

Computing Resources for CS130

This handout is a summary of information to aid those students using the Linux Operating System needed in CSCI 130.

Networking Domains

There are two networking domains at the University.

1. The “EUP” networking domain is controlled by Department of Technology and Communications (T&C). This is the domain normally encountered on the EUP campus (Ross Hall, Hamilton Hall, the Library, etc.)
2. The “CS” domain is controlled by the Math & CS Department. This is the domain you encounter in the Math/CS labs in Doucette and Dundee Lab (located in Ross Hall across from the general student terminal lab).

Different software and services are available in these domains. More importantly, although your user name (login ID) is the same, passwords are administered separately in each domain. If you cannot reach your instructor and need help, contact the CS domain administrator Jason Patalon (jpatalon@edinboro.edu) and he will try to help you.

User Name (login ID)

Your user name is your first initial followed by the last six digits of your university id followed by your last initial. For example, for the username Chris Tang with id @01023664 is c023664t. This user name is the same on T&C computers as well as Math/CS ones.

Passwords

Your password is probably initially set to *mmdyy*, the month, day, and year you were born. For example, the date November 6, 1974 is entered as 110674. This password is exactly six digits long. Note that single digit months and days must start with a zero.

The passwords for a T&C computer are usually different than for Math/CS computers. If you change the password for T&C, it does not change the password for Math/CS, and vice-a-versa.

Operating Systems

You will be using the Math/CS labs (primarily in Dundee Lab located in Ross Hall or Doucette Hall 222) for this course. You will encounter two Operating Systems (described below): Microsoft™ Windows® in Dundee and Linux in Doucette. For this course, you will be using the Linux Operating System, but you will access it through Windows from Dundee Lab most of the time. You are encouraged to use Doucette Hall 222 and learn how to use the GUI tools there for assignments.

Linux OS is an open source operating system that is patterned after the Bell Labs UNIX Operating System. There are several operating systems patterned after UNIX, including SUN SOLARIS™ and Berkeley BSD UNIX OSs. Industries engaged in scientific research and development typically use one of these variants of UNIX.

Accessing Linux OS in Doucette 222

Logging On to Linux

Enter your username (login ID) as described above. Then enter your password.

Logging Off

This is done by entering “exit” to a command line prompt, or the close button if using a GUI window.

Accessing Linux from Windows in most EUP labs and Dundee Lab

Linux can be accessed from Windows using the “putty” icon and the secure shell protocol. It can also be accessed using a command window on campus machines using the secure shell command “ssh.” Mostly you will be using “putty” (which is available as a free download, see below).

The SSH Method (secure Shell) on a campus machine

1. From a command window type “ssh cslab100.cs.edinboro.edu”
2. You will be prompted for your password if the user name is recognized in the handshaking process.

From your own PC

This procedure can be used from your remote PC running either Linux or Windows. To do this you need a copy of “putty.exe” which can be downloaded from the author’s site at:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Place the download in your C:\Windows folder and make a shortcut to “putty.exe.” In the setup encountered when you first execute putty, indicate the destination is:

cslab100.cs.edinboro.edu. You can save the setup information in the dialog box under your own setup identifier.

There is more information about remote access on the CS wiki:

<http://cswiki.cs.edinboro.edu/>

To Exit putty or the ssh: type “exit” to the shell command line prompt.

Operating in the Linux Environment

If you are using Linux in Doucette 222 you will have a GUI interface (most of the time you will be using a “Terminal” emulator from “putty”). To access the terminal emulator in Doucette 222 machines, left click on the “Applications” icon in the lower left of the desktop, select “systems tools” and select “Terminal.”

Once you have the terminal window, you should see a prompt similar to “[c023664t@localhost~]\$” (with your user login in place of “c023664t”). This is your home directory. If you issue the command “pwd” you should see: **/home/<your login id>**. The “pwd” command displays (**p**rints) your **w**orking **d**irectory (not to the printer, but to the terminal window). Directories are introduced below.

File Names

File names can be long. Generally, though, avoid spaces and punctuation symbols (except the underscore or hyphen). Often it is useful to put an extension on a file name. Typical file extensions include:

1. <name>.cpp the extension for a “C++” source file
2. <name>.txt the extension for a plain text file
3. <name>.dat an extension that can be used to indicate a data file
4. <name>.scr an extension that is used for a file that contains a shell output script
5. <name>.sh or .ksh extensions that indicate the file contains a shell input script
6. <name>.h the extension used for “C” and “C++” header files

Directories

Files are stored in directories (what are called “Folders” in Windows). When you log on, you are automatically in your home directory. Directories may contain directories and/or files, thus forming a hierarchical file system.

Linux Commands

Listed below are often-used Linux commands. There are variations of these commands—just simple uses are introduced in this document. As you gain experience or ask the right questions more details and capabilities will be presented.

Case sensitivity

Be aware that Linux is case sensitive. For example, “cd” is a recognized command for “Change Directory,” but neither “Cd” nor “CD” is recognized as such.

Table 1: Linux Bourne Shell Quick Reference Sheet

I want to:	Linux Shell Command	Descriptive Info
<u>Concatenate a file to the terminal (list its contents)</u>	cat <filename>	
	cat program.cpp	The contents of the file “program.cpp” are listed on the terminal window. You do not want to “cat” non-text files like “a.out” or files ending in “.o” or executable files.
<u>Change Directories</u>	cd	
To a child directory (down one level)	cd < name>	Changes to the child (sub directory) named “name” within the current directory (if it exists there).
To a parent directory (up one level)	cd ..	The “..” (dot dot), moves up one level in the directory structure (e.g. to the parent of the current directory).
To my home directory	cd	by itself moves to your home directory.
<u>Copy a file on the local PC</u>	cp <source file name> <destination file name>	
Example: Copy a current file version to a new version (on the same machine).	cp prog1v1.cpp prog1v2.cpp	Makes a copy of prog1v1.cpp (the input source) and names the copy prog1v2.cpp (the output destination). This is a good way of making a backup copy before making changes to a file.
Copy a file to or from a remote PC Remote is Linux	When Windows PC is Local	
	Recommend you use “FileZilla”	See http://cswiki.cs.edinboro.edu/ for instructions on how to download and install FileZilla which can be used to drag and drop copy files between Windows and Linux. It uses SFTP as the secure protocol.
	When Linux PC is local	
scp – Secure Copy	See http://cswiki.cs.edinboro.edu/ for more information on scp (available in SSH).	

I want to:	Linux Shell Command	Descriptive Info
<p><u>Execute Other Applications</u></p> <p>(the student is directed to the Edinboro University Computer Science Wiki for more information and help details: http://cswiki.cs.edinboro.edu/)</p>	<p>{<path>}\$< name> - find and execute the application “name”</p>	
	<p>Execute an application named “name”</p>	
	<p>\$< name></p>	<p>Linux searches a set of directory paths for an application file named “name” and executes it if it is found. The default search path does not include your current directory (see “./<my_program>” below.</p>
	<p>Execute the “vim” text and C++ program editor</p>	
	<p>\$vim</p>	<p>“vim” is the text editor application we will use in this course. See “vim” Quick Reference Guide (download)!</p>
	<p>Make a C++ executable using the “make” utility</p>	
	<p>\$make <name></p>	<p>The default action of “make” is to execute a sequence of dependency checks and commands to see if an executable file “name” is in sync with a file name “name.cpp.” If not, the C++ preprocessor, compiler, and loader are invoked to produce a new executable from the “name.cpp” file.</p>
	<p>Execute my executable program in the current directory</p>	
	<p>\$./<my_program></p>	<p>The “./” (dot slash) means “look in this directory” for an executable file named “my_program” and execute it (if it is an executable file). Note: File types and permissions are not covered in this quick reference.</p>
	<p>Capture the output of an application or set of applications</p>	
<p>\$script <filename.scr> \$<application> \$exit</p>	<p>Capture terminal output produced by an application (e.g. your program) and save it in a file named “filename.scr”</p>	
<p>\$ssh</p>		
<p><u>Get Help on a specific shell Command</u></p>	<p>man <command name or acronym></p>	

I want to:	Linux Shell Command	Descriptive Info
Example: To display a manual (help) page for the particular command acronym “cat”	man cat	Displays a manual page, in text format, on the “cat” command. When the manual page is greater than the default page length, the shell waits for you to press the space bar before displaying the next page. If or when you are finished reading and there is more text indicated by the “man” prompt, type the character ‘q’ to quit the display of the manual page.
List all Linux Shell Commands	help	Lists available Linux commands.
List (or display) any Print Jobs on the printer queue	lpq	
	lpq	Lists any print jobs in the printer queue.
List all the directories and files in the current directory	ls [-l]	
Example: display <u>only</u> the <u>names</u> of all directories and files in the current directory.	ls	Suppose there were three files in the current directory named: “abc.h” “abc.cpp” and “abc.txt.” The output from the “ls” command would be: abc.cpp abc.h abc.txt
Example: <u>display the properties</u> of all the directories and files in the current directory.	ls -l	The command is lower case, but the example uses upper case to make the letters clear, “LS minus L,” lists the directory and file names, sizes, dates, and permissions: abc.cpp <list of file properties> abc.h <list of file properties> abc.txt <list of file properties>
Make (or Create) a new Directory	mkdir <directory name>	
Create a new child directory:	mkdir Release2	Creates a new child directory named “Release2” in the current directory
Create a new sibling directory:	mkdir ../Project3	Suppose you are in a current directory named “Project2.” The shell goes to the parent (“..”) of the current directory and creates a sibling (same level) directory named “Project3.” Now, both “Project2” and “Project3” are in the parent directory.

I want to:	Linux Shell Command	Descriptive Info
<u>Print the current directory path</u> (Where I am in the file system)	pwd – Print Working Directory (really means list)	
	pwd	Displays the full path of the current directory from the “home” directory in the current window(it does not send anything to a printer).
<u>Print a file</u>	print<Location name> (you will be asked for your Windows password)	
	printDundee <i>filename.cpp</i>	“filename” output should be on the printer in the Dundee lab.
	printDoucette <i>filename.cpp</i>	“filename” output should be on the printer in the Doucette lab.
	Notes:	<ul style="list-style-type: none"> • Always review the file you want to print on the screen to make sure it is a text file: e.g. *.cpp, *.txt, *.dat, *.scr. • Do Not Print non-text files like *.out (like a.out), *.exe! These will ruin your day!
<u>Remove (or delete) a directory</u> Example: To remove a child directory named “Failed_Exams” in the current directory	rmdir <directory name>	
	rmdir Failed_Exams	The child directory “Failed_Exams” is deleted if it does not contain any files or child directories of its own. If it does, you first have to delete any files and child directories before attempting to delete the directory.
<u>Remove (delete) a file</u> Example: you want to remove (delete) a file named “temp.txt”	rm <filename>	
	rm temp.txt	Deletes the file named “temp.txt” if it is in the current directory. Be sure you really want to delete the file, though. There is no “recycle bin” or backup copy unless you made one.
<u>Rename (or move) a file</u>	mv <old file name> <new file name>	
Example: rename file “Project1.cpp” to “old_Project1.cpp”	mv Project1.cpp old_Project1.cpp	The shell command “mv” (move) simply renames the file.
Example: move file “Project1.cpp” to “Release1/Project1.cpp”	mv Project1.cpp Release1/Project1.cpp	The command “mv” was originally called “move” because it is an easy way to put the file into a different directory; in this case to the child directory named “Release1”

I want to:	Linux Shell Command	Descriptive Info
Setup and Connect a Secure Session on a remote Linux Server Local PC is Windows	ssh cslab100.cs.edinboro.edu login: password:	The ssh command provides a secure command line window to a remote Linux server.
To obtain and install a ssh client on your Windows PC	Download and install a copy of “Putty” by visiting the website below:	
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html		