#### CMSC 104 – Lecture 2 by S Lupoli – adapted by C Grasso

# Introduction to Linux

# What is an Operating System?

- A layer of software that runs between the hardware and the user.
  - Controls how the CPU, memory and I/O devices work together to execute programs
  - Keeps track of all programs currently running
- Allows the user to tell it what to do
  - Controls access (login) to the computer
  - Takes commands from a user and executes them
- Often referred to as simply OS

# What is an Operating System?

- A particular OS may run on several different hardware platforms
  - Linux on PCs, Smartphones, PDAs, servers
  - Windows on PCs, Smartphones, PDAs, servers
  - Customized for each platform
- One hardware platform may run multiple OS
  - Macs can run MacOS or Windows
  - PCs can run both Linux and Windows

# How Do I Communicate With the Computer Using the OS?

- You communicate using the particular OS's user interface.
- Graphical User Interface (GUI)
  - Point and Click
- Command-driven interface
  - Type in text
  - Response is text

## **GUI vs. Command-driven**

- When you boot the GL lab computer into Linux, you can choose
  - GUI
  - Command-driven Interface
  - Both

Example of Interfaces		
GUI	Command Line	
	glumbcedu - PuTTY    Image: State of Stat	

# User Interface for CMSC 104

- Linux command line interface
  - Edit
  - Compile
  - Execute
  - Submit

## **Linux Files Overview**

- A file is a sequence of bytes.
- It can be created by
  - a text editor (Nano, VI or Notepad)
  - a computer program (such as a C program)
- It may contain a program, data, a document, or other information.
- Files that contain other files are called directories (sometimes called folders).

## **Linux Filenames**

- For this class, you should stick with filenames that contain only
  - letters (upper or lower case)
  - digits
  - underscore ( \_ ) or hypen (-)
  - No spaces!
- Up to 255 characters
- Filenames are case sensitive
- Some examples: firefox.exe, things2do.txt, dinner\_menu.pdf

## **Linux Directories**

- Directories contain
  - files
  - other directories called subdirectories.
  - may be empty.
- Directories are organized in a hierarchical fashion in a tree-like structure
- They help us to keep our files organized.

# **Example Directory Tree**

#### How can you tell which are files and which are directories?



# **Directory Terms**

- Your home directory is where you are located when you log in
  - (e.g., /afs/umbc.edu/users/j/d/jdoe28/home/).
- The current directory is where you are located at any time while you are using the system.
- The / (pronounced "slash") is the root directory in Linux.
- Files within the same directory must be given unique names.
- Each running program has a current directory and all filenames are implicitly assumed to start with the name of that directory unless they begin with a slash.

# **Directory Terms**

#### Paths

- allow us to give the same name to different files located in different directories.
- Absolute Path
  - The absolute path is a path that contains the root directory and all other subdirectories you need to access the file
  - It points to the same location in the directory tree regardless of the current working directory
- Relative Path
  - The relative path is a partial path to a file in relation to the current working directory
  - If inside of the home directory in the previous directory example, a relative path would be

### Absolute



Relative



## **Subdirectories**

Are used for organizing your files



# **Frequently Used Linux Commands**

Command	Description
cd directory	make <i>directory</i> the current working directory
ls	lists the contents of the current working directory
cp sourcefile destinationfile	copy the contents of <i>sourcefile</i> into a new file named
	destinationfile
rm filename	delete the file named <i>filename</i>
mv sourcefile destinationfile	rename the file named sourcefile to destinationfile
cd	move down a directory
man <i>command</i>	shows a manual on how to use the command word (hit 'q'
	to quit)
mkdir <i>directory</i>	creates the directory named of your choosing
rmdir directory	deletes the PRE-EXISTING directory named
lpr <i>filename</i>	prints file

### **Class exercise:**

- In your home directory,
  Make a subdirectory for CMSC104
  - In the CMSC104 directory
    - Make subdirectories for each homework (12)
    - Make subdirectories for each project (2 projects)



# **Changing the Current Directory**

- . (**dot**) is the current directory.
- . . (dot-dot) is the parent directory.
- Use the Linux command cd to change directories.
  - Use dot-dot to move one directory up in the tree
  - Use the subdirectory name to move down.
  - Use the complete (absolute) directory name (path name) to move anywhere.
- Practice
  - Change your current directory to hw1/
  - Change your current directory to project1/

## **Wildcard Characters**

- You will find wildcard characters useful when manipulating files
  - listing or moving them
- The wildcard characters are \* and ?
  - ? is used to represent any single character.
  - \* is used to represent o or more characters.

# **Practice – Wildcard characters**

- ? any single character.
- \* 0 or more characters
- Given the files in the directory below, what will be the output from the following commands?
  - Is hw\*.txt
  - Is hw?.txt

hw1.txthw123.txtcmsc104.txthw2.txthw\_assignment.txtcmsc201.ppthw.txtlupoli.txt

# The Andrew File System (AFS)

- AFS is network file system.
  - AFS is a way of keeping track of your files over the network.
- Features
  - All of the lab computers: Windows, Linux, IRIX, and Mac, are capable of reading this file system.
  - The files that you maintain in your account will be accessible to you over the network from any of these computers.
  - No need to store your information on a flash drive and carry it between machines.

## Your Directory on AFS

- Every GL account is assigned a directory on AFS, which contains the home directory.
- On all of the UNIX systems here at UMBC it will be located at the following location: /afs/umbc.edu/users/u/s/username/
  - *u* is the first letter of your username,
  - s is the second letter of your username,
  - and username is your GL account logon name.

# **Creating a text file in Linux**

- Command line editors in Linux
  - vi
  - emacs
  - nano
    - easiest and resembles notepad
- Can be used with PuTTY

# **Create Your First C Program**

- To create the file and start the editor nano hello.c
- Type #include <stdio.h> main() { printf ("Hello World! \n"); }
- To save the file
  Press Cntl-X
  Press Y
  Press Enter

### **Compile & Run Your First C Program**

- Compile your program
  gcc -o hello hello.c
- Run your program
  ./hello
- Output should be:
  Hello World !

# PuTTY @ UMBC

- You can connect to the GL machines from your personal machine using PuTTY
  - When you connect to GL through Putty, you are using only the command-driven Interface.
- How to install and run PuTTY on Windows
  - <u>http://userpages.umbc.edu/~dhood2/courses/misc/springo8/</u> working-from-home/putty.html
- PuTTY download page
  - <u>http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html</u>