[StuCo 98008]
GNU/Linux for Beginners

Session 5

System Administration

By the end of this lecture you will know

- How to administer user/group information
- How to interpret and use filesystem permissions
- How to add/remove applications
- How to handle processes
- Scheduling and automating tasks
- Managing hardware

Users and groups

- /etc/passwd stores information about user accounts
- **Except** their passwords, these are at /etc/shadow
- A typical /etc/passwd entry:

freenet:x:1004:1004:Freenet Test Account,,,:/home/freenet:/bin/bash

- /etc/group stores information about groups
- The group passwords are stored in /etc/gshadow
- A typical /etc/group entry:

cdrecording:x:1007:alex,ddrew

Administering User Information

- One way: Hand-editing /etc/passwd
  - Simple but tricky
- Other way: GUI utility
  - Simper but trickier
- Best way: CLI utilities
  - useradd/groupadd : Add new user/group
  - userdel/groupdel : Delete existing user/group
  - usermod/groupmod : Modify user/group information

Filesystem permissions

- Basic file attributes, like file type, permission to read/write/execute, which user/group owns the file:

-rwxr-x**r-x** **1** alex **staff** 385K **Aug 29 06:57** .bashrc

drwxr-x**r-x** **4** ddrew **staff** 4,0K **Sep 17 22:47** Library

- File type:
  - **-** Regular file
  - **d** Directory
  - **l** Link
- Access Permissions
  - User, Group, Others
  - **r**ead, **w**rite, e**x**ecute
- Reference count
- Owner
- Group
- Size on disk
- Modification time
- Name

Changing permissions

- chown/chmod : Change the owner or group of a file

Relative Permissions

- chmod [a,u,g,o][+,-][r,w,x] <filename>

[**a**ll, **u**ser, **g**roup, **o**thers] [add, remove] [read, write, execute]

Absolute Permissions

- 4 – read
- 2 – write
- 1 – execute

So, if we want to change a file's permissions to **rwx**r-x**r-x** , the command **chmod 7**5**5** does the trick

Managing Applications

- Applications are provided as **packages** of the following types:
  - .rpm (**R**edHat **P**ackage **M**anager)
  - .deb (Debian **A**dvanced **P**ackage **T**ool)
  - .tar [.gz - .bz2] (tarballs, often compressed with gzip or bzip2)

RPM

- RPM is the preferred package management tool for Red Hat, SuSE, Mandrake, Gentoo
- # rpm -U sendmail-8.3.0.rpm **: install/upgrade**
- # rpm -q sendmail **: query installed package**
- # rpm -e sendmail-8.3.0 **: erase installed package**
- # rpm -qa **: show all installed packages**
  Combine with grep to search for a pattern, e.g.
  # rpm -qa | grep sendmail
- Add the flags **v,h** for more verbose output

DEB

- DEB is the preferred package format for Debian.
- Debian uses multiple levels of tools to manage its packages:
  - dpkg, apt-get, aptitude, dselect...
- Recommended method:
  - # apt-get **install** sendmail
  - # apt-get **remove** sendmail
- # apt-get update *: updates your local package index*
- # apt-cache search icons *: look for a package that has something to do with icons*

TARballs

- Some distributions intentionally do this (e.g. Slackware)
- Sometimes, the application you want has not been pre-packaged by your distribution (no RPM, no DEB), but source tarballs are always available
- Problem/blessing: No centralized database (like rpm -qa or dpkg --list)
- Usual procedure for installation:
  - Untar/uncompress: $ tar -xvzf qmail-3.0.1.tar.gz
  - Tailor the parameters to your system: $ *./configure*
  - Compile the application: *$ make*
  - Install it (usually needs root privileges): *# make install*

Processes

- Any program that executes in userland spawns one or more processes, each of which has a process ID (PID).
- Listing:
  - *$ ps auxf* : List all **P**rocesse**S** in tree format
  - *$ top -c* : Show list of most resource-intensive processes
- Killing:
  - *$ kill <PID>* : Sends a **TERM**inate signal to the process. Works only if you own the process, or are root
  - *$ killall <process name>* : Kill processes by name (dangerous but convenient)
  - $ *kill -9 <PID>* : Send a **KILL** signal to the process (last resort for something that's not responding)

Scheduling

- Scheduling stuff to be executed just once:
  - $ **sleep** 12m; echo “Spaghetti on fire”
    - sleep simply counts time since invocation (relative scheduling)
  - $ **at** 15:00 echo “Wake up – time to go to class”
    - relative or absolute scheduling, e.g $ at +10m echo “Time to go”
    - $ atrm 1 : removes the first scheduled job
    - $ atq : prints the queue of scheduled jobs
- Persistent scheduling: **cron**
  - **$ crontab -e :** edit my crontab, with the following syntax:

    00 4 * * **wed,sat** alex **/home/alex/scripts/backup_home.sh**

    minute, hour, **day of month**, month, **day of week**, user, **command**
  - $ crontab -l : list my crontab

Hardware Information

- **D**isk **F**ree space: $ df (-h for **h**uman-readable format)
- **Free** RAM: $ free (-m for displaying RAM in **m**egabytes)
- CPU info: $ cat /proc/cpuinfo
- Boot-time device detection : $ dmesg (| less)
- PCI bus information: $ lspci (-v for more **v**erbose output)