CS 35L: Software Construction Lab

Shrey Agarwal

Lecture 1.2

Update: OH – Tuesday 9:30am to 11:30am

Honestly…

I am teaching this course for the first time…

- What does it mean?

It means that I am learning everything with you guys!

- How does this benefit you?

You get to challenge me at every step. DO NOT BLINDLY TRUST ME ON ANYTHING. :P

Open your laptops/terminals and test it out. Correct me if you think I made a mistake.

The Basics: find

- -type: type of a file (e.g,, directory, symbolic link)
- -perm: permission of a file
- -name: name of a file
- -ls: list current file

find contd…

- ?: matches any single character in a filename
- *: matches ~~one~~ **zero** or more characters in a filename
- []*:* matches any one of the characters between the brackets. Use ‘-’ to separate a range of consecutive characters.
- Examples:
  - find . -name my*
  - find . -name my* -type f
  - find / -type f -name myfile

The Basics: Redirection

- > *file*: write stdout to a file
- >> *file*: append stdout to a file
- < *file*: use contents of a file as stdin

The Basics: Dealing with Files (Mini-task)

- Create two files

```
$ touch foo.txt                    $ touch bar.txt
```

- Add text into the files and print it

```
echo "Cat" > foo.txt
echo "Dog" > bar.txt
cat foo.txt bar.txt (concatenates the output)
```

Output:

```
Cat
Dog
```

diff command

- A file comparison utility that outputs the differences between two files.
- Usage:
  - diff file1 file2

Process: ps and kill

- Process
  - An instance of a computer program in execution
- ps
  - List processes that are currently running
- kill
  - Terminate a certain process
  - Usage
    - kill PID

Daemon

- A process that runs in the background
- Example: cron
    - Enables users to schedule jobs to run periodically at certain times (cron jobs)
    - Usage: Full Backup every month

wget command

- A computer program that retrieves content from web servers
- Usage
  - wget <URL>

The Basics: Look These Up

- **head**
- **tail**
- **du**
- **top**
- ps
- **kill**
- **diff**
- cmp
- **echo**
- **find**
- **wc**
- sort
- **ping 8.8.8.8**
- **diff**
- **wget <url>**

Text editors

Emacs and vim

vi editor

- Open a file - vi <filename> or vim <filename>
- Close a file - :q
- Save a file - :w
- Save and close a file - :wq

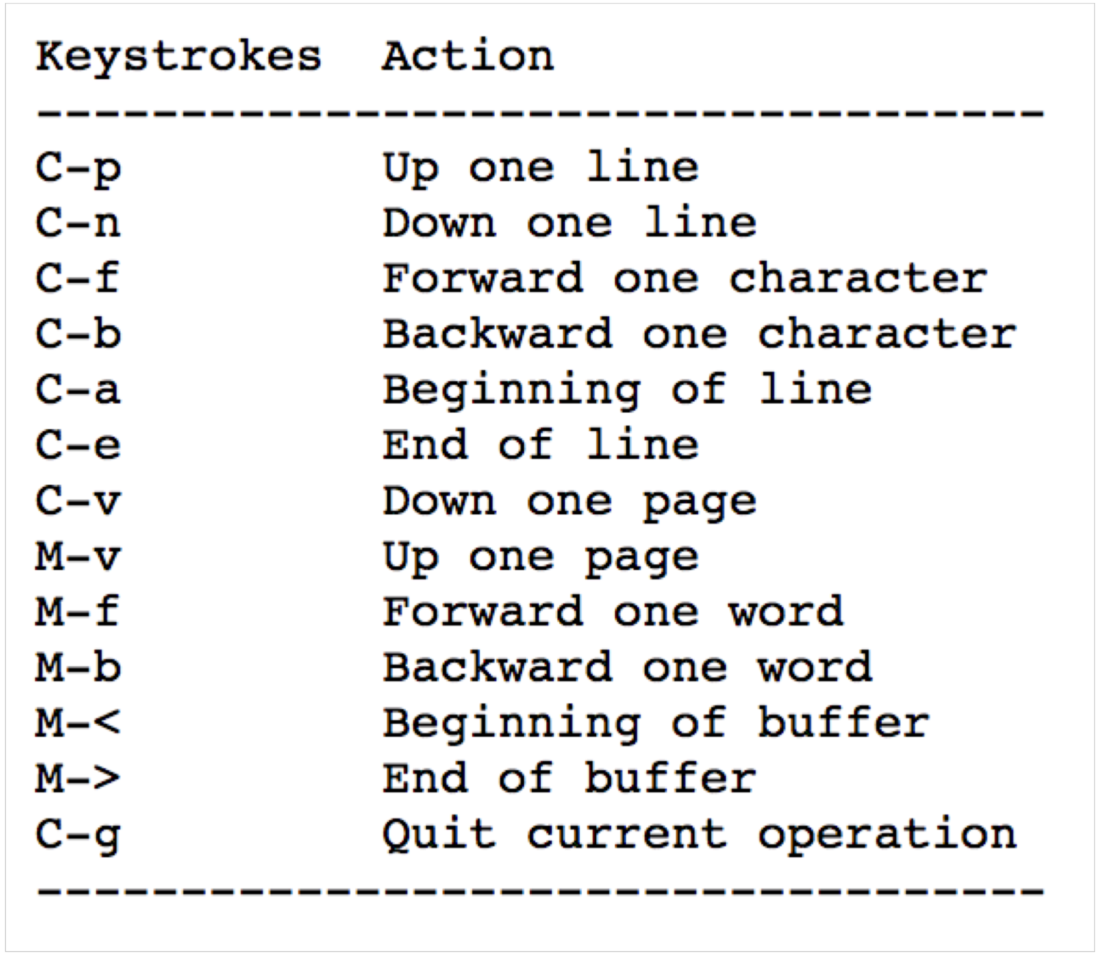
Emacs editor (We will be using Emacs for this course)

- Almost like a Windows text editor, but much more powerful
- Run emacs on the linux server
  - C-h r (manual) and C-h t (tutorial)
- All emacs commands start with ‘C’ or ‘M’
  - ‘C’ = ctrl; ‘M’ = alt (windows)/ option (Mac)
- Start emacs
  - emacs <filename>
- Exit emacs
  - C-x C-c

Basic emacs editing

- Insert text by simply typing it
- Undo by typing C-x u
- Save changes by typing C-x C-s
- Copy, cut, paste
    - C-space (starts selecting region)
    - M-w (copy a region)
    - C-w (cuts a region)
    - C-k (kill a line)
    - C-y (yank/paste)

Moving around



```
Keystrokes  Action
-----------------------------------
C-p         Up one line
C-n         Down one line
C-f         Forward one character
C-b         Backward one character
C-a         Beginning of line
C-e         End of line
C-v         Down one page
M-v         Up one page
M-f         Forward one word
M-b         Backward one word
M-<         Beginning of buffer
M->         End of buffer
C-g         Quit current operation
-----------------------------------
```

More emacs commands

- Search – C-s
- Replace – M-%
- Accessing menu – F10
- Switch buffer – C-x b
- Switch current window – C-x o
- Kill the current window – C-x 0 (zero)
- Help – C-h

Directory edit (dired) (C-x d)

- Creates an Emacs buffer containing list of directory contents
- Allows you to operate on files
- Allows you to navigate filesystem
- + - new directory, C-x C-f new file in directory, g - refresh dired buffer
- ! - run shell command
- https://www.gnu.org/software/emacs/refcards/pdf/dired-ref.pdf

EMACS tutorial links

- http://bit.ly/2CQy3H8 (some basic commands)
- http://stanford.io/2CTWNyl

Todo

- Read the tutorial and manual
- Navigate and get used to emacs in the next 15 min
- We’ll have a small task after this

Task 1

- Create a cpp file (print “hello”) using emacs
- Run the file
- Edit the file (print “hello again”) using emacs and save it as a new file
- Run the new file

Cpp program

```
#include<iostream>
using namespace std;
int main(){
    cout<< “Hello”;
    return 0;
}
```

Compiling instr:

```
g++ -o filename filename.cpp
./filename
```

Task 2

- Create 2 txt files (insert some lines) using emacs
- Find the difference between both of them using a linux command

Lab 1: Hints

Hints for lab questions:

1. man man
2. which
3. find
4. readlink
5. man chmod
6. man find
7. find
8. whereis, man find
9. find, sort
10. localedef

Assignment submission specifics

Submission details

- Go through Homework 1 again on CCLE, there may be minor changes in the wording of the assignment (Due: Oct 6th)
- Submit on CCLE under your Lab <number> (if specified)
- No submissions will be accepted via email
- Test your files on seasnet before submitting
- key1.txt should record keystrokes of Homework
- ans1.txt should have keystrokes and answers of the lab assignment

[I hope the lateness penalty policy is clear]

Assignment 7 requirements

You will have to buy a BeagleBone Green which will be required for Assignment 7.

Assignment 1: Example key1.txt

key1.txt is for HOMEWORK section

Exercise 1.1

1. C-s H E L L O W O R L D
2. …
3. …

Exercise 1.2

1. C-s H T M L
2. …

so on…

Assignment 1: Example ans1.txt

ans1.txt is specifically for LABORATORY section

- 1. Here is the answer to question 1
- 2. Here is the answer to question 2
- 3. Here is the answer to question 3
- …..

Rest of the guidelines are on the link

Plagiarism Policy

- Any instance of cheating is intolerable – serious consequences
- You are requested to do the assignments on your own and not refer to any public forums or github
- Do not put your assignments on a public platform

Tasks

1. Create a file using emacs. Try the Cut/Copy/Paste commands.
2. Write a command to find the hidden files in your home directory.
3. Remove a non-empty directory.
4. Assign the value 10 to a variable x, 15 to a variable y; Compute and display their addition in the shell. Do not use any editor!
5. Create a directory ‘test’ and ‘subtest’ inside test (in the home directory) without using mkdir twice. (*mkdir –p test/subtest*)
    1. Create empty files ans.txt, ans.html, assign.txt, assign.html
    2. Find all directories and subdirectories in your home directory
        1. *find . –type d*
    3. Find all files in the test directory starting with ‘a’ (find ./test –type f –name ‘a*’)
    4. Find all text files inside test directory
        1. find ./test –type f –name ‘*.txt’
    5. Find all html files inside test directory
        1. Find ./test –type f –name ‘*.html’
    6. Delete test directory from the home directory (*rm –rf <dirname>*)