Week 8

Knowing your hardware

Hardware items

- CPU
- Storage & Partitions
- Graphics Card
- Memory Modules
- Battery & status
- Network devices & configuration

Packages to install

- `clinfo`
- `coreutils`
- `dmidecode`
- `fdisk`
- `hardinfo`
- `hdparm`
- `hwinfo`
  - Will probe hardware and show output on the screen. Information about PCI, partitions, keyboard, tablet, earphones etc..
- `lshw`
  - List hardware. Output is in various sections. You can choose a particular section eg : `lshw -c display` or `lshw -c processor` or `lshw -c memory`
- `memtester`
- `net-tools`
- `pciutils`
- `procps`
- `sysstat`
- `upower`
- `util-linux`
- Demo

- `cat /proc/cpuinfo` gives information about the CPU.
- `cat /proc/partitions` gives partition information. The loop partitions are meant for snap packages
- `lsblk -o NAME,SIZE` gives information about the number of block devices that are available.
- `lspci` gives the list of PCI devices connected to the computer using the PCI bus.
- `free` gives details about the amount of memory used. It is a practice to have double the size of the memory as swap
- `sudo dmidecode --type memory` gives information about the memory and modules.
- `hardinfo` is a GUI utility
- `clinfo` gives information about the graphics card.
- `upower -e` to know about the battery status. This will give a list.Choose the one that says battery and execute `upower -i /org/freedesktop/UPower/devices/battery_BAT0` for example.
- `sudo hdparm -Tt /dev/sda` runs diagnostics on the ssd or hdd. Timing cached reads and buffered disk reads.
- `iostat -dx /dev/sdb` gives information about speeds of various disks.
- `ifconfig` is a network utility that gives information about ethernet / loopback / wifi adapter.

Prompt strings

- Context for prompt strings
  - `bash` , `dash` , `zsh` , `ksh` , `csh`
  - `python`
  - `octave` - Mathlab compatible numerical package
  - `gnuplot` - Plotting tool
  - `sage` - symbolic cumputing package. Perhaps better than Mathematica
- bash prompts
  - PS1 : primary prompt string : `$`
  - PS2 : secondary prompt for multi-line input : `>`
  - PS3 : prompt string in select loops : `#?`
  - PS4 : prompt srting for execution trace : `+` Explanation: There are 4 bash prompts that are configured. What we see is normally the primary prompt when we open the shell. PS2 is shown when a command is incomplete. PS3 is shown when we run a bash script in a select loop. PS4 is shown when every command that is executed is displayed on the screen - when we use the option `set -x`
- Escape sequences

`\u@\h:\w\$` - This is the default value of PS1. Username @ machine name : current dir $ if user is not superuser

To change what is displayed in the prompt string.

- Python command line
  - ps1 and ps2 are defined in the module sys
  - Change `sys.ps1` and `sys.ps2` if needed
  - Override `__str__` method to have dynamic prompt `>>>` Default python command prompt

Demo

- `echo $PS1` - gives `\[\e]0;\u@\h: \w\a\]${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$`
- It can be changed `PS1="\u@\h:\w\$ "` . The color will be lost from above prompt string.
- If you do `source .bashrc` you get back the prompt after messing up the prompt string.
- `less .bashrc` to see where it is defined.
- `echo $PS3` doesn't display anything.
- `select x in alpha beta gamma; do echo $x; done` displays PS3
- In octave `x=[1:1:100]` creates array
- In SageMath `plot(sin(x),x,0,2*pi)`

Important Utilities

- `find` – locating files and processing them
- `tar , gzip` etc – packaging collections of files
- `make` – conditional actions

find

- `find [pathnames] [conditions]`

file packaging

- Deep file hierarchies
- Large number of tiny files
- `tar` : collect a file hierarchy into a single file
- `gzip` : compress a file
- Applications: backup, file sharing, reduce disc utilization Explanations:
- Sometimes when there are several small files in a hierarchy structure, the files may occupy the minimum block size so there is a wastage of space. In such situations doing a `tar` will save space.

Possibilities

- `tar , zip`
- compress (ncompress), gzip (ncompress), bzip2 (bzip2), xz (xz-utils), 7z (p7zip-full)
- Tarballs like bundle.tgz for package + compress
- Time & memory required to shrink / expand versus size ratio
- Portability
- Unique names using timestamp, process ID etc., for backup tarballs Explanation :
- Plain text or ASCII files can be compressed to a very good ratio (almost 1:10) if the file contains repeting patterns.
- For more efficiency, first zip and then make a tar. Zipping the files while adding it to tar - .tgz file format combines tar and gzip together.
- The deciscion on which method to use is taken based on time required, space occupied etc..

make

- `make -f make.file`

Network & ssh

Accessing remote machines on command line

- IPv4 address range
  - Localhost
    - `127.0.0.0/8`
  - Private network
    - `Class A : 10.0.0.0/8` - 16,777,216
    - `Class B : 172.16.0.0/12` - 1,048,,576
    - `Class C : 192.168.0.0/16` - 65,536
  - Public network
- Ways to gain remote access
  - VPN access
  - ssh tunneling
  - Remote desktop : x2go, rdp, pcoip,
  - Desktop over browser: Apache Guacomole
  - Commercial, over internet : Teamviewer, AnyDesk,Zoho assist, ...
- Some important ports | Port | Service | Description | |---|---|---| | `21` | `ftp` | File transfer | | `22` | `ssh` | Secure Shell | | `25` | `smtp` | Simple Mail Transfer Protocol | | `80` | `http` | Hypertext Tranfer Protocol | | `443` | `https` | Secure Hypertext Tranfer Protocol | | `631` | `cups` | Common Unix Printing System | | `3306` | `mysql` | MySQL database |
- Firewall
  - Ports open on my machine
  - Ports needed to be accessed on remote machine
  - Network routing over the port
  - Firewall controls at each hop
- Protecting a server
  - Server with a public service > Web Application Filter > Network Firewall > Anonymous users
- SELinux
  - Security Enhanced Linux mode available on Ubuntu too, apart from server grade flavors like CentOS, Fedora, RHEL, SuSE Linux etc.,
  - Additional layer of access control on files to services
  - Role Based Access Control
  - Process sandboxing, least privilege access for subjects
  - Check using `ls -lZ` and `ps -eZ`
  - RBAC items: user `(unconfined_u)` , role `(object_r)` , type `(user_home_t)` , level `(s0)`
  - Modes: `disabled` , `enforcing` , `permissive`

    - Tools: `semanage`, `restorecon`
    - SELinux is recommended for all publicly visible servers
- Network tools

- High Performance Computing
    - Look at www.top500.org for statistics
    - Accessing a remote HPC machine is usually over SSH
    - Long duration jobs are submitted to a job scheduler for execution
    - Raw data if large needs to be processed remotely before being transferred to your machine (network charges? bandwidth?)
    - Comfort with command line is a must

Automating scripts

Scheduled, recurring, automatic execution of scripts

- `cron`
    - Service to run scripts automatically at scheduled times
    - Tools: `at`, `crontab`, `anacron`, `logrotate`
    - Script locations:
        - `/etc/crontab`
        - `/etc/cron.d`
        - `/etc/cron.hourly`
        - `/etc/cron.daily`
        - `/etc/cron.weekly`
        - `/etc/cron.monthly`

Example of job definition: .---------------- minute (0 - 59) | .------------- hour (0 - 23) | | .---------- day of month (1 - 31) | | | .------- month (1 - 12) OR jan,feb,mar,apr ... | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat | | | | |

-
    -

        -
            -
                - user-name command to be executed

- Job definition
    - `5 2 * * 1-5 root cd /home/scripts/backup && ./mkbackup.sh`
        - `5` minute (0-59)
        - `2` hour (0-23)
        - `*` day of the month (1-31)
        - `*` month (1-12) or jan, feb, ...
        - `1-5` day of week (0-6) or sun,mon, ...
        - `root` user-name
        - `cd /home/scripts/backup` command
    - The above command runs mkbackup.sh as root every working day at 02:05 AM
- Demonstration
    - The first time crontab is used you have to select the default editor
    - crontab is in `etc` directory
    - `anacron` is run by the System administrator
    - `cron.daily` is a folder in `etc` that displays daily tasks.
      Similarly `cron.hourly` , `cron.monthly` , `cron.weekly`
    - By placing a script in any of these directories you can make it run at the specified schedule
    - By running `crontab -e` you can execute a specific script at a time. Customize timely running of scripts.
- Startup scripts
    - Startup scripts: `/etc/init/` , `/etc/init.d/`
    - Runlevel scripts:

Managing Storage

LVM & RAID

- LVM
  - Logical Volume Management
  - Pooling multiple storage devices as a single logical volume
  - `lvm2 tools` : create and manage virtual block devices from physical devices
  - Suppose you need a very large partition but there is no HDD available of that size, you can define a logical volume that spans over multiple HDDs.
  - Logical Volumes are mounted by the GNU Linux OS, which are mapped over multiple physical disks.
- RAID
  - Redundant Arrays of Independent Disks
  - Distributing data over multiple discs for redundancy / speed / increased capacity
  - Raid Controller : software or hardware
- RAID modes
  - usable capacity < actual capacity

- Explanation
  - RAID 0 - You are using 2 disks as 1. Half of one file is stored on 2 disks. Doubles speed of access of a file. Write Speed is 2x and Read Speed is 2x for 2 disks. If there are n disks in RAID 0 equivalent storage is size of minimum disk * n.
  - RAID 1 - Any piece of the file is written to both the disks. Reading is 2x but writing is n-1. People tend to use RAID 1 for OS alone.
  - RAID 5 - When you have more than 3 disks. Data is written to more than one disk. If one fails nothing is lost.
  - RAID 6 - Parity over 2 disks. If 2 disks fail you still have all your data.
  - Most of the hardware supprts hot-swap.
  - Useable capacity is less than the actual capacity
  - For storage people use RAID 5 or RAID 6.
- Demo
  - `df -h` to check system storage
  - WHich RAID configuration to use to improve read performance and sustain at least one disk failure without losing data ? RAID 4, RAID 6.