# 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 Mountain Lion, X11 is no longer maintained by Apple but a replacement called XQuartz, maintained by independent developers, is available and suggested by Apple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.

Lie       DDD:/home/rajat/Desktop/TA/Reditation/rec02/linprog/wcount-bad.cpp       Image: Command: Status Source (Bata       Use         Lie       Lie       Command: Status Source (Bata       Use       Image: Command: Status Source (Bata       Use         C       word-dar-agi       total       Image: Command: Status Source (Bata       Use       Image: Command: Status Source (Command: Status Source (Bata       Image: Command: Status Source (Command: Status Source (Commad: Status Source (Command	😴 Applications Places System 🌏 🍪 🦰 🔟 🧔				<b>i</b> a .	đ	<b>d</b> ) 🖂	10:44 AM	🖄 rajat 🛽	٥
File [dit View Program [commands Systems Source [lota]       Use Program [commands Systems Source [lota]         0:1       usent-bad.cpp:30       Image: Program [commands Systems Source [lota]         struct words wordscravg]       image: Program [commands Systems Source [lota]       Image: Program [commands Systems Source [lota]         wordscravg = most TCAS to order[SIZE];       Image: Program [commands Systems Source [lota]       Image: Program [commands Systems Source [lota]         wordscravg = most TCAS to order[SIZE];       Image: Program [commands Systems Source [lota]       Image: Program [commands Source]       Ima	X DDD: /hom	e/rajat/Desktop/TA/Recitation/rec02/linprog/wcount-bad.cpp								×
O: wount-bad.cpp:30 store: words reards reads reads reads reads to be a finite field of the store of the st	<u>File Edit View Program Commands Status Source Data</u>								Hel	р
<pre>strut words =wordersup; char = sti; strl = new char(255); in f voundnest = 0; wordersup = new strut words(SIZE); wordersup = new strut words(SIZE); wordstrang = new str</pre>	(): wcount-bad.cpp:30		V N	ring≫	Break 6	GQ Iston	Print District	NA STOT R	Arte art	22 569
<pre>uurdarrag = neu struct words152E1; conf (&lt; ') Trisse enter word (-1 to quit)" (&lt; endl; torst (::, ') "); int (::, '', '''); int (::, '''', ''''''''''''''''''''''''''''</pre>	<pre>struct words *wordarray; char * strl; strl = new char{2561; int foundindex = 0;</pre>							<b>X</b>	DDD X	X
<pre>cut &lt;</pre>	wordarray = new struct words[SIZE];							Ste	n Steni	
<pre>Line (ref: 1-1) = 0) Properties</pre>	cout << "Please enter word (-1 to quit)" << endl;							Nex	t Nexti	
<pre>cont Kill properties properties production (ort) - 1) [= 0) properties production (ort) - 1) [= 0) properties production (ort) - 1) [= 0) properties production (ort) - 1) [= 0] properties (ort) - 1] [= 0] [</pre>	streat (str1, "\0");							Unt	1 Finish	Fi (
Winzable Breakpoint       Convertions         Delate Breakpoint       Ufoundines: Lowd, strll;         Delate Breakpoint       Ufoundines: Lowd, strll;         Set Escention Position Medil.count+*;       und Peda         und rugh foundindes!.count+*;       und rugh foundindes!.count+*;         cut (~ "Plase entrue und (-1 to quit)" (< endl;	Properties							Con	t Kill	Ш
Delate Breakpoint Set Escoulin Position Position underlag       ufformindeel.uwrd, strl); Edit Make         alse //Nord exists { uwrderragifoundindex].count+*;	Disable Breakpoint ) //Word does not exist							Up	Down	Ш
<pre>lss / lss / l</pre>	Delete Breakpoint y[foundindex].word, str1);							Edi	t Make	Ш
<pre>iss //Mord exists {     subscription     subscriptio</pre>	Set Execution Position Index1.count++;									н
juordaragi tauninneej, count+*; cout (~ These enter word (-1 to quit)* (< endl; cin >> stri; strat (stri, "0"); } cout (< "Exiting program." (< endl; [ONU DDD 3.3.12 (1686-pc-linue-pun), by Dorothes Lättehaus and Andreas Zeller. (Dorgright 5 1959-1959 Inchnische Universität As auschneig, Gernang. Coparight 5 2001 Universität As Saelandes, Gernang. Coparight 5 2001 Universität des Saelandes, Gernang. Coparight 5 2001 Universität des Saelandes, Gernang. Coparight 5 2001 Duiversität des Saelandes, Gernang. Coparight 5 2001 Duiversität des Saelandes, Gernang. Coparight 5 2001 Zubertop/IDREcitation/rec02/liprog/wbaddone. Breakpoint 1 do 0x00407bf; file wcount-bad.cpp, line 34. (gdb) break wcount-bad.cpp; 37 Breakpoint 2 at 0x000407ef; file wcount-bad.cpp, line 37. (gdb) : M Loome to DDD 3.3.12 "Dale Head" (1686-pc-linux-gnu) M cop4530@linprog3:- M DD: wcount-bad.cpp M DD: wcount-bad.cpp	else //Hord exists									
<pre>cout &lt;&lt; "Please enter word (-1 to quit)" &lt;&lt; endl; cin) &gt; str) stroat (stri, "v0"; } cout &lt;&lt; "Exiting program." &lt;&lt; endl; cout &lt;&lt; "Exiting program." &lt;&lt; endl; GNU DDD 3.5,12 (1686-pc-linux-gnu), by Dorothes Lütkehaus and Rodreas Zeller. Coopright 9 2001 Universität des Suerlandes. Gernamy. Coopright 9 2001 Universität des Suerlandes. Gernamy. Coopright 9 2001-2004 Free Software Foundation, Inc. Reading symbols from /home/rajst/Desktop/IM/Recitation/rec02/limprog/wbaddone. (gdb) break wount-bad.cpp; 21 Greaterint 2 at 0x800407ef; file wount-bad.cpp, line 34. (gdb) i (gdb) : Melcome to DDD 3.5,12 "Bale Head" (1686-pc-linux-gnu) Melcome to DDD 3.5,12 "Bale He</pre>	wordarrayLfoundindex1.count++;									
cout < cout < "Exiting program.," << endl;  (dW DDD 3.5.12 (1686-pc-linux-gru), by Dorothea Lükkehaus and Andreas Zeller. Copyright 0 1995-1999 Technische Universität Braunschweig, Bernany. Copyright 0 2993-2001 Universität Braunschweig, Bernany. Copyright 0 2993-2001 Universität Pasaunschweig, Bernany. Copyright 0 2004 Frität off Sarbander, Gen Duit, Copyright 0 2004 Frität off Pasaunschweig, Bernany. Copyright 0 2004 Fritä	<pre>cout &lt;&lt; "Please enter word (-1 to quit)" &lt;&lt; endl; cin &gt;&gt; str1; strcat (str1, "\0"); }</pre>									
<pre>GMU DDD 3.3.12 (1986-pc-linue-pm), by Borethes Lükkehaus and Andreas Zeller. Copyright 5 1955-1957 n-thicks Mu how way, hwelg, Germany. Copyright 5 1955-1957 n-thicks Mu how way, here is a standard stan</pre>	cout << "Exiting program" << endl;									x
 △ Helcone to DDD 3,3,12 "Dale Head" (1686-pc-linux-gnu)	DBU DDD 3.3.12 (1686-pc-linnc-prv), by Dorothes Lükchaus and Madra Copyright 8 1955-1939 Tachinsche Muiserulät Braunschweig, Germany. Copyright 8 1939-2001 Universität Passan, Gernany. Copyright 8 2001-2004 Free Software Foundation, Inc. Reading synbols From Anewr / jakt/Bekktor/IN/Recitation/rec02/linpro Breakpoint 1 at 0004087b7; file uccount-bad.cpp, line 34. (gdb) break uccount-bad.cpp; 37 Breakpoint 2 at 0x004087cf; file uccount-bad.cpp, line 37. (gdb)	ss Zeller. z/ubaddone.								
a melulume tu unu 31312 usla mau (1000-pu-liituk-gmu)						_				臣
📺 📾 coprovemptogo 🗮 coor mountradutpp	Cop 4530@lipprop3:-     A DDD: wcouptbad cop									
	Copersolettinprogs.~ [@ DDD: wcounebad.cpp									

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>'?

n Applications Places System 🕘 🕹 🕂 🔤 📀			8	B	đ	(j)		11:09 AM	🖄 rajat	٥
×.	DDD: /home/rajat/Desktop/TA/Recitation/rec02/linprog/wcount-good.cpp									IX
<u>File Edit View Program Commands Status Source</u>	Data								Į	elp
O: wordarraŭ		V U	諸 <sup>*</sup> Find»	50 Break	QQ Watch	Print	Display		otete Set	Undisp Undisp
<pre>wordarray = new struct words[SIZE]; cout &lt;&lt; "Please enter word (-1 to quit)" &lt;&lt; endl; cin &gt;&gt; str1; stract (str1, "\0"); while (stroep (str1, "1") != 0) { foundindex = FindPusi(to [Sizplay wordarray] if (foundindex = FindPusi(to [Sizplay wordarray] if (foundindex = FindPusi(to [Sizplay wordarray] if (foundindex = FindPusi(to [Sizplay wordarray] auordarray[foundind } else cout &lt;&lt; "Rhray is } else //Word exists { tourdarray[foundindex].count++; } uordarray[foundindex].count++; } cout &lt;&lt; "Please enter word (-1 to quit)" &lt;&lt; endl cin &gt;&gt; str1; strcat (str1, "\0");</pre>	); " << endl; ;							I St Ne Unt Co U U U D G	DDD × Run nterrupt pp Stepi tt Nexti 11 Finish nt Kill > Down do Redo it Make	
Please enter word (-1 to quit) bdc Please enter word (-1 to quit) abc Please enter word (-1 to quit) -1 Breakpoint 1, main () at wcount-good.cpp:61 (gdb) print wordarray \$1 = (words *) 0x804b110 (gdb) print wordarray \$2 = (words *) 0x804b110 (gdb) \$43 = (words *) 0x804b110	= 23									
📷 🔳 cop4530@linprog3:~ 🛛 👗 DDD: wcount-good	l.cpp							× I		6

#### 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: <u>http://www.yolinux.com/TUTORIALS/GDB-Commands.html</u>. 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: <u>http://www.gnu.org/software/ddd/</u>
- 2. GDB: <u>http://www.gnu.org/software/gdb/</u>
- 3. Valgrind: <u>http://valgrind.org</u>