

Physics 234: Lab 1

Tuesday, January 18, 2011 / Thursday, January 20, 2011

1. Each of the laboratory's 33 computers is configured to run both Linux and Windows. Linux is a variant of the UNIX operating system, and we will be using it exclusively as the programming environment for this course. The particular distribution is called *Fedora*. If you find that your machine is booted into Windows, please restart it and select the Linux option from the boot loader menu.
2. User accounts have been created under the names p234u1xx with xx ranging from 00 to 69. Please claim one of these userids by adding your name to the list at the front of the class.
3. Your initial password has been set to phys_234. Your first order of business after successfully logging in is to change this password to one of your own choosing. Open a terminal window by following the sub-menus

Applications ▸ Accessories ▸ Terminal

starting from the top-left corner of the screen. At the prompt, run the command yppasswd.

4. Read over the first three sections of

http://www.ualberta.ca/~kbeach/comp_phys/bash.html

Be sure you understand what a path is and why the ./ path prefix is necessary to run your own programs.

5. Now let's create a file hello.cpp that contains the "Hello World!" program. Write the program using the emacs text editor. Compile it with g++ and run the resulting executable. You should be able to reproduce the following terminal session:

```
$ emacs -nw hello.cpp
#include <iostream>
using std::cout;
using std::endl;
int main()
{
    cout << "Hello World!" << endl;
    return 0;
}
[ctrl-x][ctrl-s][ctrl-x][ctrl-c]
$ g++ -o hello hello.cpp
$ ls -F
hello*   hello.cpp
$ ./hello
Hello World!
```

You can find more information on the emacs editor here:

http://www.ualberta.ca/~kbeach/comp_phys/tools.html#emacs

Which editor you use is a matter of taste. vim and pico are popular alternatives that operate within the terminal window. gedit has a more familiar windowing interface.

6. Use the `curl` command to download some additional code you'll need for the lab.

```
$ WEBPATH=http://www.ualberta.ca/~kbeach/phys234
$ curl $WEBPATH/src/Lab1.tar.gz -O
$ tar xzf Lab1.tar.gz
$ cd Lab1
```

The file ending in `.tar.gz` is an archived and compressed directory of files (sometimes called a *tarball*). The last command above moves you into the newly created `Lab1` directory.

7. Compile and run the program `intreps`. You should see it write out the letters A through K.

```
$ ls
intreps.cpp      makefile      maxint.cpp
$ make intreps
g++ -o intreps intreps.cpp -O -ansi -pedantic -Wall
$ ls -F
intreps*        intreps.cpp   makefile
$ ./intreps
    A      A
    B      B
    C      C
    D      D
    E      E
    F      F
    G      G
    H      H
    I      I
    J      J
    K      K
```

Look at the source code in `maxint.cpp` and try to understand what the program is doing. Now make some changes. Cast the variables `a` and `b` to type `int` [using either `int(a)` or `(int)a`] as they're being written out to the terminal. Also change the initial assignment of `a` from 64 to 122.

```
$ make intreps
g++ -o intreps intreps.cpp -O -ansi -pedantic -Wall
./intreps
    123     123
    124     124
    125     125
    126     126
    127     127
   -128     128
   -127     129
   -126     130
   -125     131
   -124     132
   -123     133
```

Explain what's going on and why the two columns differ.

8. Run the program maxint.

```
$ make maxint
g++ -o maxint maxint.cpp -O -ansi -pedantic -Wall
$ ./maxint
2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
```

Extend the program to determine the largest int that can be represented on your machine.

9. Can you rewrite intreprs.cpp and maxint.cpp using a for loop?

10. If you have time, start work through the first few sections of the C++ notes, available at

http://www.ualberta.ca/~kbeach/comp_phys/cpp_basic.html

Try to reproduce the examples and solve the exercises.

11. Before you leave, make sure to close out of your session by selecting

[System](#) ▷ [Logout](#)