# Sys Admin commands:

Table of Contents

[Sys Admin commands: 1](#_Toc165225346)

[Common: 1](#_Toc165225347)

[Memory and Process 1](#_Toc165225348)

[Network: 2](#_Toc165225349)

[To configure DHCP 2](#_Toc165225350)

[DNS: 2](#_Toc165225351)

[File: 2](#_Toc165225352)

[Inode: 3](#_Toc165225353)

[Directory: 3](#_Toc165225354)

[Backup and Recover: 4](#_Toc165225355)

[Backup Tools: 4](#_Toc165225356)

[Automatic Backup: 5](#_Toc165225357)

[Backup Recovery: 6](#_Toc165225358)

[User: 7](#_Toc165225359)

## Common:

* connect to a database server from linux; isql –S serverName –U username –P password.
* Shebang line at top of each script determines the location of the engine which is to be used in order to execute the script.

## Memory and Process

* dmesg: command to view the contents of the kernel ring buffer.
* cat /var/log/messages command will show some informational linux message.
* Ipconfig: To get the network information
* Insmod /lib/modules/<version>/kernel/fs/fat/fat.ko – to load the kernel.
* Rmmod fat to remove the moule
* fdisk for mbr disk partition
* mkfs make file sys : mkfs -t ext4 /dev/sda3.
* mkswap and swapon ommand in Linux is used to activate a swap space or swap file, allowing it to be used for virtual memory.
* df- to display disk usage by partition.
* du- display disk usage by directory; good for finding users or app that are taking up the most disk spaces.
* iostat: Displays a real-time chart of disk statistics by partition.
* lsblk: Displays current partition sizes and mount points.
* Unmount or mount /media/cdrom (if in fstab)
* mount -t vfat /dev/sdb1 /media/usbdisk (not in fstab)
* ps and top: viewing process.
* yum or apt-get install/update/upgrade/remove/clean/list are package manage to manage packages.
* ps -ef (e= all process. F=full format)
* free -h( show memory usage)
* kill <pid> to kill process, sigterm is the safest and default way to kill process.
* nice priority numder (ranges from -20 to +19) process; to create new process with defined priorities.
* renice priority numder (ranges from -20 to +19) process; to change priority.
* mount | grep '^/dev/': command in Linux is used to list all currently mounted filesystems and then filter the output to display only those filesystems whose device names start with "/dev/"
* lvs: command in Linux is used to display information about logical volumes within a volume group.

## Network:

* systemctl start NetworkManager.service ; to start the network service or internet in linux.
* Change the hostname: hostname hostname.example.com; for transient change, and for persistent change; edit the /etc/hostname file directly and reboot the sys.
* to display and manage network configuration: Ipconfig for win and Ifconifg for other os:
* Open /etc/sysconfig/network-scripts/ifcfg-ensXX file using an editor to modify network configuration.
* to get the default gateway: Route -n get default.
* to add gateway: route add default gw <address>
* to add ip address: ifconfig <ethernet name> <ipadd> netmask <netmask add
* To deactivate the connection ifdown ensxx or nmcli con down ensxx
* to bring the connection up ifup ensxx or nmcli con up ensxx
* to check if port 80 is open on a server with IP address 192.168.1.100, you would use: nc -zv 192.168.1.100 80

To configure DHCP:

* first isntall dhcp-server then edit the real /etc/dhcp/dhcpd.conf file and then Start the dhcp service➔systemctl start dhcpd
* To View DHCP database➔/var/lib/dhcpd/dhcpd.leases

### DNS:

* to install dns server; dns install bind, update the name server configuration file; /etc/named.conf, create the zone file by copying the existing zone file in /var/named directory, then check configuration file with named-checkconf and named-checkzone command and then start the named server.
* /var/log/messages to check log file.
* Test server with dig command.
* /etc/resolv.conf; to get the host name resolution.

## File:

* To create file – nono,
* find command; flexible search by filename, owner, size, permissions, etc.
* locate command; based on database that is updated by updated (usually once a day)
* whereis command; locates the binary source, and manual page files for a command.
* which command; searches the binary, source, and manual page files for a command.
* “ftp” ,”scp” or “rsync” ,and “cp” to copy file from one machine to other.
* a file find the count of lines containing word “ABC”; grep –c “ABC” file1
* egrep is Extended grep that supports added grep features like “+” (1 or more occurrence of previous character),”?”(0 or 1 occurrence of previous character) and “|” (alternate matching).
* To change the file permission; chmod
* To print the file; lp or lpr
* Command line filters; tail, head, and sort
* Vi, vim; text file editor.
* To prints newline, word, and byte counts for each file; wc
* To display hidden files in a directory; ls -a

### Inode:

* Each file (or directory) has an i-node! The i-node (or inode) contains all the file’s information other than its actual content.
* The actual content is stored in a certain number of blocks; their addresses in the partition are in the i-node.
* When you refer to a file’s name in a command, the system goes to the corresponding i-node
* Each i-node is uniquely identified by an i-node number.
* To view the inode of a directory; ls -i

### Directory:

* pwd; it displays the absolute path of the working directory
* cd: to change directory.
* mkdir; to make directory.
* rmdir; to remove directory except empty directory.
* To remove non-empty directories and sub-directories; rm -rf <path>
* /bin: This directory contains basic commands and programs that are needed to achieve a minimal working environment upon booting.
* /boot: This directory contains the actual files, images, and kernels necessary to boot the system.
* /dev: This directory houses the files that represent devices on your system. Every hard drive, terminal device, input or output device available to the system is represented by a file here.
* /etc: This directory is basically a configuration directory for various system-wide services.
* /home: This location contains the home directories of all of the users on the system (except for the administrative user, root).
* /lib: This directory is used for all of the shared system libraries that are required by the /bin and /sbin directories.
* /lost+found: This is a special directory that contains files recovered by /fsck, the Linux filesystem repair program.
* /media: This directory is typically empty at boot. Its real purpose is simply to provide a location to mount removable media (like cds).
* /mnt: This directory is similar to the /media directory in that it exists only to serve as a organization mount point for devices.
* /opt: This directory's usage is rather ambiguous. It is used by some distributions but ignored by others. Typically, it is used to store optional packages.
* /proc: The /proc directory is actually more than just a regular directory. It is actually a pseudo-filesystem of its own that is mounted to that directory.
* /root: This is the home directory of the administrative user (called "root"). It functions exactly like the normal home directories, but is housed here instead.
* /sbin: This directory is much like the /bin directory in that it contains programs deemed essential for using the operating system.
* /tmp: This is a directory that is used to store temporary files on the system. It is writable by anyone on the computer and does not persist upon reboot. This means that any files that you need just for a little bit can be put here. They will be automatically deleted once the system shuts down.
* /usr: This directory is one of the largest directories on the system. It basically includes a set of folders that look similar to those in the root / directory, such as /usr/bin and /usr/lib. This location is basically used to store all non-essential programs, their documentation, libraries, and other data that is not required for the most minimal usage of the system. This is where most of the files on the system will be stored. Some important subdirectories are /usr/local, which is an alternative to the /opt directory for storing locally compiled programs. Another interesting thing to check out is the /usr/share directory, which contains documentation, configuration files, and other useful files.
* /var: This directory is supposed to contain variable data. In practice, this means it is used to contain information or directories that you expect to grow as the system is used. For example, system logs and backups are housed here. Another popular use of this directory is to store web content if you are operating a web server.

## Backup and Recover:

Restore entire filesystem after crash (rare), restore individual files after user accidently delete them (more common).

### Backup Tools:

* **rsync**:

- Example: `rsync -avz /source/directory/ /destination/directory/`

- Explanation: Rsync is a versatile command-line tool for synchronizing files and directories between two locations. It efficiently transfers only the differences between source and destination, minimizing data transfer over networks. The `-avz` options enable archive mode (preserving permissions and ownership), verbose output, and compression for data transfer.

* **tar** :

- Example: `tar -czvf backup.tar.gz /path/to/directory`

- Explanation: Tar is a widely used archiving utility that creates compressed archives of files and directories. In this example, `-czvf` options create a gzip-compressed archive (`-c`), specify compression (`-z`), enable verbose mode (`-v`), and specify the archive filename (`-f`). The source directory is `/path/to/directory`.

* **Rsnapshot** :

- Example: `rsnapshot daily`

- Explanation: Rsnapshot is a backup utility that uses rsync and hard links to create incremental backups. It simplifies backup management by allowing users to define backup schedules (e.g., daily, weekly) in its configuration file. Running `rsnapshot daily` initiates the daily backup process according to the configured schedule.

* **rsync + cron** :

- Example: `rsync -avz /source/directory/ /backup/location/`

- Explanation: Combining rsync with cron, the Unix job scheduler, allows users to automate regular backups. By scheduling a cron job to run the rsync command at specified intervals, such as daily or weekly, users can ensure that their data is regularly backed up to the designated backup location.

* **dd**:

- Example: `dd if=/dev/sda of=/backup/location/backup.img`

- Explanation: The `dd` command is a versatile tool for low-level copying and conversion of data. In this example, `if` specifies the input file or device (`/dev/sda`, the entire disk), and `of` specifies the output file (`/backup/location/backup.img`, the backup image). This command creates a disk image backup of the entire disk `/dev/sda`.

* **Cpio**:

- Example: `find /path/to/directory -depth -print0 | cpio --null -ov --format=ustar > /backup/location/backup.cpio`

- Explanation: cpio is a command-line tool used for creating and extracting archives. In this example, `find` command is used to generate a list of files and directories under `/path/to/directory`, which is then piped (`|`) to cpio. The `-depth` option ensures that files and directories are processed in depth-first order. `--null` option tells cpio to use null-terminated file names. `-ov` option specifies cpio to create a new archive. `--format=ustar` option specifies the archive format as ustar. Finally, `>` redirects the output to a file named `backup.cpio` located at `/backup/location/`.

### Automatic Backup:

Automating backups or running any job in the future. Three tools: cron, anacron and at

* **cron** = run commands at regular intervals.
  + The software utility cron is a time-based job scheduler in Unix-like computer  
    operating systems. Use cron to schedule jobs (commands or shell scripts) to run periodically at fixed times. It is a system daemon which can be started up at thebeginning, it will read its own conf file /etc/crontab (cron tables)
  + Reads /etc/crontab, /etc/cron.d and user's own crontab file (crontab –l  
    command – also see other crontab commands)  
     /etc/crontab often runs scripts in /etc/cron.daily, /etc/cron.weekly,  
     /etc/cron.monthly, /etc/cron.hourly
  + /etc/crontab (not recommend editing directly): check by man 5 crontab  
    02 4 \* \* \* root /bin/somecommand  
     minute, hour, day-of-month, month, day-of-week, owner, command
  + User's own crontab is similar, but without the username crontab -e
  + User access is controlled with /etc/cron.allow and /etc/cron.deny
* **anacron** = catch up on jobs missed by cron
* cron runs jobs at specified times, but if the computer is turned off when a cron job is scheduled, it does not run.
* anacron can ensure that regular maintenance tasks are performed at reasonable  
  intervals on workstations, e.g., once 7 days.
* On the day when the backup.sh job is supposed to be executed, if the system is  
  down for some reason, anacron will execute the backup.sh script 15 minutes after the system comes back up (without having to wait for another 7 days).
* configured in /etc/anacrontab
* 7 15 test.daily /bin/sh /home/sathiya/backup.sh
* period delay job-identifier command
* Period: number of days, delay: delay in minutes to execute the command after machine starts up, job-identifier: a file  
  containing one line indicating last time when this job was executed, command: Command or shell script that needs to be  
  executed.
* **at** = run a command once in the future.
* at -f /usr/local/bin/myscript 5pm Friday
* atq (shows jobs in the at queue)
* atrm 3 (removes job 3 from the at queue)
* Access is controlled with /etc/at.allow and /etc/at.deny
* at command can be useful for shutdown system at specified time, taking one time backup, sending email as reminder at specified time etc.
* **Systemd Timers**:
* Systemd timers are a feature of systemd, the system and service manager for Linux. They provide a more modern alternative to cron for scheduling tasks, including automated backups.
* **Windows Task Scheduler**:
* Windows Task Scheduler is a built-in utility in Microsoft Windows that enables users to schedule tasks to run automatically at specified times or intervals. It can be used to schedule backup scripts or backup software to run at desired times.
* **Automator (macOS)**:
* Automator is a built-in application in macOS that allows users to automate repetitive tasks by creating workflows. It can be used to create automated backup workflows and schedule them to run at specified times.

### Backup Recovery:

Important to verify backups:

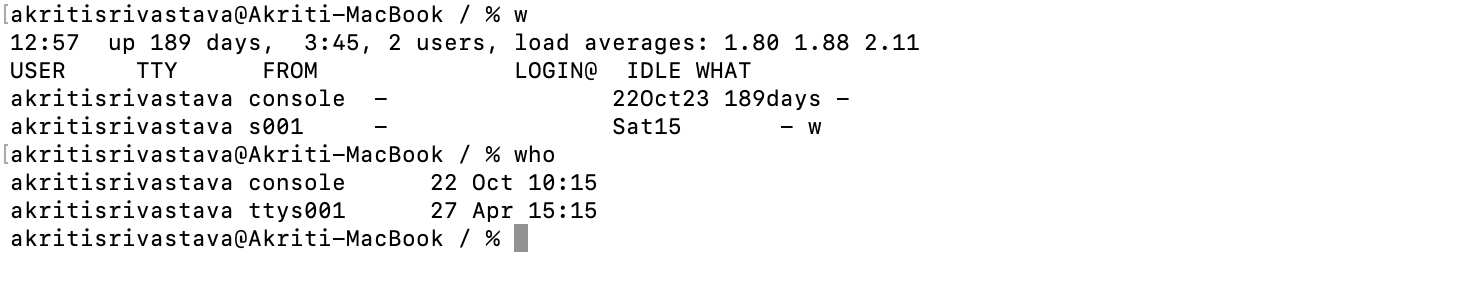
* Periodic checks that files can be restored  
   At least once a fortnight, restore a random file  
   At least once a month, restore a whole filesystem.
* Bonus: retain skills for when they're needed

Store backup media carefully:

* Offsite and onsite storage essential  
   Offsite for disaster recovery (how often to send tapes offsite?)  
   Onsite for quick restoration after lesser disasters.
* Long-term storage of tapes requires care  
   Tapes must be exercised, re-written occasionally, kept in controlled conditions.
* How long to keep? Legal/organisational requirements

## User:

* Command to print user, and group information: id.
* Use w, who command to see logged in users.



TTY: Direct login, PTS: remote login

* To add, modify and delete the user; useradd/usermod/userdel <username>
* To add, modify and delete the group; groupadd/groupmod/groupdel <groupname>
* To login as new user; ssh <username@localhost>
* Files:

|  |  |
| --- | --- |
| /etc/passwd | Original unix user data file; readable by everybody by writable only by root. |
| /etc/shadow | A sys file only readable by root |
| /etc/group | To get group information |
| /etc/skeleton | Skeleton files |
| /etc/issue | Local login messages |
| /etc/issue.net | n/w login msg |
| /etc/motd | Message of the day |
| /sbin/login | Local user account |

* sudo chown -R username:group directory; will change ownership (both user and group) of all files and directories inside of directory and directory itself.
* sudo chown username:group directory; will only change the permission of the folder directory but will leave the files and folders inside the directory alone.
* To lock an account; usermod -l <username>
* To unlock an account; usermod -l <username>