Linux Lab 4 - gdb

Instructions for Lab 4

1. Watch the videos in Module 10, Lesson **Linux Tutorials** before you do this Lab 4. You must also complete Chapter 5 in the PCC Linux and Vim Manual to be able to do this lab.
2. Here is a Video for Lab4, that you can follow along as you do this Lab.
3. First, login to PCC Linux Server (`cslinux.pcc.edu`)
4. Make sure you are in your home directory.
5. (Your home directory should look like this: `yourname@vmcslinuxprod01:~/`)
6. If not type `cd` and hit enter to get you back to the home directory
7. From your home directory, start the `script` command - type `script lab4.txt` at the command line.
8. List the files in your current directory `ls`
9. Move into the directory for your class `cd CS161B`
10. Make a new directory for Lab4 `mkdir Lab4`
11. Change into that directory `cd Lab4`
12. Copy a file from my folder to your folder
    `cp ~gd.iyer/CS161B/samples/debug/nestedBug.cpp .`
    (Do not forget the period at the end, this tells the command to copy the file to your current folder.)
13. Type `vi nestedBug.cpp` and hit enter
14. Notice the code has a nested loop and is trying to print a tic tac board like this:

```
X  O  X
O  X  O
X  O  X
```

15. Type `g++ -Wall -g nestedBug.cpp -o nestedBug` to compile your file.
16. Type `./nestedBug` to run your code.
17. Notice the output is not what you expected, even though we did not get any compile errors.
18. Now open another putty window (or a new terminal window on the mac). On putty, right click on the white bar in the current window, and select Duplicate Session from the pull-down menu. This will allow you to login to a second window.
19. Open the code in the new window using `vi nestedBug.cpp`, and we will use the `gdb` command to debug the code, in the other window where you have your script command active.
20. Type `gdb ./nestedBug` in the window with the script to run your code using `gdb`.

21. You should have a `(gdb)` prompt. Type `b 12` to set a breakpoint on line 12.
22. Now type `r` to run your program.
23. Your program should run to line 12 and wait. Type `n` to execute the next line.
24. You should see the output line on the screen.
25. Type `n` again to execute the next line. Now you are going through the for loop.
26. Type `p i` to print the value of `i`. You can watch your variables using the `p` command and you can step through your code using `n`.
27. See if the variables `i` and `j` are doing what you expect them to do as you step through the code. Watch the Linux Lab 4 Video to see how I do this if you are not sure.
28. When you find the bug, fix the code file.
29. The next step is to compile the C++ source code file. To compile the nestedBug.cpp code type:

    ```
    g++ -Wall -g nestedBug.cpp -o nestedBug
    ```
30. To run your program type `./nestedBug`, and the output will appear directly below.
31. See if you fixed the bug.
32. Type `exit` and you are almost done with Lab 4 and you should have a `lab4.txt` file.
33. Type `ls` to list the contents of your directory. You should see `lab4.txt`
34. Follow the instructions in the Transferring Files section of your PCC Linux Manual and transfer the file to your pc and upload to D2L with your Final Project.
35. For this lab, you should submit the fixed code file `nestedBug.cpp` along with your `lab4.txt`
36. Now you are done with Lab 4.