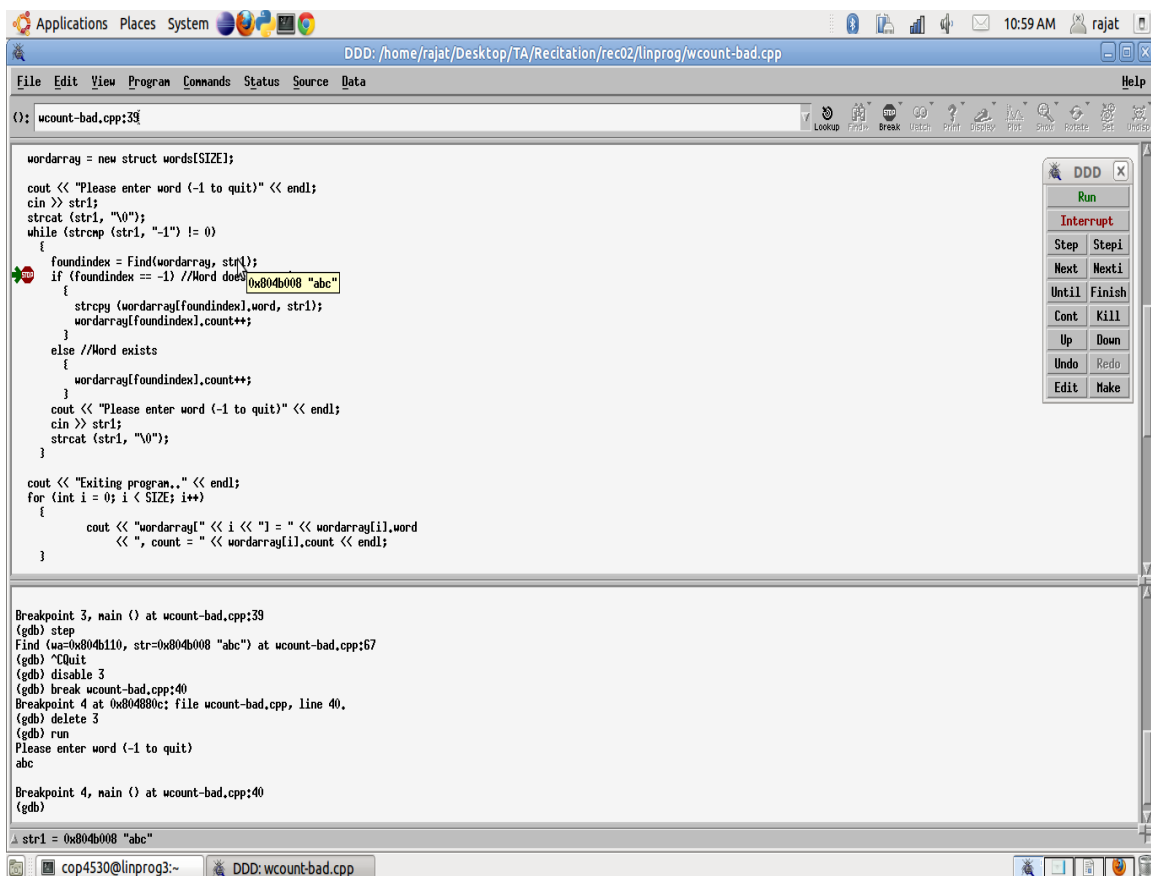


Once your program reaches a breakpoint, you can continue execution by using these buttons:

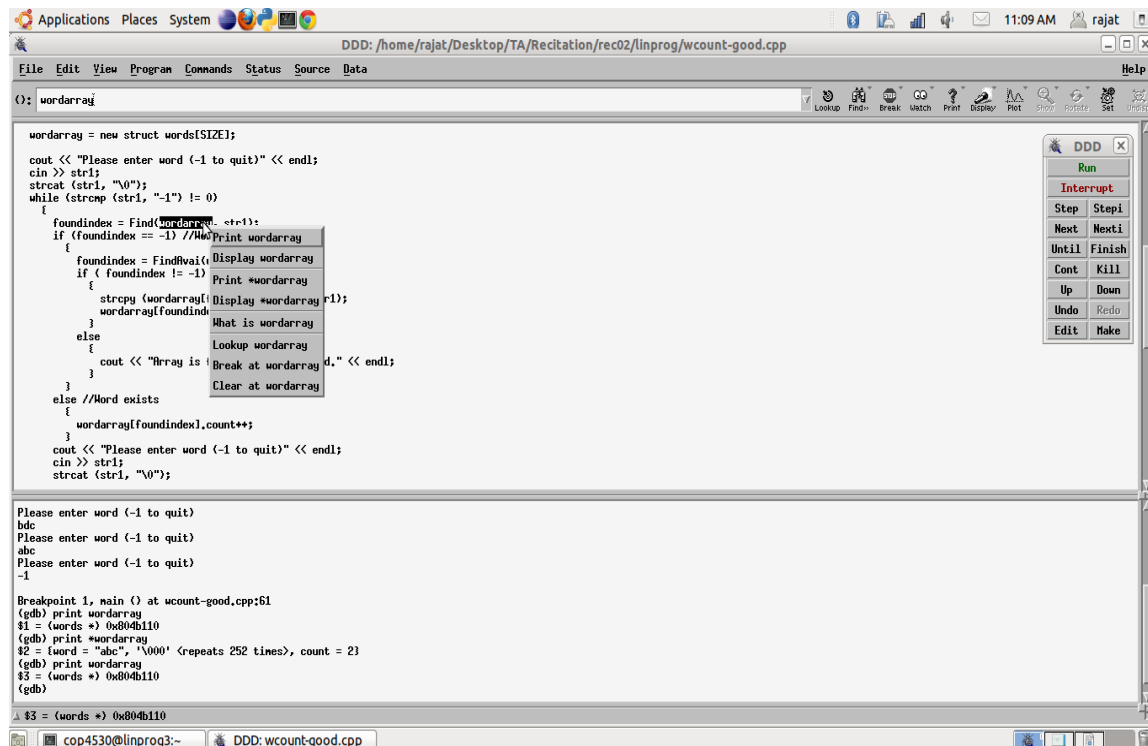
- Step, execute the current line and enter a function if any
- Next, same as Step but do not enter a function
- Cont, continue the execution of the program until the next breakpoint

The DDD documentation covers other commands available in DDD.

While your program is in a suspended, or paused, state you can view the value(s) of program variables. If you have over a variable, its value will appear as a tool tip.



Right-click and select 'print \*<array-name>' to view the contents of an array. What will print if we select 'print <array-name>'?



## C. Gnu Project Debugger (GDB)

If you comfortable with command line interfaces (CLIs), you may find GDB a more pleasant experience. A cheat sheet of GDB commands can be found here: <http://www.yolinux.com/TUTORIALS/GDB-Commands.html>. Recitation will mainly focus on using DDD.

## Valgrind

### A. What Valgrind is and what Valgrind is not

Valgrind is a collection of tools to help detect memory management errors or memory leaks in a program's execution. Valgrind checks for errors dynamically. Therefore although Valgrind may run fine on your test set, another set of input may still cause memory problems with your program despite Valgrind not showing any errors during the initial test set's execution. Additionally Valgrind does not check the bounds of non-dynamic arrays.

## B. Using Valgrind

Starting Valgrind:

```
valgrind myprogam.x
```

Valgrind will then run through the execution of your program and report errors, if any.

If you want to check for memory leaks you can use this:

```
valgrind --leak-check=full myprogram.x
```

Valgrind will run as normal, but at the end of the execution it will report any memory leaks.

## References:

1. DDD: <http://www.gnu.org/software/ddd/>
2. GDB: <http://www.gnu.org/software/gdb/>
3. Valgrind: <http://valgrind.org>