Linux SysOps Handbook

An essentials notebook for the common knowledge and tasks of a Linux system admin.

Table of Content

Processes

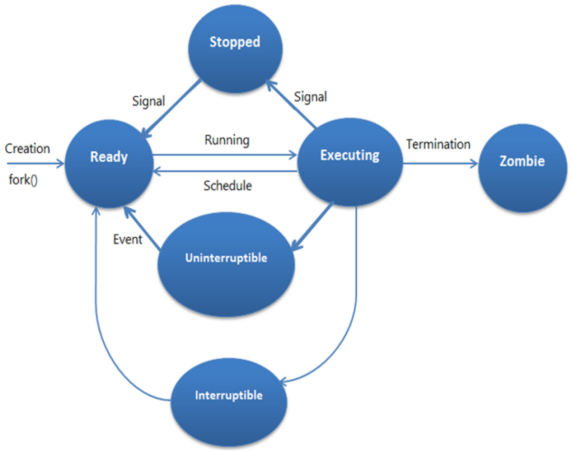
List the current active process with their statuses, numbers, resource usage, etc. using the command `ps`.

```
$ ps auxc
```

Quoting man's page documentation on `ps`: "A different set of processes can be selected for display by using any combination of the `-a, -G, -g, -p, -T, -t, -U, and -u` options. If more than one of these options are given, then ps will select all processes which are matched by at least one of the given options".

The daemon `systemd` process starts during boot time, and remains active until the shutdown. It's the parent process for all other process in the system.

Each process contains several main parts, such as: PID, state, virtual space address (memory), threads, network and file descriptors, scheduler information, and links. Processes are controlled and respond to signals. The states that a process can transition among are depicted below:



To observe the states and other information of the processes interactively, use the `top` command.

To run executables as background process (job), append an ampersand to it:

```
$ echo "Hi .. looping:" | sleep 10000 | echo "done." &
```

To view the current jobs, and their details run `job`, `ps j` commands respectively.

To bring back a job in the foreground in the current session, and send it back use the following:

```
$ fg %<job-no>
$ ctrl+z
$ bg %<job-no>
```

Use the command `kill -l` to see the available signals to send to processes, like interrupt, terminate, resume, etc.

```
$ kill -l
$ kill -9 5921
$ kill -SIGTERM 6152
```

Use `killall` to operate on multiple processes using their executable name. Use `pkill` for filtering with more options.

```
$ killall -15 nginx
$ pkill -U tester
```

Finally, use `pstree` and `pgrep` to view process parent/child tree and search for processes by pattern.

```
$ psgrep -u abdullah -l
```

User Management

The users and groups are managed in `/etc/passwd` and `/etc/group` files.

```
$ tail /etc/passwd
$ tail /etc/group
$ tail /etc/shadow
```

The commands to manage a user are as follows:

- `useradd`
- `usermod`
- `userdel`

And for groups:

- `groupadd`
- `groupmod`
- `groupdel`

Each user in the system is associated with unique user id `uid`, and each group is associated with `gid`.

```
$ id abdullah
```

Use flags `-g` and `-aG` for users to replace group or append group, respectively:

```
$ sudo usermod -G admins abdullah
$ sudo usermod -aG staff abdullah
```

To lock or unlock a user account, us the `-L`, `-U` options respectively.

```
$ usermod -L <username>
$ usermod -U <username>
```

To restrict service user accounts (e.g. accounts for web servers), the shell can be set to `nologin`:

```
$ usermod -s /sbin/nologin nginx_usr1
```

To change a user password, use the command `passwd` interactively. Additionally `change` command sets the password policy in the system.

Use the command `su - <username>` to switch to the specified user. which will promote for her password. Running the command without username will switch to the root user. To avoid cases where password is not available, use `sudo` to switch accounts using current user password only and according to rules in `/etc/sudoers` directory. Use `sudo -i` to gain an interactive root shell.

Shell Tips and Tricks

Getting used to bash language and its fundamentals like conditions, looping, functions, etc. is recommended.

The popular files and text processing and manipulation utilities are important to master, such as:

- `cat`
- `cp`
- `rm`
- `mkdir`
- `rmdir`
- `touch`
- `less`
- `more`
- `head`
- `tail`
- `grep`
- `find`
- `locate`
- `wc`
- `sed`

Use the command `date` to print the current date and time or others in the past and future:

```
$ date +%x
```

The standard terminal channels in Linux are 3: `stdin`, `stdout`, and `stderr` where the first is for input stream and the latters for output and error streams.

By default the successful command results are outputted to `stdout` (equivalent to `>`). You can explicity redirect to `stdout` or `stderr` as follows:

```
$ echo "hi there!" 1> error_log.txt
$ cat ~/incorrect-path 2> error_log.txt
# To both:
$ (echo "hi" && cat ~/wrong) >> log.txt 2>&1
```

To discard output stream, redirect it to the special directory `/dev/null`.

The standard input can be captured via redirection or file pipes:

```
$ cat <<EOF
This is coming from the stdin
EOF

$ cat LICENSE | wc -l
```

The `ssh` command used to connect to servers in secure manner using OpenSSH library using public key cryptography. The configuration and known hosts are kept under `/etc/ssh` system-wide or in `~/.ssh/` in current user's home directory. On the other hand `scp` is used for secure copy on secure shell fashion.

The following list of commands are used to generate and manage ssh keys between client and server:

1. `ssh-keygen` : to generate new key pairs.
2. `ssh-copy-id` : to copy the public key to the remote machines.
3. `ssh-agent` : to simplify working with the private key passphrase if used.
4. `ssh-add` : to cache the passphrase in the current session.

File Permissions

A file permissions are considered in three dimensions: the owner user, the owner's group, and rest of other users.

Showing the permisison of files and directories can be using `ls -l`, `ls -ld` respectively.

The basic permission types are: read (r), write (w), and execute (x) on both folders and files:

```
$ ls -l
-rw-r--r--  1 abdullah  staff  35149 Jan 30 17:20 LICENSE
```

Setting the files and folders permission is done by `chmod` command and can be using symbols or digits.

The symbols/letter way is made for `u`, `g`, `o`, or `a` basis for the user, group, others, or all. Whereas, the digits are written for all at once in sequence for user, group, and others. Examples are below for both cases:

```
# Use + to add, - to remove, and = to reset.

# adding execute permission to user
$ chmod u+x my-file.txt
# setting read, execute to all on a folder and its content
$ chmod -R a=rX my-folder

$ chmod 740 special.txt
$ chmod -R 444 read-only-files/
```

`chown` is used to change the ownership of folder/files to users or groups respectively. `chgrp` is a shortcut to group change only. The root or the owner are only people can change ownership and in the latter, she needs to be part of the new target group before the change.

```
$ chown sarah file-10.txt
$ chown sarah:staff file-12.txt

$ chown :admins server_log.txt
$ chgrp operators server_log.txt
```

Lastly, a fourth dimension at the start can be added to represent the special permissions of `suid s`, `sgid s`, and `sticky t` which control executable nature of files to be of owner users, and groups regardless of the current user. The last is to restrict deletion for only the root and owner always.

```
$ chmod a+t protected-folder/
$ chmod -R 1444 read-only-protected/
```

Background Services and Crons

`systemctl` is the command used to list, manage, and check background processes or so called `daemons`.

To list the available categories of daemons, run:

```
$ systemctl -t help
```

There are 3 types of daemons: 1. services, 2. sockets, 3. paths. Use the following to see the system's processes in each:

```
$ systemctl
$ systemctl list-units --type=service
$ systemctl list-units --type=socket --state=LOAD
$ systemctl list-units --type=path --all
$ systemctl list-unit-files
```

The states `enabled` and `disabled` indicate wether a service is lanuched on startup or not. The subcommands `enable` and `disable` can be used to control this aspect.

To view the status of a daemon use the `status` command or its state shortcuts:

```
$ systemctl status kubelet
$ systemctl is-active dockerd
$ systemctl is-enabled sshd.service
```

Use the subcommands `start`, `stop`, `restart`, and `reload`, `reload-or-restart` to control daemons.

Additionally, use the following to list a daemon dependencies:

```
$ systemctl list-dependencies nginx.service
```

Finally, to resolve conflicting services making them unavailable, the `mask` and `unmask` commands can be used to point a deamons config to `dev/null` then back to normal respectively.

The cron daemon `crond` is responsible for managing the user's and system's scheduled jobs. Use the command `crontab` to manage jobs and their files in the user account or in the system wide `/etc/crontab`, `/etc/cron.d/` locations.

```
$ sudo crontab -l
$ sudo crontab -e
$ vim /etc/cron.d/my-backup
```

The syntax of crontab entries is captured by the diagram below. Use the following tool to quick assistance.

An example of a cron entry that runs backup command, every day at 5:00 AM:

```
0 5 * * * /usr/bin/daily-backup
```

Linux Distros

In 1991, Linux kernel was introduced by Linus Torvalds, and combined with GNU project, which was previously created in 1983-1984 as open source OS programs and components. This formed what we call today Linux distribution, a Unix-like operating system.

Today the Linux operating system is supported on most hardware platforms. Linux works on almost every architecture from i386 to SPARC. Linux can be found on almost every type of device today, from watches, televisions, mobile phones, servers, desktops, and even vending machines.

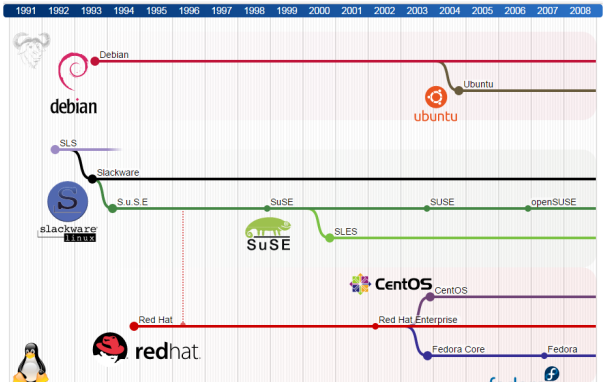
One of the major distinction between Linux distributions is the package management part and how software is installed and managed. There are multiple package formats, and the most common ones are Debian (deb), RedHat Package Manager (RPM).

Here's a listing for the common Debian based distributions:

- Debian.
- Ubuntu.
- Linux Mint.
- Kali Linux.

And here's for RPM based distributions:

- Fedora.
- RedHat Enterprise Linux (RHEL).
- CentOS.
- openSUSE.



Logs, Monitoring, and Troubleshooting

You can monitor the system's resources usage, uptime, and sessions' load leverages over time as follows:

```
$ top
$ uptime
$ w
```

Use `lscpu` to see the system's CPU in use and other details.

The system events and processes traces are usually kept in as logs in `/var/log` directory. There are two categories of logs: 1. essential system logs via `journald`, that are wiped across boots by default (can be configured to persist). 2. `rsyslog` logs that persist by default and organized inside `/var/log/` folder. Mainly, the logging mechanism in Linux follows the standard `syslog` protocol for the system's messages, events, security incidents, mailing, and jobs logs, while other programs may or may not follow `syslog` format identically.

As explored in section (3), use `cat`, `head`, `tail` commands to interactively see or follow the logs.

```
$ head -n 50 /var/logs/mail.log
$ tail -f /var/logs/mysql.log
```

You can configure `rsyslog` service and manage it as any daemon:

```
$ vim /etc/rsyslog.conf
$ systemctl reload rsyslog
```

On the other hand, use `journalctl` to view and follow the system's `journald` log entries, which resides in `/run/log/journal`.

Network Essentials

For effective work on the system network configurations and troubleshooting, it is essential to review network/internet protocols (TCP/UDP) and IPv4/IPv6 concepts (Ref.1), (Ref.2).

See the hostname of current machine or set it as below:

```
$ hostname
$ hostnamectl set-hostname rhel.n1.apps.com
```

The host name is managed under `/etc/hostname` .

The host connection is either managed dynamically (`DHCP`) configured in `/etc/resolv.conf` or manually in `/etc/hosts` file.

The `ping` utiltiy helps for connectivity checking:

```
$ ping 172.168.9.13
$ ping -c4 github.com
$ ping6 2001:db8:3333:4444:5555:6666:7777:8888
```

To see the network routing table and interfaces, use the following:

```
$ ip route
$ ip -6 route
$ ip help
$ ip show link
```

Use the command `nmap` for advanced network investigation and security monitor and scan.

```
# Scan a single ip address
$ nmap 192.168.1.1

# Scan a host name
$ nmap -v server1.cyberciti.biz

# View open ports:
$ nmap --open 192.168.2.18

# Trace all pakets:
$ nmap --packet-trace 192.168.1.1
```

`NetworkManager` is the kernel feature to manage network configurations in Linux. `nmcli` is the terminal utility

```
$ nmcli device wifi list
$ nmcli dev status
$ nmcli general hostname centos-8.cluster.internal
$ nmcli con show
```

System Updates and Patching

Managing the system packages varies depending on Linux distributions, but the essential parts are the same (installation, repositories, package managers, etc.). For Debian based distributions, `apt` is the package manager, whereas for Fedora / RHEL, `yum` is used.

Search for some package:

```
$ apt search <KEYWORD>
$ yum search <KEYWORD>
```

Install a package:

```
$ apt install <NAME>
$ yum install <NAME>
```

Update a package or all packages:

```
$ apt upgrade <NAME>
$ yum update <NAME>
```

Remove a package:

```
$ apt remove <NAME>
$ yum remove <NAME>
```

Show details on a package:

```
$ apt show <NAME>
$ yum info <NAME>
```

List all current packages on the system:

```
$ apt list --installed
$ yum list
```

Audit the history of package management actions:

```
$ cat less /var/log/apt/history.log | less
$ cat less /var/log/dnf.rpm.log | less
$ yum history
```

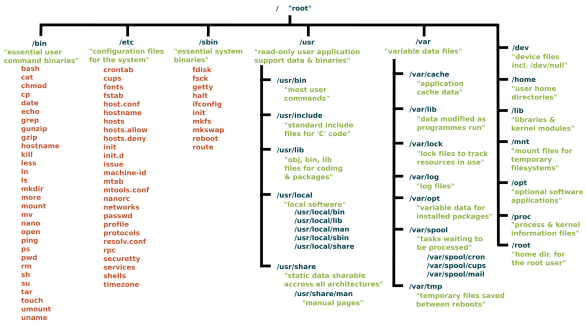
And finally, the package source repositories can be set up and updated through the following:

```
# list current enabled repos
$ yum repolist all
$ apt-cache policy

# manage and add repos in these directories:
$ cat /etc/apt/sources.list /etc/apt/sources.list.d/*
$ cat /etc/yum.repos.d/*
```

Storage

Linux is formed for a unified file-system consists of all file systems provided by the hardware or virtual storage devices attached to the system. Essentially, everything in Linux is a file. It can be viewed as a reversed tree of nested directories starting from the root directory `/`.



Block devices are the mechanism that the kernel detects and identify raw storage devices (HDD, SSD, USBs, ..). As the name indicates, the kernel interfaces and references them by fixed-size blocks (chunks of spaces). The block devices are stored in `/dev` directory by the OS, and has letters naming convention such as `/dev/sda`, `/dev/sdb`, `/dev/vda`, and appended numbers in case of partitions `/dev/sda3`. The attachment of the block device into the system is done through mounting it to a directory in the system.

Two operations are essential for using block storage:

1. Partitioning:

Breaking the disk into reusable smaller units, each treated as own disk. The main partitioning methods are MBR (Master Boot Record) and GPT (GUID Partition Table). Use `parted` or equivalent commands

2. Formatting:

Preparing the device as a file-system to be read and write to. Many file-system formats exists like:

- `Ext4`.
- `XFS`.
- `Btrfs`.
- `ZFS`.

Additionally, LVM concepts focus on building more extensible storage layout by grouping physical volumes (PV) into logical groups (VG), then creating logical volumes from, with possibility of extending or reduction later on.

To see the block devices and currently attached file system with mounts, and disk usage:

```
$ blkid
$ mount
$ df -h
$ du -h /opt/data
```

The `lsof` command lists all active processes using the block device.

The permanent mounting process rely on `/etc/fstab` file to determine devices to mount on the boot time.

Use the commands `lsblk`, `mount`, and `unmount` to check and mount filesystem devices, respectively.

Notes and Additional Resources

Use the `man` command to lookup the manual information on commands or topics in the system.

Additionally, the `info` command is the GNU documentation tool and provide more detailed materials.

Both provide shortcuts, navigation, and searching capabilities (e.g. `man -K <keyword>` to search across manual).

Recommended Reading List

Books:

1. How Linux Works What Every Superuser Should Know, Brian Ward, *2nd Edition, No Starch Press*..
2. Linux Command Line and Shell Scripting Bible, R. Blum and C. Bresnahan, *3rd Edition, Wiley*.
3. Linux Bible, Christopher Negus, *9th Edition, Wiley*

Websites & Blogs:

1. Digital Ocean Knowledge Hub.
2. 9 to 5 Linux Blog.
3. nixCraft.

License

Last modified 5mo ago