Week 01M: Getting to Know Your System

Thuy Vu

- General
  - `web.cs.ucla.edu/classes/winter17/cs35L/`
  - `piazza.com/ucla/winter2017/cs35l/home`
  - 100% = 50% Assignments + 50% Final
    - Each assignment counts 5%
    - Late penalty is $5\% \times 2^{\#\text{daylate}-1}$ of the 100%
  - DO NOT violate the UCLA Student Conduct Code
  - Thuy Vu — thuyvu@cs.ucla.edu
  - Final Exam: Tuesday, March 21, 2017, 11:30am-2:30pm (Location: TBD)
- Assignment 1
  - `web.cs.ucla.edu/classes/winter17/cs35L/assign/assign1.html`
  - Time due 23:55 this Saturday, January 14 (not Friday)

Why do we need to take this course?

to work on

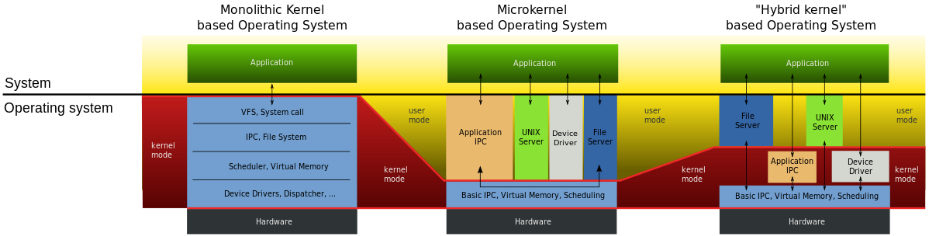
- open-source tools and environments
- quick overviews on: Linux, shell scripting, `git`, `gdb`, `ssh`, system call, multithread/parallel, dynamic library, and presentation
- "fundamentals of commonly-used software tools and environments, particularly open-source tools likely to be used in upper-division computer science courses."

upper-division computer science courses, what are they?

"but my major is not computer science...?"

...

Linux



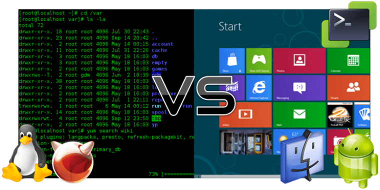
- Kernel – manages system resources and communications
- Shells – user interface of the kernel
- Applications

- Everything is either a file or a process

- Ubuntu 16.04.1 x86 Desktop (LTS)
  - `www.ubuntu.com/download/desktop/`
  - simple installation
    - bootable CD, USB stick from Windows and OS X
    - VMWare, VirtualBox, ...
    - `ssh your_id@lnxsrv.seas.ucla.edu`;
      hint: good up-to-date servers: lnxsrv06, lnxsrv07, and lnxsrv09

CLI vs GUI

Command Line Interface



```
[root@localhost ~]# cd /var
[root@localhost var]# ls -la
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x.  2 root root 4096 May 14 00:15 account
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache
drwxr-xr-x.  3 root root 4096 May 18 16:03 db
drwxr-xr-x.  3 root root 4096 May 18 16:03 empty
drwxr-xr-x.  2 root root 4096 May 18 16:03 games
drwxrwx--T.  2 root gdm  4096 Jun  2 18:39 gdm
drwxr-xr-x. 38 root root 4096 May 18 16:03
drwxr-xr-x.  2 root root 4096 May 18 16:03 lo
drwxr-xr-x.  2 root root 4096 May 18 16:03 pr    ve
drwxr-xr-x.  2 root root 4096 Jul  1 22:11 repo
lrwxrwxrwx.  1 root root    6 May 14 00:12 run      run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
drwxrwxrwt.  4 root root 4096 Sep 12 23:50 tmp
d    r-xr-x.  2 root root 4096 May 18 16:03 yp
    @localhost var]# yum search wiki
    d plugins: langpacks, presto, refresh-packagekit, re
    ion-f
    en-f           rimary_db
    -n             s
    et
                                   73% [==========
```

Interface

- have to write script (code)

  `me$gzip dump.txt`

- can click

  right-click on file, click “compress”

let’s compress 1,000 files separately :-)

- can code

  ```
  me$for i in {1..1000};
  me$do gzip dump_${i}.txt;
  me$done
  ```

- cannot code (?)

  ...

Unix Basics

- Path – absolute (/root/to/right_here/) and relative (right_here/)
- Wildcards – ?, *, and [] (- for range, ! to exclude)
- History – ↑, ↓, tab, !!, ...
- Redirection
  - > my_file (over)write stdout to my_file
  - >> my_file append stdout to my_file
  - < my_file content of my_file as stdin
  - 0> my_stdin, 1> my_stdout, and 2> my_stderr
    - 2>&1

Unix Commands

1. man
2. cd, pwd
3. $\boxed{\sim}$ home; $\boxed{.}$ current folder; $\boxed{..}$ parent folder; $\boxed{/}$ root or separator;
4. cat, head, tail, sort, ls
5. mv, cp, rm, mkdir, rmdir
6. ln, touch, chmod (r, w, x)
7. find $\boxed{\text{-perm ...}}$ $\boxed{\text{-name ...}}$ $\boxed{\text{-type ...}}$ $\boxed{\text{-user ...}}$ $\boxed{\text{-maxdepth ...}}$ $\boxed{\text{-ls}}$
8. du, diff, cmp, wc
9. ps, kill
10. whereis, whatis, which, who, whoami, ...

Emacs

- Install – `sudo apt-get install emacs`
- Have both GUI and CLI
- All commands start with Ctrl or Meta (Alt, ESC)
- C-x C-f open/create file, C-x C-s save, C-x C-w save as
- C-x C-b show buffer list, C-x o other window
- C-space/@ set *transient* mark, M-w copy, C-w cut, C-y yank
- C-x u / C-_ for undo/redo, C-x C-s for save change

Elisp

- C-x b enter `*scratch*`
- type `(random)` then C-j
- M-: and an expression to evaluate, e.g. `( * 1 2 3)`
- syntax (`func` a_1 a_2 `...`)
  - `( * 1 2 3)` → 6
  - `( setq x (random))` → execute function `random` which returns a random number, and use that return value to set to `x`