Outline of Lab 1 (CSE 5910)

- Obtaining a CSE account
- Linux Desktop
- Simple Linux commands
- Creating a C++ program
- Compiling and Running a C++ program
- Instruction for remove access to CSE Linux servers

Obtaining an Account

- Login as **newuser** (no password required).
- Enter the information as requested. Make sure you memorise the password you chose, as you will need it to login to your account later.
- Log out as follows
  - –move the mouse to the desktop.
  - –press and hold the right button.
  - –highlight *Quit*, and then release the button.
  - –Confirm in the next window that you do want to quite the session.
- Go to the Lab Monitor and present your photo ID
- Note that it may take up to 30 minutes before your account becomes active.

(*Linux*) Desktop

Simple Linux Commands

The **pwd** command

- Usage:
  pwd
- Description:
  Displays the current directory

Simple Linux Commands (Cont.)

The **nano** program (a file editor)

- Usage:
  nano *filename*
- Description:
  - Create a file named "filename" in your current directory and allow you to edit the file
  - If the file already exists, open the file for editing
- Example:
  nano *test.txt*
  This will create or open the file "**text.txt**" for editing.

Simple Linux Commands(Cont.)

The **man** command

- Usage:
  man *commandName*
- Description:
  Displays the user manual for the specified command. To exit the user manual, simply press the Q-key.
- Example:
  man *nano*

Simple Linux Commands (Cont.)

The **mkdir** command

- Usage:
  mkdir *dirName*
- Description:
  Creates a subdirectory with the specified name in the current directory.
- Example:
  mkdir *lab1*
  The example creates a subdirectory named "lab1".

Simple Linux Commands (Cont.)

The **cd** command

- Usage:
  cd *dirName*
- Description:
  Goes to the subdirectory named "dirName"
- Example 1:
  cd *lab1*
  Goes to the subdirectory named "**lab1"**
- Example 2:
  cd (without any argument)
  It changes the working directory to your home directory
- Example 3:
  cd ..
  Goes to the parent directory.

Simple Commands (Cont.)

The **ls** command

- **Example 1:**
  ls
  It lists the names of the files in the current directory.
- **Example 2:**
  ls **.txt*
  Lists the names of all the files in the current directory that have a ".txt" extension.
- **Example 3**:
  ls *mail*
  If the argument (i.e., mail) is a subdirectory name, lists the file names in that subdirectory (i.e., the subdirectory called "mail").

Simple Commands (Cont.)

The **rm** command

- Usauge:
  rm *filename*
- Description:
  Delete files or subdirectories
- Example 1:
  rm *test.txt*
  It deletes the file "test.txt" in the current directory.
- Example 2:
  rm –r *lab1*
  It removes directory called "lab1" and all of its contents.
  Be careful when using this command!

Simple Commands (Cont.)

The **cp** command

- Usage:
  cp *file1 file2*
- Description:
  Copy the content of ***file1*** to ***file2***. The content of ***file2*** will be the same as ***file2***.
- Example:
  cp *test.txt test_backup.txt*
  It creates a copy of "test.txt", called "test_backup.txt".

Simple Commands (Cont.)

The **mv** command

- Usage:
  mv *oldname newname*
- Description
  - If "newname" is a directory name, it **moves** the file named "oldname" into directory named "newname"
  - Otherwise, this command **renames** the file named "oldname" to "newname".
- Example:
  mv *First.cpp Second.cpp*
  The example renames file "First.cpp" to "Second.cpp".

Simple Commands (Cont.)

The **nedit** program (a full screen text editor)

- Description:
  a GUI (Graphical User Interface) style text editor for editing programs and plain-text files. It provides mouse based editing (compared to **nano**).
- Usage:
  nedit *filename*
  - If there is no file named "filename" in the current directory, it creates the file and allows you to edit it
  - If the file exists, it opens the file for editing.

Creating a C++ Program

- Login to your CSE account and open a terminal.
- Type the following command:

  nedit *First.cpp*

- Type the program shown below:

```
1  // my first program in C++
2
3  #include <iostream>
4  using namespace std;
5
6  int main ()
7  {
8    cout << "Hello World!";
9    return 0;
10 }
```

Compiling and Running a C++ program

- Type the following command to **compile** the program:

  g++ first.cpp -o first

  This converts a C++ source program into an executable program

- Type the following command to **run** the program:

  first

Remote Access to CSE Linux Servers

- You may want to write and compile your C++ programs at home
- For such a purpose, you can remotely access the CSE Linux server **red.cse.yorku.ca** from your home machine or laptop.

Running Remote Applications from Mac

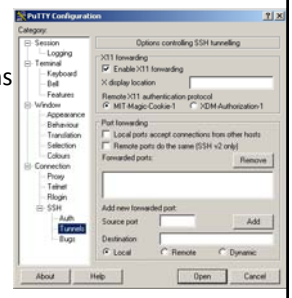
- Go to Applications –> Utilities
- Invoke Terminal or X11. A command line window will appear.
- On the command line, type
  - "ssh -X red.cse.yorku.ca"
- Login with your CSE account username and password.

Running Remote Applications from Windows (Method 1)

- SSH Client
  –Download Putty
  –http://www.chiark.greenend.org.uk/~sgtatham/putty/
- X-Server
  –Download Xming X-Server
  –http://sourceforge.net/projects/xming/
- After a short while, you will see the **X** logo in the system tray.

Running Remote Applications from Windows (Method 1 Cont.)

- Launch **Putty**
- Enable X11 forwarding under the Tunnels options

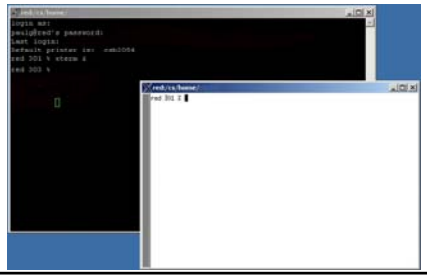


Running Remote Applications from Windows (Method 1 Cont.)

- Enter the hostname of the server you want to connect to. For example, *red.cs.yorku.ca*
- Click **Open** to connect to the remote server.
- If this is your first time connecting to the server from this workstation, you will be presented with a **Security Alert**, Click **Yes.**

Running Remote Applications from Windows (Method 1 Cont.)

- Next login and run a X-1 application. In this example we will run **xterm**.



Running Remote Applications from Windows (Method 2)

- Download and install Xming at http://sourceforge.net/projects/xming/files/Xming-mesa/6.9.0.31/
- To create an Xming connection to the CSE server,
  - Start XLaunch from the Xming program directory.
  - Select “Multiple windows”, then click “Next”.
  - Select “Start a program”, then click “Next”.
  - Select “Using PuTTY”, enter the server name **red.cse.yorku.ca** and your CSE account username, then click “Next”.
  - Click “Next” again.
  - Click “Save configuration” and save the file “CSE.xlaunch” to your desktop
- To connect to the CSE server, double-click the file “CSE.xlaunch” (shown as) on the desktop.