ICCS207: Term I/2018-19

Lecture 2: Linux Essentials

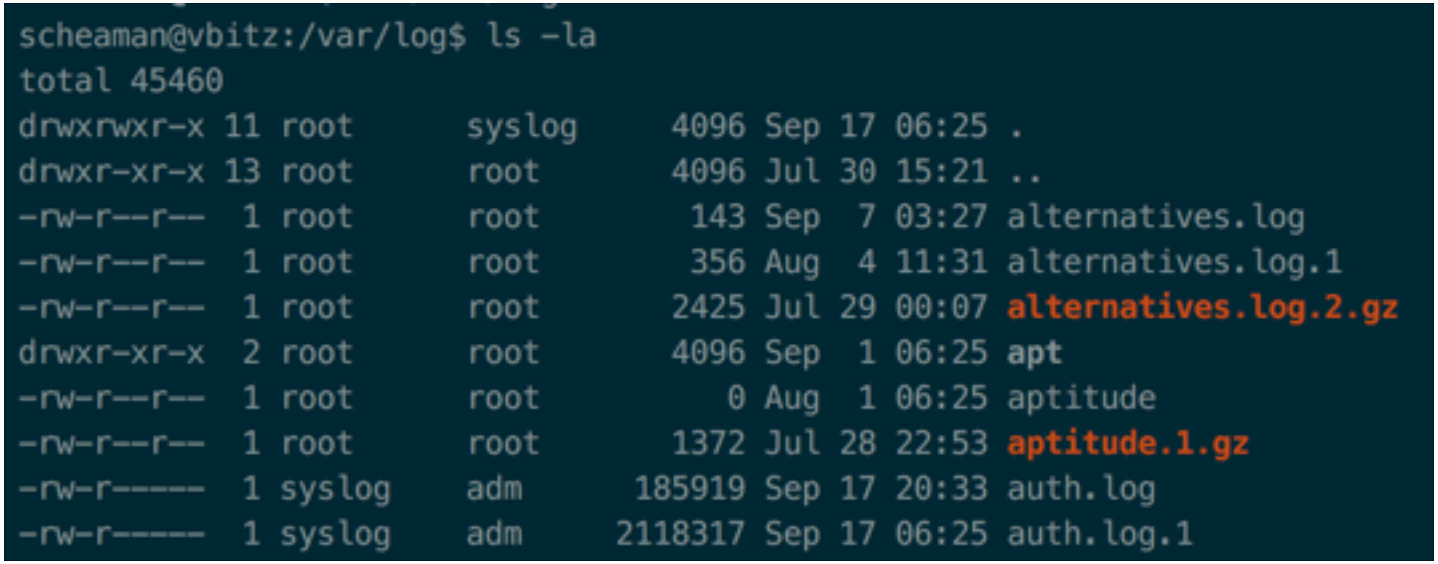
Sunsern Cheamanunkul (sunsern.che@mahidol.edu)

Shell

- A program that interprets commands.
- Allows a user to execute commands by typing them at a terminal, or automatically in programs called shell scripts.
- Remember a shell is NOT an operating system. It is just an interface to the OS.

Files and Directories

- Data and programs are stores in **files**
- **Files** are organized in directories



```
scheaman@vbitz:/var/log$ ls -la
total 45460
drwxrwxr-x 11 root      syslog     4096 Sep 17 06:25 .
drwxr-xr-x 13 root      root       4096 Jul 30 15:21 ..
-rw-r--r--  1 root      root        143 Sep  7 03:27 alternatives.log
-rw-r--r--  1 root      root        356 Aug  4 11:31 alternatives.log.1
-rw-r--r--  1 root      root       2425 Jul 29 00:07 alternatives.log.2.gz
drwxr-xr-x  2 root      root       4096 Sep  1 06:25 apt
-rw-r--r--  1 root      root          0 Aug  1 06:25 aptitude
-rw-r--r--  1 root      root       1372 Jul 28 22:53 aptitude.1.gz
-rw-r-----  1 syslog    adm      185919 Sep 17 20:33 auth.log
-rw-r-----  1 syslog    adm     2118317 Sep 17 06:25 auth.log.1
```

Paths

- Absolute path — specifies a location (file or directory) in relation to the root directory (/)
    - /var/log
    - /home/scheaman
- Relative path — specifies a location (file or directory) in relation to the current directory
    - ./assn/01
    - ../../lect/01

Paths

- ~ is your **HOME DIRECTORY**
  - This is where you start after you log in
  - On bash, you can also use $HOME
- . is an alias for your **PRESENT WORKING DIRECTORY**!
- .. is the file path for the **PARENT DIRECTORY** of your present working directory.
- / is the file path for the **TOP-LEVEL DIRECTORY** or **ROOT** directory.

Basic commands

- `pwd` – print working directory
- `ls` – list the current directory
  - `ls -l` – list the current directory with details
  - `ls -l /etc` – list /etc with details
  - `ls -la /var/log` – list all files in /var/log with details

cd <dir>

- Change the current working directory to <dir>
- Main tool for navigating a unix file system
- Some examples:
  - `cd ..` – Go up 1 level
  - `cd /` – Go to root level
  - `cd hello` – Go into directory hello
  - `cd ~` – Go back home

mkdir <dirname>

- Make a directory <dirname>
- Directories are like folders in Windows.
- Try it!
  - `mkdir test` — Create a directory called test in the current working directory
  - `mkdir /boo` — Create a directory called boo in the <u>root</u> directory
  - `mkdir ~/foo` — Create a directory called foo in your home directory

rmdir <dir>

- Remove directory <dir>
- The directory must be empty.
- Try it!
  - `rmdir test` – Create a directory called test in the current working directory
  - `rmdir /boo` – Create a directory called boo in the root directory
  - `rmdir ~/foo` – Create a directory called foo in your home directory

More about files in Linux

- Everything is a file
- File extensions are meaningless
- Filenames are case-sensitive
- You can have a space in file names but
  - you have to use quotes correctly:
    - `cd "test space"`
  - Or, use escape character (\)
    - `cd test\ space`

More about files in Linux

- Filenames that begin with . is considered hidden
  - `mkdir .this.is.hidden`
- To list the hidden files, use -a option:
  - `ls -a .`
  - `ls -la`

mv <src> <dst>

- Move a file from <src> to <dst>
- You can move both files and directories
  - Directories in Linux are just special type of files!
- If <dest> is an existing directory, the <src> will be put inside of <dest>.
- Also used for renaming a file/dir.
- Example:
  - `mv ~/.ssh/config ~/.ssh/config.bak`

cp <src> <dest>

- Copy from <src> to <dest>
- Similar to `mv`
- To copy the entire directory, use `cp -r`
- Example:
  - `cp foo bar`
  - `cp -r ~/hw1 ~/hw1-backup`

rm <file1> <file2> … <fileN>

- Remove the files
- To remove a non-empty directory
  - Use `rm -rf <dir>`
  - Please be very careful especially when combining with a wildcard character *
    - `rm -rf *`

touch <file>

- create a blank file
- `touch afile` – this will create a blank file called afile

cat <file1> <file2> … <fileN>

- Display the contents of the files in the terminal window.
- `cat hello.txt world.txt`

man <command>

- Open a manual for the <command>
- Try it!
  - `man grep`
  - `man printf`
  - `man strlen`

Wildcards

- * - represents zero or more characters
  - `ls u5*`
- ? - represents a single character
  - `ls ?m*`
- **[]** - represents a range of characters
  - `ls [0-9]*`

More examples

- Find the file type of every file in a directory:
  - `file /home/scheaman/*`
- Move all files of type either jpg or png (image files) into another directory.
  - `mv public_html/*.??g public_html/images/`

Exercise I

Redirection

- Use > to redirect output of a command to a new file.
  - `ls > myfiles.txt`
- Use >> to <u>append</u> to an existing file
  - `ls >> myfiles.txt`
  - `ls -l >> myfiles.txt`
- Use < to redirect a file as input to a program

cat <file1> <file2> … <fileN>

- You can combine multiple files into one.
  - `cat hello.txt world.txt > file.txt`

Pipe

- Use | to redirect the output of one command into another command
  - `ls | more`
  - `ls -la | grep config`
  - `cat somefile | wc`

User and group

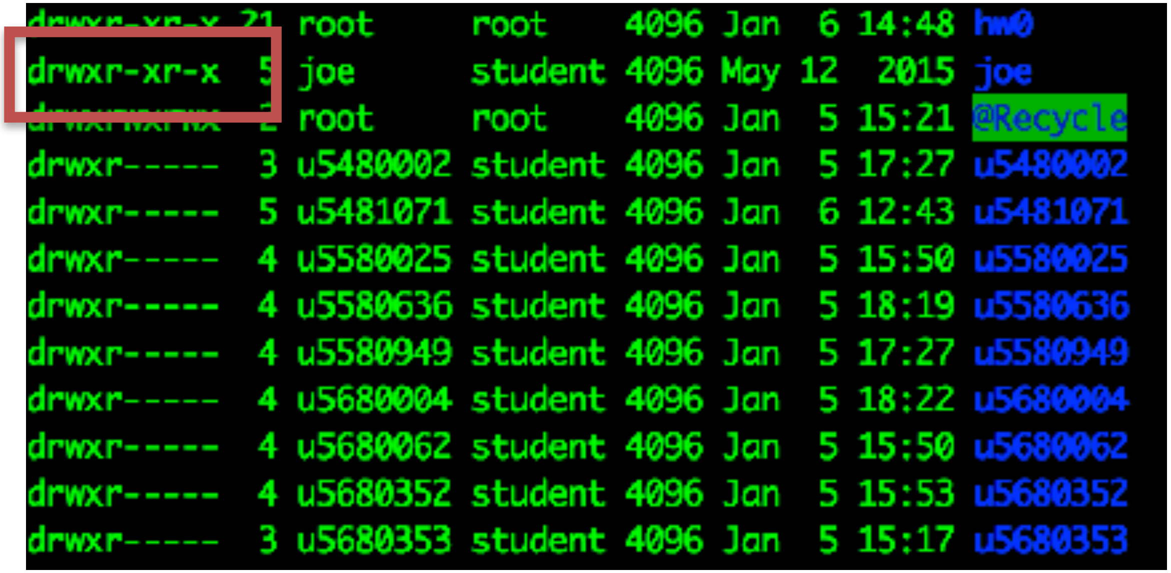
- Regular users vs super users
- Each user are belonged in some groups
- A user can be in multiple groups.
- Use `groups` to check which groups you are in.
- Some users can temporarily become super-users if they have “sudo” permission.

User and group

- Each file has “owner” — the creator of the file
- Each file has “group”
- Each file has 3 sets of permissions:
  - Read
  - Write
  - Execute
- Each set of permissions apply to:
  - Owner
  - Group
  - Other

Permissions

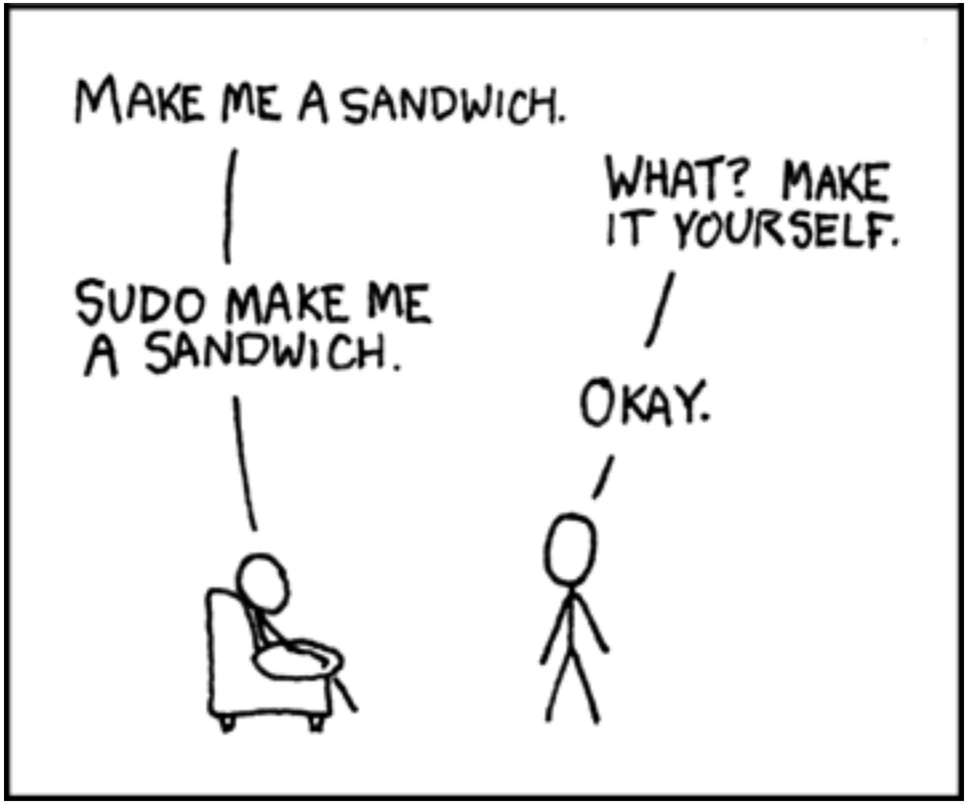
755



```
drwxr-xr-x 21 root      root    4096 Jan  6 14:48 hw0
drwxr-xr-x  5 joe       student 4096 May 12  2015 joe
drwxrwxrwx  2 root      root    4096 Jan  5 15:21 @Recycle
drwxr-----  3 u5480002 student 4096 Jan  5 17:27 u5480002
drwxr-----  5 u5481071 student 4096 Jan  6 12:43 u5481071
drwxr-----  4 u5580025 student 4096 Jan  5 15:50 u5580025
drwxr-----  4 u5580636 student 4096 Jan  5 18:19 u5580636
drwxr-----  4 u5580949 student 4096 Jan  5 17:27 u5580949
drwxr-----  4 u5680004 student 4096 Jan  5 18:22 u5680004
drwxr-----  4 u5680062 student 4096 Jan  5 15:50 u5680062
drwxr-----  4 u5680352 student 4096 Jan  5 15:53 u5680352
drwxr-----  3 u5680353 student 4096 Jan  5 15:17 u5680353
```

Superuser and root

- **superuser** is the most privileged user on a computer system.
- In Linux, the superuser is called **root**
- **root** can override all permission settings
- **su** — a command switch to a superuser (root password is required)
- **sudo** — a command to run something as superuser


MAKE ME A SANDWICH.
WHAT? MAKE IT YOURSELF.
SUDO MAKE ME A SANDWICH.
OKAY.

chmod <bitmask> <files>

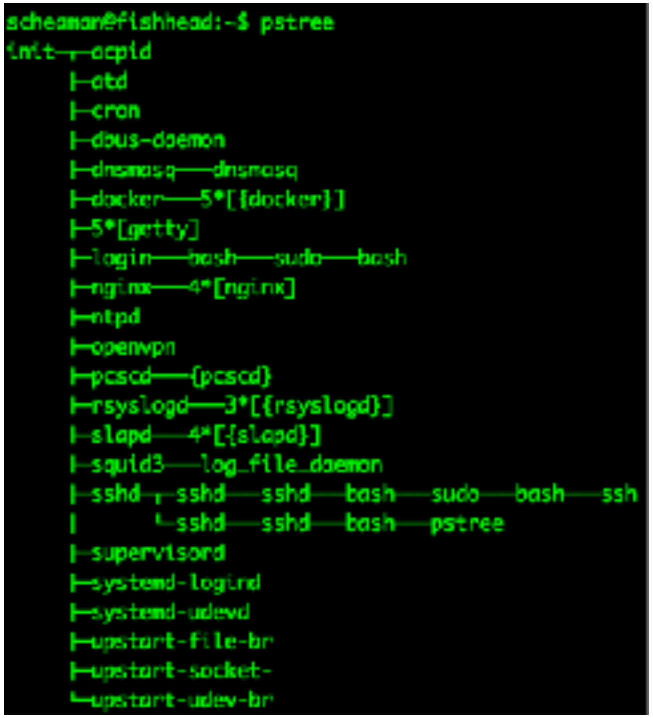
- Change the permission of <files>
- Must be the owner or a super user.
- Try it!
  - chmod 777 test
  - chmod 755 blah

The PATH Variable

- Most commands are located in your shell's PATH variable.
- `ls` is actually located in `/bin/ls`
- PATH must have '/bin' in it!
- Use `which` to check the binary location of a commend:
  - which ls

Program vs Process

- A <u>program</u> is a sequence of binary data that encodes machine instructions.
- A <u>process</u> is an running instance of a program.
- You can open up multiple terminals. Each terminal runs a shell in a separate process.
- Processes on a machine have a tree structure.



```
scheaman@fishhead:~$ pstree
init─┬─acpid
     ├─atd
     ├─cron
     ├─dbus-daemon
     ├─dnsmasq───dnsmasq
     ├─docker───5*[{docker}]
     ├─5*[getty]
     ├─login───bash───sudo───bash
     ├─nginx───4*[nginx]
     ├─ntpd
     ├─openvpn
     ├─pcscd───{pcscd}
     ├─rsyslogd───3*[{rsyslogd}]
     ├─slapd───4*[{slapd}]
     ├─squid3───log_file_daemon
     ├─sshd─┬─sshd───sshd───bash───sudo───bash───ssh
     │      └─sshd───sshd───bash───pstree
     ├─supervisord
     ├─systemd-logind
     ├─systemd-udevd
     ├─upstart-file-br
     ├─upstart-socket-
     └─upstart-udev-br
```

Process Management

- top — View real-time data about processes running on the system.
- ps — Get a listing of processes running on the system.
- kill — End the running of a process.
- jobs — Display a list of current jobs running in the background.
- fg — Move a background process into the foreground.
- Ctrl + z — Pause the current foreground process and move it into the background.

A few more useful tools

- wget — download a file from the internet
- zip/unzip — compress and decompress
- tar — packing multiple files into one
- curl — HTTP client
- head — display first N lines
- tail — display last N lines

Some tips when using terminal

- Pressing `tab` will **autocomplete** file and folder names!
- `Control+C` will **stop** execution of your current program!
- `Control+R` will let you **search** your command history!
- `Control+L` will **clear** your screen!
- Use the **up** and **down** arrow keys to **scroll through your command history**!

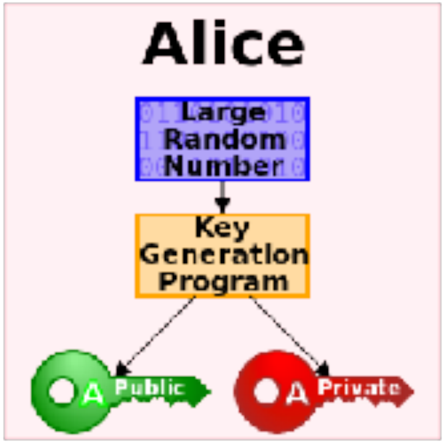
Activity Time!

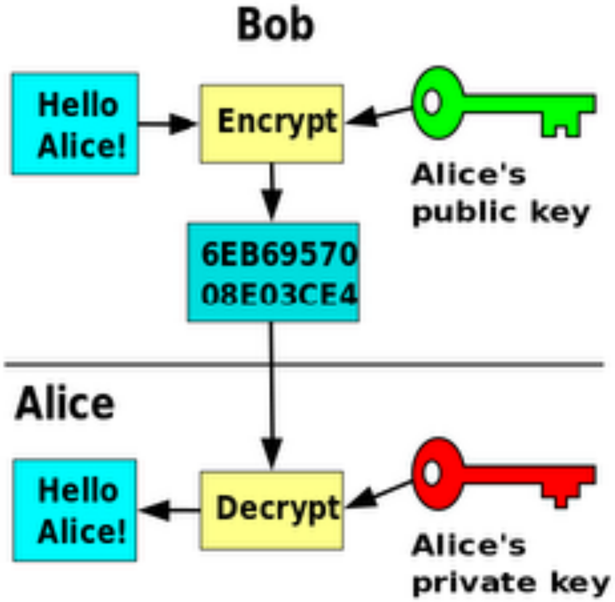
- Copy between machines
- Password-less login

scp <src> <dest>

- Copy files between hosts
- <src> and <dest> should be written:
  - <user>@<hostname>:<file>
  - Examples:
    - `scp ~/.vimrc u12345@hamachi:~/`
    - `scp u12345@hamachi:~/.vimrc .`
- Use the `-r` option to copy the entire directory

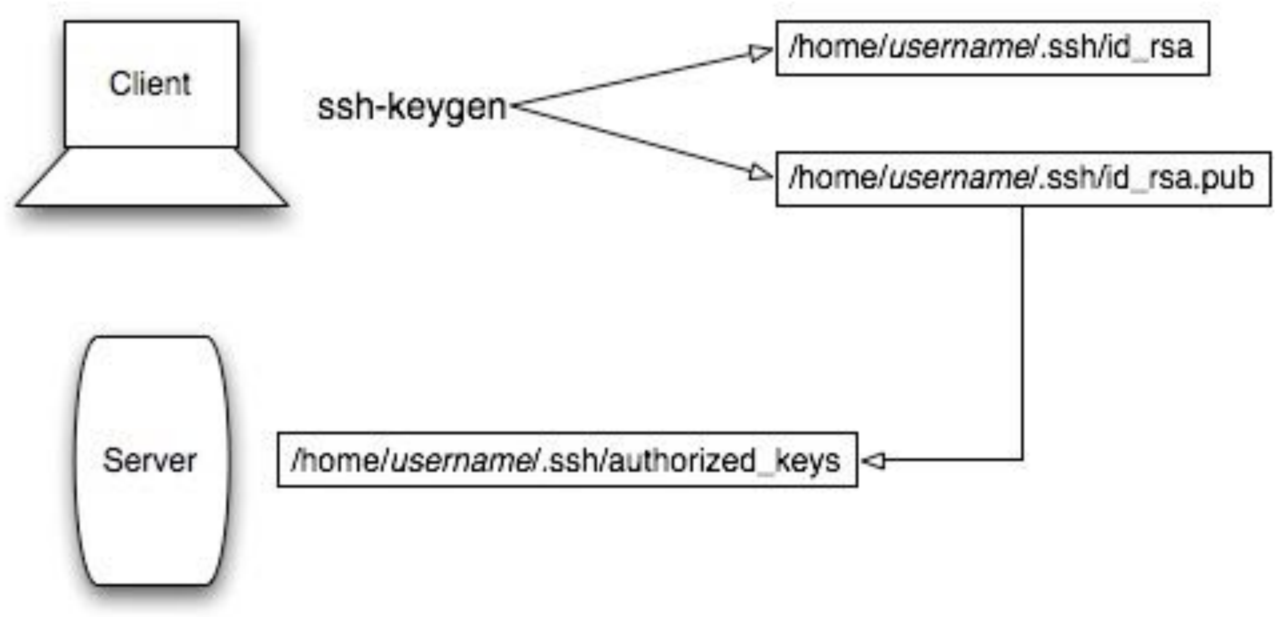
Public-key Cryptography





Password-less Login

- Use SSH key authentication instead of typing password in every time.


Client
ssh-keygen
/home/*username*/.ssh/id_rsa
/home/*username*/.ssh/id_rsa.pub
Server
/home/*username*/.ssh/authorized_keys

Password-less Login

- 2 simple steps:

1. Generate a SSH key pair

```
ssh-keygen -t rsa -b 4096
```

2. Append id_rsa.pub to the server's authorized_keys located at:

```
~/.ssh/authorized_keys
```