System Administration

Many regular users and only ONE super user (root)

- Regular users have access only to the files and processes they own or are given specific permissions
- Super user can have access to all the files and processes on the system, and has certain system-wide privilege that are beyond those of ordinary users, such as
 - Halt, and/or reboot the system
 - Install or remove software /packages (rpms or deb, etc.) system wide
 - Read, write and execute any file and can change permissions of any files, create or delete any files and directories
 - Delete any processes owned by any users
 - Can reset any user's password w/o knowing the current one
 - Can add/delete any user to/from the system
 - System-wide backups

The separation of normal user and root makes Linux system more secure than other

Super User (root)

Switch to a Superuser with su (Switch to a different User) command

- su <root> (root is optional), you will be asked to provide password of the "root".
- su username (to log on as another regular user), you will be asked for password of that user
- A special prompt for super user # instead of others (\$ or %)
- Because of the enormous powers of "destruction" of a Super user, it is suggested to switch the Super user only when it's necessary
 - All system administrators have their regular accounts
 - Normally, the administrator first logon the system as regular user, and switch to super user root only when the task requires super user privilege.
 - Return (exit) to regular user afterwards

sudo and sudoers

- In Ubuntu, the root user account is disabled by default for security reason
- * "sudo" allows an authorized user to execute a command as the super user (root) or another user.
 - User's "privilege" is defined by "security" policy stated in file such as /etc/sudoers
- sudoers file contains the information regarding privileges for different users and groups. It's one of the most im[portant files in the system.
 - file looks like the following"
 - #user privilege specification

user (host)=(user:group) commands

A given user can run the specified commands as the specified user or group from the specified hosts

- root ALL=(ALL:ALL) ALL

A root user can execute any command as any user or any group from any host

Group Privileges

Any user having a group sudo can execute any command as any user or any group from any host

You can give a sudo privilege to a regular user by adding the user to the admin or sudo group, whichever exists

sudo adduser jsmith admin

Booting up the System

- Booting is the process of reading the Linux system kernel into the system memory and start to run it
- * "init" is the first process with PID=1, the ancestor of all the processes
 - "init" has several 'run levels', normally the system would be in any one of them
 - Level 0: halt
 - 1: single user mode
 - 2: multiuser mode (without NFS)
 - 3: full multiple user mode
 - 4: unused
 - 5: full multiple user with X11 (the default run level)
 - 6: shutdown and reboot: (# init 6 will reboot the system)
 - S or s: single mode
 - #/sbin/init 0 will halt the system
 - #/sbin/init s or S will switch to single user mode
 - #/sbin/init 5 will boot the system in the regular multiuser mode
 - #/sbin/init 1 will boot the system in the single user mode (root)

System Shutdown and Restart

- Shutdown from a terminal session with command /sbin/shutdown
- Use "wall" to write shutdown messages to all current logon users, normally give a short time such as one minute before actions:
 - Sends signals to all running processes so they can terminate normally
 - Logs users off and kills remaining processes
 - Unmounts all secondary file system using the "umount" command
- Change to root with su or as a sudoer with admin privilege
 - ♦ # /sbin/shutdown -r now
 - -r: reboot after shutdown
 - now: start right now
 - ◆ Halt the system: /sbin/shutdown -h now
 - Shutdown in some time later

/sbin/shutdown -r +30

- system will be shutdown in 30 seconds"
- Learn more about shutdown online: man shutdown

Single User Mode

- System console is enabled, only "root" can access the system
- Mainly for system maintenance, not all system processes are running, and many services are not available in this state
 - No network configuration, no daemon running /started, no NFS

User Administration

Create new accounts:

- /usr/sbin/useradd newuser
 - It creates a directory under such as /home/users with the default shell
 - Get the next available user ID number
 - The user can only create/modify/delete files under the home directory
 - The user might be able to read and copy files of other users
- ◆ passwd newuser
- More options in creating a user account
 - Assign a different shell for the to be added new user
 useradd -s /bin/tcsh jsmith
 - Make it under a different directory other than \$HOME with option

useradd -d dir_name jsmith

• The LOGIN (name) is always put at the last, ex:

/usr/sbin/useradd -d /home/tempusers -s /bin/tcsh LOGIN

/etc/passwd file

/etc/passwd file has the following format

chwalker:x:2061:110:Christopher Walker:/home/csuser/chwalker:/bin/bash hlin:x:1688:110:Hong Lin:/home/csuser/hlin:/bin/bash aschillo:x:2064:110:Alexander Schilloff:/home/csuser/aschillo:/bin/bash jcampbel:x:2065:110:Joseph S Campbell:/home/csuser/jcampbel:/bin/bash

- 1. Username: up to 8 characters, case sensitive, normally lower case
- 2. "x" means the passwords are saved in /etc/shadow
- 3. uid- user id
- 4. group id
- 5. User's full name
- 6. User's home directory
- 7. User's default start up shell

/etc/shadow file

Normally only the root can read/write this file

/ etc/shadow

hlin:\$1\$TZUQqyzz\$d5CU0mCyYtq.gtTPZwhzJ.:13346::::: aschillo:YAiZcb1ocEaGQ:13388:::::13531: jcampbel:gf8eGjphMmkUY:13388:::::13531:

- 1. Username
- 2. Encrypted password
- 3. The number of days (since January 1, 1970) since the password was last changed
- 4. the number of days before password may be changed (0 indicates it may be changed at any time)
- 5. The number of days after which password *must* be changed (99999 indicates user can keep his or her password unchanged for many, many years)
- 6. The number of days to warn user of an expiring password (7 for a full week)
- 7. The number of days after password expires that account is disabled
- 8. The number of days since January 1, 1970 that an account has been disabled
- 9. A reserved field for possible future use

Disable / Remove User Accounts

- To disable a user account
 - Open /etc/passwd or /etc/shadow file
 - Replace the encrypted password field with "*" for that user
 - prefix "*" of the encrypted password field
- To re-enable the user
 - Re-assign the user a new password
 - The "*" field will be refreshed with the new password (encrypted)
 - re-enable the user by simply removing the prefix * if it is disabled that way
- Remove a user account

/usr/sbin/userdel -r user_name

- You can choose to save the user's account by not using the "-r" option
- The record in file /etc/passwd about this user will be removed.

System Backup and Restore

- System backup is one of the daily routines a system administrator should do
 - In case of file loss due to system malfunctions
 - In case a user delete a file or files accidentally
- Backup files on a different media
 - Floppy, Tape, removable drive, writable CD or DVD, backup server, cloud, etc.
- Full backups and partial backups
 - A full backup makes copies of all files, regardless of when they were created or accessed
 - A partial backup makes copies of the files that have been created or modified since the last backup

Debian Package (.dpkg)

- Ubuntu System uses Debian package management
- Originally derived from the system used by the Debian GNUL/Linux distribution
- To list all packages installed on the system from the shell dpkg -1
- Use grep for a specific package of interest

dpkg -1 | grep python

- To list the files installed by a packgage
 - ♦ dpkg -L perl
- To find out which package contains a file of interest dpkg -S
- To install a local package, such as zip_4.0_i386.deb
 sudo dpkg -i zip_4.0_i386.deb
- To UN-install a package, such as zip_4.0_i386.deb sudo dpkg -r zip

apt-get

Advanced Packaging Tool (APT), a powerful command line tool

- Installation / Removal of software packages
- Upgrade existing software packages
- Updating of the package list index
- Upgrading the entire Ubuntu system
- Install a package

sudo apt-get install packagename

Remove a package

sudo apt-get remove packagename

- Update package index
 - The apt package index is essentially a database of available packages from the repositories defined in the /etc/apt/sources.list

sudo apt-get update

Update packages (update source list first)

sudo apt-get update

sudo apt-get upgrade

RPM: Redhat Package Manager

Package

- A collection of one or more programs that performs a single welldefined task
- A specific part of an operating system, such as initialization files, etc.
- One easily manageable chunk
- The package contains knowledge about what it takes to install itself to the system, as well as how to un-install from your system

Managing packages

- There are 1988+ packages on RHEL 7.6 !!
- RPM uses its database to track which packages and what version of them have been installed on the system, and the dependence among the packages
 - You can not remove a package if some other packages depends on it!
- Install, upgrade, and remove packages easily

Upgrade and Install Software

Using RPM on Redhat Linux system

rpm -Uvh package-name-version-release.xxx.rpm

Install software is in the form of tarball

- You need to decompress the tarball, then change to the top of the uncompressed directory, read the README or/and INSTALL file
- Normally need to do the following

./configure; make; make install

Kernel updates

- Stable version: 2nd number is EVEN, such as 2.2.8, 2.4.9, 2.6.3
- Unstable version: the 2nd number is ODD, such as 2.1.2, 2.3.5, 2.7.4
- Try to use stable version if possible which has been fully tested

**NOTE:

 try to use RPM (instead of from tarball) if possible, so the removing can be an easy task

RPM Types and Naming

Binary RPM: ending with "rpm"

- Application programs for installation
- Can be architecture dependent, i386, i586, i686:
 - findutils-4.2.27-4.1.i386.rpm
- Not architecture dependent: noarch, such as:
 - fedora-logos-6.0.6-1.fc6.noarch.rpm
- Check link: <u>http://mirror.centos.org/centos/8-stream/BaseOS/x86_64/os/Packages/</u>

Source RPM: srpm

Contains source code of the corresponding RPM

http://download.fedora.redhat.com/pub/fedora/linux/releases/9/F edora/source/SRPMS/

Query Pkg Information

- Knowing the installed package name, such as perl
 - rpm -q perl # get the package name, version and release information
 - rpm -ql perl | more # get the list of the files installed with this package
 - rpm -qi perl # get the package information only
 - rpm -qd perl # list all the documentation of this package

Check if a package (such as python) has been installed

- ♦ rpm -qa |grep python
- rpm -qa | wc # how many packages installed on system
- Partial match when not sure about the exact package name.
 - rpm -qa |grep string (knowing part of pkg name)
- Find the package a file belongs to
 - rpm -qf /path/of/the/file
- Find packages and files that perl depends on

Slide #18
 rpm -q -requires perl

New Packages (Not Installed Yet)

- Search and download the package to your system
 - Ex: download : aspell-0.60.3-7.1.i386.rpm
- Check information about this package
 - ◆ rpm -qpi aspell-0.60.3-7.1.i386.rpm
- Get the list of files contained inside this package before installing
 - ♦ rpm -qpl aspell-0.60.3-7.1.i386.rpm

The added 'p' option tells rpm to search the specified package, not from rpm database

New Package Installation

rpm -i (or -install) package_name-version-release.arch.rpm

- Performs dependency checks
- Performs any tasks required before the installation
- Unpacks files from the package and puts them in the proper places defined inside the package itself
- Performs any tasks required after the installation
- Keeps track of what it has done
- Two extra handy options:

rpm -ivh package_name-version-release.arch.rpm

- -v: display what is being installed when installing multiple packages
- -h: progress bar (moving hash symbols) in case you want to see it is working...
- --test option: perform installation test only

rpm -i -test perl-5.8.8-20.i386.rpm

- --nodeps: do not check for dependencies before install
- --force: install anyway no matter what
- --excludedocs: do not install documentation for this package to save disk space in some cases

Erase Packages

* rpm -e package_name

- Package Name ONLY, no Version, no Release and the rest...
- Checks the RPM database to make sure no other packages depend on it
- Checks and deletes the files belonging to this package from RPM database
- Removes all traces of the package from the database
- Erase will NOT be performed if other packages depending on this package.
- More options for erasing
 - -vv option to get more info about the erasing process
 - -test option for erasing test
 - -nodeps: do not check for dependency before erasing

Upgrade Packages

rpm -U [whole package name]

- Installs the desired package
- Erase all older version of the package if exits
- rpm -U can also be used for new installation

Used in most of package installation and updating:

rpm -Uvh package_names-version-release.i586.rpm

YUM- Yellowdog Updater Modified

- Command-line package management utility for Linux system using RPM package management such as Centos, Federa, etc
- **Package installation:** yum -y install python
- Package removal: yum -y remove python
- Package update: yum update mysql
- Package info: yum info hdf5
- Package search: yum search hdf5
- List all packages installed: yum list installed
- **List all available package:** yum list
- **Check for update:** yum check-update
- Database/list of repositories: /etc/yum.repos.d