

CPE 212 - Fundamentals of Software Engineering

...

Linux Review

Linux:

An operating system, which is a collection of programs, that is open-sourced. The basis of Android and is used in data centers throughout the world.

Physical Access to Linux Workstations

- Use workstations in EB 216 and EB 246
- Doors to both of these labs are locked, but you can access the labs using your Charger Card as long as the front door to the Engineering Building is open

IMPORTANT NOTE

Access to EB 246 is restricted during officially scheduled laboratory sections appearing on the UAHuntsville Course Schedule. Please do not enter EB 246 during one of these reserved times without permission of the laboratory instructor. EB 216 is available to you at any time.

Requesting an Account

- If you are enrolled in CPE 112, 212, 353 or 453, then your account is typically generated automatically
- Inactive accounts from previous terms are disabled or deleted
- If you need to request a new account, go to <http://labs.ece.uah.edu/>
- This may require help from a TA or an instructor so please let me know if this is the case

Accessing Your Account

- The easiest way to access your account is from one of the workstations in EB 216 or EB 246
- It is possible to access your account remotely, but it will require additional work on your part to familiarize yourself with the required software
- Your home directory (containing all your files) is shared and accessible from the Blackhawk Linux system

Login Information

Username

Students new to the College of Engineering use your campus email username

Example:

xyz007@uah.edu

You will use xyz007 as your username

Password

first initial + last initial + last 6 digits of student ID (SID)

Initials are lower case, SID, is the student's A number not their Social Security Number.

WARNING - The first time you login, you MAY be required to change your password and re-login using that password.

IMPORTANT NOTE

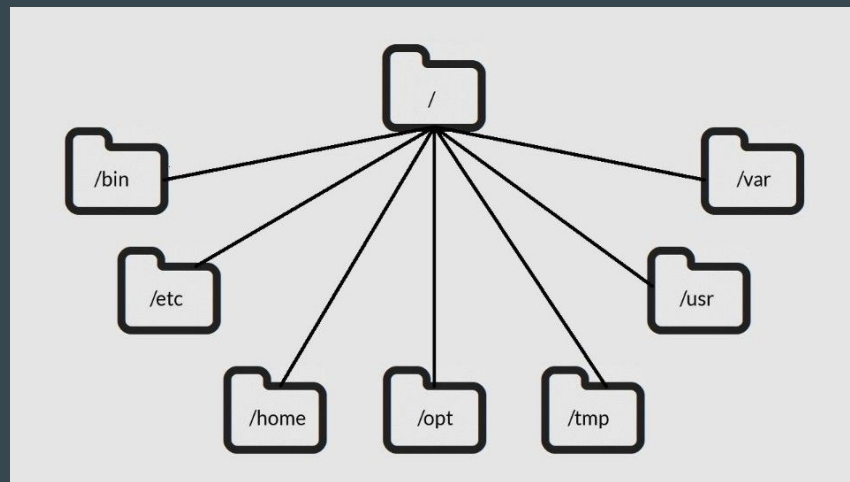
If you have forgotten your password please go to the <http://ece.uah.edu> website to reset your password. You can also visit Jason Winningham in EB237 with your student ID.

Linux

Commands and Terminology

Organizing Your Work

- With Microsoft Windows, files are organized using nested folders
- In UNIX and Linux, a folder is known as a directory
- A directory may contain files or nested directories (subdirectories)
- When you first open a terminal window, you will be in your home directory, the location where your files will be stored



Naming

Filenames and directory names in UNIX/Linux are Case Sensitive

data.txt
Data.txt
data.TXT } (All different files)

Linux systems prefer NO SPACES within a filename or directory name.

“Project10” or “Project_10” instead of “Project 10”

File extensions may be appended to the end of a name to denote the type of file

Project10.cpp (a C++ source code file)
P10_input1.txt (a text input file for Project 10)

Basic Commands

Print Working Directory (pwd)

- Prints the name of the current working directory
- If you have just opened a terminal window, you will be in your home directory

```
mars$ pwd
/home/student/hjs0001
```

List Directory Contents (ls)

Lists the contents of the current working directory. Can be used with a folder name to show the contents of the folder.

```
mars$ pwd
/home/student/hjs0001
mars$ ls
somefile  anotherfile  project1
project2
mars$ ls project2
project2.cpp  p2_input.txt
```

Advanced uses of ls

- To see additional information about the contents of a directory, type `ls -l` to see the longer listing
 - Lines that start with a `d` indicate a directory
 - Lines that start with a `-` indicate a file
 - Lines that start with a `l` indicate a link
 - You will also see the file permissions, the file owner, file size, and last date of modification along with the name of the file or directory
- To see the complete listing of the contents of the current directory, type `ls -a`
 - Lists all files and directories within the current working directory including hidden files
 - You will see `.` and `..` and perhaps some other hidden files in addition to any visible files and directories
 - `.` refers to the current working directory
 - `..` refers to the parent of the current working directory
- To see additional information about the contents of a directory, type `ls -al`
 - Lists additional information about all files and directories within the current working directory including hidden files
 - Combines the `-a` and `-l` options

Basic Commands

Create a directory (mkdir)

- Creates a new directory within the current working directory
- For example, to create a directory named project01 in the current working directory

```
mars$ pwd
/home/student/hjs0001
mars$ mkdir example
mars$ ls
somefile    anotherfile  project1
project2    example
mars$ ls example/

mars$
```

Change Directory (cd)

- When used alone, cd switches the current working directory back to your home directory
- When used with a valid directory name, cd switches the current working directory to the specified directory

```
mars$ pwd
/home/student/hjs0001
mars$ cd project2
mars$ pwd
/home/student/hjs0001/project2
mars$ ls
project2.cpp  p2_input.txt
```

- To navigate back to a previous directory you use the “..” notation

```
mars$ pwd
/home/student/hjs0001
mars$ cd project2
mars$ pwd
/home/student/hjs0001/project2
mars$ ls
project2.cpp  p2_input.txt
mars$ cd ..
mars$ pwd
/home/student/hjs0001
```

Copy Files (cp)

- Copies the source file to the specified destination file
- For example, to make a backup copy of project1.cpp in the current working directory

```
mars$ pwd
/home/student/hjs0001
mars$ cp project1.cpp project1_backup.cpp
```



- To copy a file from the current working directory (project1) to another directory (project2)

```
mars$ pwd
/home/student/hjs0001
mars$ ls
somefile  anotherfile  project1
project2
mars$ cd project1
mars$ ls
project1.cpp  p1_input.txt
mars$ cp project1.cpp
../project2/project2.cpp
mars$ cd ../project2
mars$ pwd
/home/student/hjs0001/project2
mars$ ls
project2.cpp
```

Remove File (rm)

- Removes a specified file
- To remove a folder or directory you must specify the “-r” flag
- To avoid the prompt if you want to delete an item you can use the “-f” flag

```
mars$ pwd
/home/student/hjs0001
mars$ mkdir example
mars$ ls
somefile    anotherfile    project1
project2    example
mars$ rm somefile
mars$ ls
anotherfile    project1    project2
example
```

Manual Page (man)

- Displays the manual page for the specified Linux command
- To advance line-by-line through the manual page press the “return” button
- To advance page-by-page through the manual page press the “space” button
- To get out of the manual page press the “q” button

```
mars$ man pwd
```

```
NAME
```

```
pwd -- return working directory name
```

```
SYNOPSIS
```

```
pwd [-L | -P]
```

```
DESCRIPTION
```

The pwd utility writes the absolute pathname of the current working directory to the standard output.

Some shells may provide a builtin pwd command which is similar or identical to this utility. Consult the builtin(1) manual page.

The options are as follows:

-L Display the logical current working directory.

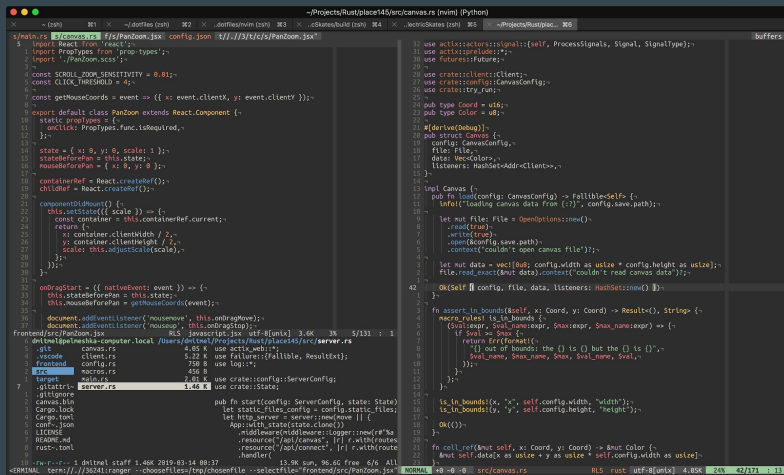
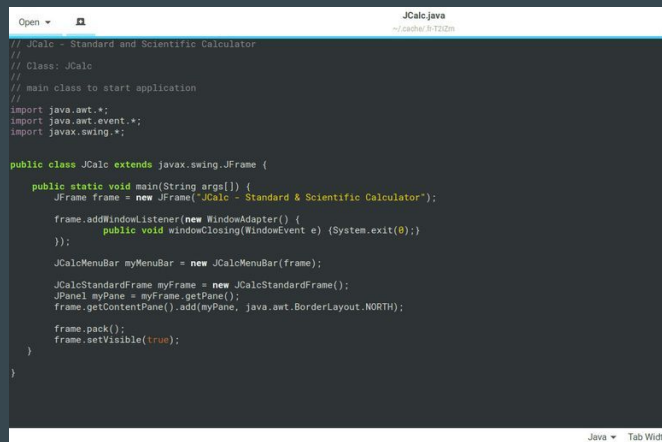
-P Display the physical current working directory (all symbolic links resolved).

If no options are specified, the -L option is assumed.

Editing and Building

Editing Programs

- Linux text editors
 - gedit - graphical editor
 - vim - terminal editor
 - Visual Studio Code (maybe)
- Regardless of the editor you use make sure to enable frequent automatic backups to provide a means of recovery in the event that the text editor crashes



Basic Compilation

- The compiler on the Blackhawk Linux system is “g++”

```
mars$ g++  
/home/student/hjs0001  
mars$ cp project1.cpp -o project1
```

Source File Output Flag Executable File

- For compiling and linking multi-file programs use see the make Tutorial

Editor Files

- On Blackhawk, `gedit` appends a `~` to the end of the name of the backup copy of the file

```
mars$ gedit project10.cpp
mars$ ls
Project10.cpp project10.cpp~
```

IMPORTANT NOTE

Be sure you are submitting the correct version of the file for grading. If you submit the project file with the `~` you will not receive credit.

Input/Output Redirection

- The Standard Input Device (stdin) is normally the keyboard
- The Standard Output Device (stdout) is normally the monitor
- With UNIX/Linux, one can redirect stdin inputs to come from a specified file or redirect stdout outputs to be written to specified file

```
// The code below attempts to input an
// integer from stdin and write it to stdout
int  someInt;
cin >> someInt;
cout << someInt;
```

Proceed with Caution!
There are no undo
commands when using
UNIX or Linux