COSC 1436 – Intro to Problem Solving II

Lab 3 UNIX Exercise Answer Sheet

# Erman Gurses

# February 1, 2012

2Aiv. ***Describe what happened. What can you infer about the order of the switches from this?***

Actually it gives e some kinds of errors. These errors point some text section addresses which means that program code because I entered parameters reverse order. The program takes options in a certain order so it gives me these kinds of messages

2Biii. ***Describe the difference(s) between the two commands.***

gdb –q *debug1*: It gives user some license information about that software(In certain conditions). I learned some information for example it is free and open, it has General Public license. Copyrights belong to Free Software Foundation, Inc.

gdb *debug1*: It directly start program before it gives path information about program.

***Which do you prefer? Why?***

I prefer “gdb –q *debug1”* because it is helpful to get some information about what I use. For example, I learned that I can change this program. So maybe my project will be required more debugging feature in the future, I can add some more features in this program therefore this is important information for me.

2Ci. ***Describe in detail what you see.***

First, it shows target’s path then shows output of program and then if program runs without any problem, it prompts “Program exited normally”.

2Cii. Just for grins, type **run** again and press <Enter>. ***What happened?***

It showed same thing which I described as “First, it shows target’s path then shows output of program and then if program runs without any problem, it prompts “Program exited normally”.”

2Eii. ***Describe what you see. What do you think the long number after the word “at” might be?***

I saw a hexadecimal number, filename and line number. The hexadecimal number should be a memory address. This address should be first command or variable of line 13.

2Eiii. ***Describe what you see. What are the differences, if any, between this and the previous command (*break*)?***

I saw almost same thing. I put another break point at line 13 so it shows “Breakpoint 2 at 0x8048747”. Therefore; I think there is no difference between these commands.

2Eiv. Type **run** and then press <Enter>. ***Describe in detail what you see.***

First, it shows target’s path, then shows which function contains this break point, in here main function contain this break point. Moreover it shows source code file and line number. Last one is that it shows (this line has a breakpoint) line’s source code.

2Fii. ***Describe what you see. What is the value of* i*?***

I saw “$1=0”. It is value of “i” which is zero.

2Fiii. ***Describe what you see. What is the value of* sum*? Why do you think it has such a strange value?***

I saw “$2=5304308”. It should be value of “sum”. It has strange value because it has not been so it has garbage value.

2Fiv. ***What is being displayed with this command?***

It showed *numbers* array’s i th element’s value. I shows garbage value because It has not initialized yet.

2Fvi. ***What is displayed with these commands? What are the differences, if any, between this and the previous command (*print*)?***

Before,

*print i* displayed *$1 = 0*

*print numbers[i]* displayed *$3 =1294149*

After I entered *“c* or *continue”* andentered *16* for theprompt *of “Enter the first number”*

*print i* displayed *$4 = 1*

*p numbers[0]* displayed *$5 = 16*

I think “*print* “and “*p”* do same operation.

2Giii. ***What’s different about how the program runs now that you’ve typed the* del break *command?***

It does not stop at line 13 anymore it directly gets second and third number because we removed break point at line 13.

2Giv. ***What would you need to type if you wanted to remove the second breakpoint?***

del break 2

2Gv. ***If you had 6 breakpoints set in a program that had 350 lines of code, what command would you type if you wanted to remove the fourth breakpoint at line 185?***

del break 4

2Iii. ***Describe what you see. What do you think the long number after “numArray=” is? What is your location in the program right now?***

Before I entered *step* command it showed *GetInput()* function on line 15. After I entered *step* command it get into the function therefore it showed me line 28. Because of line 15 has a function it pass a value by reference the hex number should be address of *numArray*. Sure location of program is now on line 28.

2Iiii. ***What do you think this does? Is it any different than the* step *command?***

It does same thing with *step* command. There is no difference between them.

2Jii. ***Describe what you see. What are the differences between this command and the* step *command?***

It skipped *DisplayMessage*() function and went next line after the function. Step enters into function but next does not it runs function but does not show inside the function line by line.

2Jiii. ***What do you think the program would do if you typed* next *again?***

It skipped *GetInput()* function and went next line after the function. However, *GetInput*() has to collect data so even it did not get into function, it runs function and collects data.

2Jiv. ***What do you think this does? Is it any different than the* next *command?***

It does same thing with *next* command. There is no difference between them.

2Kii. Now type the word **continue** and press <Enter>. ***Describe what you see.***

It goes thru until need some value or to another break point.

2Kiv. ***What do you think this does? Is it any different than the* continue *command?***

It does same thing with *continue* command. There is no difference between them.

2Lii. Now type the word **list** and press <Enter>. ***Describe what you see.***

It shows me lines from 9 to 18. Break point is between these lines.

2Liii. ***Describe what you see this time. What do you think would be displayed if you typed* list *again?***

It shows me lines from 19 to 28. If I write list again it will show me lines from 19 to 28.

2Mv. ***Why do you think* sum *has such a strange value when you typed* display sum*?***

It is normal because program has not initialized sum yet so it has garbage value. Even our break point at line 10, it does not execute line 10. If I type step command it will execute line 10.

2Mvi. Now type **s** (**step**) or **n** (**next**). ***Describe what you see.***

Now *sum = 0* and because line 10 has executed however *“i”* still has garbage value.

2Mvii. ***Describe what you see. Can you see any value in using the* display *command?***

Now, both of them are equal 0, and when I use *step* command compiler shows me step by step their values.

2N. ***Under what circumstances is the message displayed? What do you think happens if you answer ‘n’ to the message?***

If I enter “*gdb ./debug4”* and put break point any line in the source code and run the program, debugger asks this question because program is running and I have a break point.

If I do not have a break point it does not ask this question because the program finishes itself s you cannot control the program.

So the criteria for this yes/no question running program and it has to have break point in it.

It I choose “n” it will not quit program and stay current position.

2O. ***Under what circumstance(s) would you want to use the* where *command? What does the* where *command show?***

Segmentation fault is good reason to use *where* command because it says you where (which line) your program has crashed. It gives entries’ information so you can find which entry has a problem.