CS35L Lab4

Spring 2017

Introduction about myself

Sophia Yan

- Master student in CS @ UCLA
- B.S. in Math & CS @ UIUC

-TA for CS35L Lab4

-sophiaxueting@gmail.com

-Office hour: TBA

Introduction to the course

- 2 unit course, lab-oriented
- Instructor Prof. Paul Eggert
- Prerequisite CS 31
- Course website
  http://web.cs.ucla.edu/classes/spring17/cs35L/
- Piazza (https://piazza.com)
- Use SEASnet account to log in server
- No attendance taken
- Discussion is encouraged, but final work should be done individually

- **Grading**
  - **Assignments - 50%** (equally weighted)
  - **Final exam - 50%**
- 10 assignments
  - 9 regular assignments + 1 presentation
- Regular assignments
  - Lab exercises (expected to be done in the lab)
  - Homework
  - Due every week Friday 11:55pm
- Presentation (Assignment 10)
  - Will distribute sign up sheet
  - 10 mins presentation + research report
  - Due the end of week 10

Week 1	Introduction to linux
Week 2	Shell scripting
Week 3	Modifying and rewriting softwares (Python)
Week 4	C programming and debugging
Week 5	System call programming and debugging (with C)
Week 6	Multithreaded performance (with C)
Week 7	SSH
Week 8	Dynamic linking
Week 9	Change management (git)
Week 10	Research presentation

- All assignments to be done individually
- Submitted on CCLE
- Lateness penalty (with some exceptions)
  - 2^N % deduction for being up to N days late
  - Exception 1: Last assignment must be submitted on time
  - Exception 2: Not accepting submissions after last day of instruction
  - Other exceptions need approval from Prof. Eggert

- Final
  - 11:30 AM - 2:30 PM, Thursday, June 15, 2017
  - Different cross sections
  - Open-book, open-notes, no electronic device
  - All materials will be covered, including concept, programming and presentations

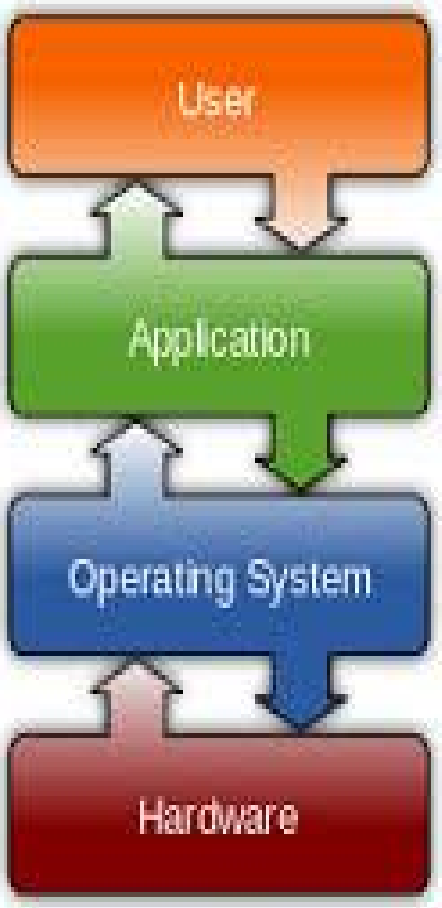
Introduction to Linux

Week 1

What is Linux?

- Operating system !
- Created by Linus and a group of people (online)
- Unix-like open source software
- Free to contribute, free to use

- Four Components (linux distribution)
  - Linux kernel
  - GNU utilities
  - Graphical desktop environment
  - Application software



Linux Kernel

- Four main functionalities
  - System memory management
  - Software program management
  - Hardware management
  - Filesystem management

GNU utilities

- System utilities to run on linux kernel
- Contains coreutils package
  - Handling files
  - Manipulating text
  - Managing text
- Shell is a special interactive utility (CLI)
  - Bash is the default shell in Linux

Graphical desktop environment

- Two common graphical environment
  - KDE
  - GNOME desktop

A Brief History of Operating Systems

- The Dark Ages
  - No OS until 1960s
  - Manually loaded programs
  - Reboot after each program
- Batch OS
  - Unified application development across systems
  - Output via printer, later via monitor
  - I/O via magnetic tape or disk
  - Written in assembler (e.g., OS/360)
  - Multiprocess

A Brief History of Operating Systems

- Timesharing OS
  - Multiuser
  - Multics (1964)
    - Segmented memory
    - Paged virtual memory
    - Applications written in many languages
    - Shared multiprocess memory
- Personal Computer
  - Single machine for single user
  - OS must manage screen and input devices
  - Window, Icon, Menu, Pointing Device (WIMP, e.g., MacOS, 1984)
- Cutting-Edge OS
  - High performance computer (HPC) clusters (e.g., BlueGene/L at LLNL rated at 280.6 teraFLOPS)
  - Cell phones, video
  - Video games
  - Browsers

GUI – Graphic User Interface

- Human-computer interface using graphic icons and visual indicators
- Intuitive
- Limited Control
- Easy multitasking
- Limited by pointing
- Bulky remote access
- Example GUI in Linux : X (Windows), Gnome, KDE (Linux)

CLI – Command Line Interface

- Human-computer interface using solely text input and output
- Pure control (e.g., scripting)
- Cumbersome multitasking
- Speed: Hack away at keys
- Convenient remote access (e.g. ssh)
- Example CLI: bash (linux), xterm (Windows)

Unix File System Layout

- A file system used by many Unix and Unix-like operating systems, including Linux
- Everything is a file (including devices)
- Tree structured hierarchy (with some exceptions)

Demo

- Log in with your Seasnet account
- Use Putty or ugrad or lnxsrv server
- Lost?
  - man <command>
  - Look up command usage in manual pages!
  - https://www.tutorialspoint.com/unix/unix-file-system.htm

The Basics: Moving Around

- pwd: print working directory
- cd: change working directory
- ~: home directory
- .: current directory
- /: root directory, or directory separator
- ..: parent directory

The Basics: Dealing with Files

- The basics continued…
  - mv: move a file (no undos!)
  - cp: copy a file
  - rm: remove a file
  - mkdir: make a directory
  - rmdir: remove a directory
  - ls: list contents of a directory
    - -a: list all files including hidden ones
    - -l: show long listing including permission info
    - -s: show size of each file, in blocks

The Basics: Look These Up

- cat
- head
- tail
- du
- ps
- kill
- diff
- cmp
- wc
- sort

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: wh.. command

- whatis <command>
  - return name section of man page
- whereis <command>
  - locates the binary, source and man page files for a command
- which <command>
  - locate a program file in the user's path

Q: difference between whereis and which?

Format of submission

- For lab questions(ans1.txt)
  - Answer 15 questions using natural language
  - List all the commands used to solve the problem
  - Give some explanations about your choice of commands
  - No need for keystrokes in this file
  - Will be graded manually