

Unit-4 Assignment:

1.What are the various system units? Give the significance of any three.

Unit Type	File Extension	Description
Service unit	. service	A system service.
Target unit	. target	A group of systemd units.
Automount unit	. automount	A file system automount point.
Device unit	. device	A device file recognized by the kernel.
Mount unit	. mount	A file system mount point.
Path unit	. path	A file or directory in a file system.
Scope unit	. scope	An externally created process.
Slice unit	. slice	A group of hierarchically organized units that manage system processes.
Snapshot unit	. snapshot	A saved state of the systemd manager.
Socket unit	. socket	An inter-process communication socket.
Swap unit	. swap	A swap device or a swap file.
Timer unit	. timer	A systemd timer.

1.Service unit:

* Specifying Service Units

For clarity, all command examples in the rest of this section use full unit names with the . service file extension, for example:

systemctl stop nfs-server.service

* Listing Services

To list all currently loaded service units, type the following at a shell prompt:

systemctl list-units --type service

* Displaying Service Status

To display detailed information about a service unit that corresponds to a system service, type the following at a shell prompt:

systemctl status name.service

2.Target Unit

* Viewing the Default Target

systemctl get-default graphical.target

* Viewing the Current Target

To list all currently loaded target units, type the following command at a shell prompt:

systemctl list-units --type target

* Changing the Default Target

systemctl set-default name.target

Unit-4 Assignment:

3.Device Unit

1) What is systemd? What are the features of systemd?

Systemd is a system and service manager for Linux operating systems. It is designed to be backwards compatible with SysV init scripts, and provides a number of features such as parallel startup of system services at boot time, on-demand activation of daemons, support for system state snapshots, or dependency-based service control logic.

Features

- **Socket-based activation** — At boot time, systemd creates listening sockets for all system services that support this type of activation, and passes the sockets to these services as soon as they are started. This not only allows systemd to start services in parallel, but also makes it possible to restart a service without losing any message sent to it while it is unavailable: the corresponding Chapter 8 . Managing Services with systemd 1 socket remains accessible and all messages are queued.
- **Bus-based activation** — System services that use D-Bus for inter-process communication can be started on-demand the first time a client application attempts to communicate with them. Systemd uses D-Bus service files for bus-based activation.
- **Device-based activation** — System services that support device-based activation can be started ondemand when a particular type of hardware is plugged in or becomes available. Systemd uses device units for device-based activation.
- **Path-based activation** — System services that support path-based activation can be started ondemand when a particular file or directory changes its state. Systemd uses path units for path- based activation
- **System state snapshots** — Systemd can temporarily save the current state of all units or restore a previous state of the system from a dynamically created snapshot. To store the current state of the system, systemd uses dynamically created snapshot units.
- **Mount and automount point management** — Systemd monitors and manages mount and automount points. Systemd uses mount units for mount points and automount units for automount points.
- **Aggressive parallelization** — Because of the use of socket-based activation, systemd can start system services in parallel as soon as all listening sockets are in place. In combination with system services that support on-demand activation, parallel activation significantly reduces the time required to boot the system.
- **Transactional unit activation logic** — Before activating or deactivating a unit, systemd calculates its dependencies, creates a temporary transaction, and verifies that this transaction is consistent. If a transaction is inconsistent, systemd automatically attempts to correct it and remove non-essential jobs from it before reporting an error.

Unit-4 Assignment:

- 2) **Give the command to get the status of any service. Also give different descriptions on its output.**

●To display detailed information about a service unit that corresponds to a system service

systemctl status name.service

●Replace name with the name of the service unit you want to inspect
●command displays the name of the selected service unit followed by its short description -- like

Loaded

Active

Main PID

Status

Process

CGroup

Description on its output

●gdm.service - GNOME Display Manager

●Loaded: loaded (/usr/lib/systemd/system/gdm.service; enabled)

●Active: active (running) since Thu 2013-10-17 17:31:23 CEST; 5min ago

●Main PID: 1029 (gdm)

●CGroup: /system.slice/gdm.service – └─1029 /usr/sbin/gdm – └─1037 /usr/libexec/gdm-simple-slave --display-id – /org/gno... – └─1047 /usr/bin/Xorg :0 - background none -verbose -auth – /r... –Oct 17 17:31:23 localhost systemd[1]: Started GNOME Display Manager

- 3) **Give the procedure to check whether gdm service is enabled or not. If not start and then stop the service.**

*To configure a service unit that corresponds to a system service to be automatically started at boot time, type the following at a shell prompt as root:

systemctl enable name.service

*Replace name with the name of the service unit you want to enable (for example, httpd). This command reads the [Install] section of the selected service unit and creates appropriate symbolic links to the /usr/lib/systemd/system/name.

service file in the /etc/systemd/system/ directory and its subdirectories. This command does not, however, rewrite links that already exist. If you want to ensure that the symbolic links are re-created, use the following command as root:

systemctl reenablename.service

*To configure the Apache HTTP Server to start automatically at boot time, run the following command as root

Unit-4 Assignment:

systemctl enable httpd.service

Disable the service

To prevent a service unit that corresponds to a system service from being automatically started at boot time, type the following at a shell prompt as root:

systemctl disable name.service

Replace name with the name of the service unit you want to disable (for example, bluetooth). This command reads the [Install] section of the selected service unit and removes appropriate symbolic links to the /usr/lib/systemd/system/name.service file from the /etc/systemd/system/ directory and its subdirectories. In addition, you can mask any service unit to prevent it from being started manually or by another service. To do so, run the following command as root:

systemctl mask name.service

This command replaces the /etc/systemd/system/name.service file with a symbolic link to /dev/null, rendering the actual unit file inaccessible to systemd. To revert this action and unmask a service unit, type as root:

systemctl unmask name.service

4) Give the difference between following commands:

systemctl enable httpd.service and **systemctl start httpd.service**.

=>To configure a service unit that corresponds to a system service to be automatically started at boot time, type the following at a shell prompt as root:

systemctl enable name.service

*Replace name with the name of the service unit you want to enable (for example, httpd). This command reads the [Install] section of the selected service unit and creates appropriate symbolic links to the /usr/lib/systemd/system/name.

service file in the /etc/systemd/system/ directory and its subdirectories. This command does not, however, rewrite links that already exist. If you want to ensure that the symbolic links are re-created, use the following command as root:

systemctl reenablename.service

*To configure the Apache HTTP Server to start automatically at boot time, run the following command as root

systemctl enable httpd.service

=>To start a service unit that corresponds to a system service, type the following at a shell prompt as root:

systemctl start name.service

Example 8.5. Starting a Service

The service unit for the Apache HTTP Server is named httpd.service. To activate this service unit and start the httpd daemon in the current session, run the following command as root:

Unit-4 Assignment:

systemctl start httpd . service

5) Give the difference between following commands:

systemctl restart httpd.service and **systemctl try-restart httpd.service.**

To restart a service unit that corresponds to a system service, type the following at a shell prompt as root:

systemctl restart name. service

Replace name with the name of the service unit you want to restart (for example, httpd). This command stops the selected service unit in the current session and immediately starts it again. Importantly, if the selected service unit is not running, this command starts it too. To tell systemd to restart a service unit only if the corresponding service is already running, run the following command as root:

systemctl try-restart name. service

*Note that system services that do not support this feature ignore this command altogether. For convenience, the systemctl command also supports the reload -o r-restart and reload -o r-try-restart commands that restart such services instead. For information on how to determine the status of a certain service unit

*Ex:Restarting a Service

In order to prevent users from encountering unnecessary error messages or partially rendered web pages, the Apache HTTP Server allows you to edit and reload its configuration without the need to restart it and interrupt actively processed requests. To do so, type the following at a shell prompt as root:

systemctl reload httpd . service

6) What is the significance of systemctl reload httpd.service command?

How it differs from systemctl restart httpd.service

In order to prevent users from encountering unnecessary error messages or partially rendered web pages, the Apache HTTP Server allows you to edit and reload its configuration without the need to restart it and interrupt actively processed requests. To do so, type the following at a shell prompt as root:

systemctl reload httpd . service

To restart a service unit that corresponds to a system service, type the following at a shell prompt as root:

systemctl restart name. service

Replace name with the name of the service unit you want to restart (for example, httpd). This command stops the selected service unit in the current session and immediately starts it again. Importantly, if the selected service unit is not running, this command starts it too. To tell systemd to restart a service unit only if the corresponding service is already running, run the following command as root:

systemctl try-restart name. service

*Note that system services that do not support this feature ignore this command altogether. For convenience, the systemctl command also supports the reload -o r-restart and reload -o r-try-restart commands that restart such services instead. For information on how to determine the status of a certain service unit

Unit-4 Assignment:

7) How will you find default target unit for linux? How to change it to another target unit?

To determine which target unit is used by default, run the following command:

```
systemctl get-default
```

Example : Viewing the Default Target

To display the default target unit, type:

```
systemctl get -default
```

Changing the Default Target

To configure the system to use a different target unit by default, type the following at a shell prompt as root:

```
systemctl set -default name. target
```

Replace name with the name of the target unit you want to use by default (for example, multi-user). This command replaces the /etc/systemd/system/default.target file with a symbolic link to /usr/lib/systemd/system/name.target, where name is the name of the target unit you

Example :

Changing the Default Target To configure the system to use the multi-user.target unit by default, run the following command as root:

```
systemctl set -default multi-user.target
```

8) What is system targets in RedHat Linux? Name the different target units and its significance

Previous versions of Red Hat Enterprise Linux, which were distributed with SysV init or Upstart, implemented a predefined set of runlevels that represented specific modes of operation. These runlevels were numbered from 0 to 6 and were defined by a selection of system services to be run when a particular runlevel was enabled by the system administrator. In Red Hat Enterprise Linux 7, the concept of runlevels has been replaced with systemd targets.

Comparison of SysV Runlevels with systemd Targets

Unit-4 Assignment:

Runlevel	Target Units	Description
0	runl evel 0 . targ et,po wero ff. targ et	Shut down and power off the system.
1	runl evel 1. targ et,rescue. targ et	Set up a rescue shell.
2	runl evel 2. targ et,mul ti - user. targ et	Set up a non-graphical multi-user system.
3	runl evel 3. targ et,mul ti - user. targ et	Set up a non-graphical multi-user system.
4	runl evel 4 . targ et,mul ti - user. targ et	Set up a non-graphical multi-user system.
5	runl evel 5. targ et,g raphi cal . targ et	Set up a graphical multi-user system.
6	runl evel 6 . targ et,rebo o t. targ et	Shut down and reboot the system.

Significance

- *Systemd targets are represented by target units. Target units end with the .
- * target file extension and their only purpose is to group together other systemd units through a chain of dependencies.
- *For example, the **graphical . target** unit, which is used to start a graphical session, starts system services such as the GNOME Display Manager (g d m. servi ce) or Accounts Service (**accounts- daemon. servi ce**) and also activates the **multi -user. target** unit. Similarly, the **multi - user. target** unit starts other essential system services such as NetworkManager (Netwo rkManag er. servi ce) or D-Bus (d bus. servi ce) and activates another target unit named basi c. target.

9) What do you mean by masking a service? Give the procedure to mask a specific service

To prevent a service unit that corresponds to a system service from being automatically started at boot time, type the following at a shell prompt as ro o t:

systemctl disable name. servi ce

Replace name with the name of the service unit you want to disable (for example, bl ueto o th). This command reads the [Instal l] section of the selected service unit and removes appropriate symbolic links to the /usr/l i b/systemd /system/name. servi ce file from the /etc/systemd /system/ directory and its subdirectories. In addition, you can mask any service unit to prevent it from being started manually or by another service. To do so, run the following command as ro o t:

systemctl mask name. service

This command replaces the /etc/systemd /system/name. servi ce file with a symbolic link to /d ev/nul l , rendering the actual unit file inaccessible to systemd. To revert this action and unmask a service unit, type as ro o t:

systemctl unmask name. service

Example

Unit-4 Assignment:

Disabling a Service Example 8.6, “Stopping aService” illustrates how to stop the bluetooth.service unit in the current session. To prevent this service unit from starting at boot time, type the following at a shell prompt as root:

systemctl disable bluetooth.service

10) How to get in to rescue mode and emergency mode in case of RedHat Enterprise Linux? Give its significance.

Rescue Mode

Rescue mode provides a convenient single-user environment and allows you to repair your system in situations when it is unable to complete a regular booting process. In rescue mode, the system attempts to mount all local file systems and start some important system services, but it does not activate network interfaces or allow more users to be logged into the system at the same time.

*To change the current target and enter rescue mode in the current session, type the following at a shell prompt as root:

systemctl rescue

This command is similar to `systemctl isolate rescue.target`, but it also sends an informative message to all users that are currently logged into the system. To prevent systemd from sending this message, run this command with the `--no-wall` command line option:

systemctl --no-wall rescue

Emergency Mode

Emergency mode provides the most minimal environment possible and allows you to repair your system even in situations when the system is unable to enter rescue mode. In emergency mode, the system mounts the root file system only for reading, does not attempt to mount any other local file systems, does not activate network interfaces, and only starts a few essential services. In Red Hat Enterprise Linux 7, emergency mode requires the root password.

To change the current target and enter emergency mode, type the following at a shell prompt as root:

systemctl emergency

This command is similar to `systemctl isolate emergency.target`, but it also sends an informative message to all users that are currently logged into the system. To prevent systemd from sending this message, run this command with the `--no-wall` command line option: **systemctl --no-wall emergency**

11) What is the significance of suspend and hibernate in case of RedHat Enterprise server?

Suspend hibernate

To suspend the system, type the following at a shell prompt as root:

systemctl suspend

* This command saves the system state in RAM and with the exception of the RAM module, powers off most of the devices in the machine.

Unit-4 Assignment:

- * When you turn the machine back on, the system then restores its state from RAM without having to boot again.
- * Because the system state is saved in RAM and not on the hard disk, restoring the system from suspend mode is significantly faster than restoring it from hibernation, but as a consequence, a suspended system state is also vulnerable to power outages.

Hibernate System

To hibernate the system, type the following at a shell prompt as root:

systemctl hibernate

*This command saves the system state on the hard disk drive and powers off the machine.

*When you turn the machine back on, the system then restores its state from the saved data without having to boot again. Because the system state is saved on the hard disk and not in RAM, the machine does not have to maintain electrical power to the RAM module, but as a consequence, restoring the system from hibernation is significantly slower than restoring it from suspend mode.

To hibernate and suspend the system, run the following command as root:

systemctl hybrid -sleep

12) Give the different usage and significance of shutdown command in RedHat Linux

To shut down the system and power off the machine, type the following at a shell prompt as root:

systemctl poweroff

To shut down and halt the system without powering off the machine, run the following command as root:

systemctl halt

By default, running either of these commands causes systemd to send an informative message to all users that are currently logged into the system. To prevent systemd from sending this message, run the selected command with the `--no-wall` command line option, for example:

systemctl --no-wall poweroff

To shut down the system and power off the machine at a certain time, use a command in the following format as root:

shutdown --poweroff hh:mm

To shut down and halt the system after a delay, without powering off the machine, use a command in the following format as root:

shutdown --halt +m

A pending shutdown can be canceled by the root user as follows: **shutdown -c**

Unit-4 Assignment:

13) What is systemd? Give the procedure to manage the remote machine using systemd.

Systemd is a system that is designed specially for the linux kernel it replace the sysvinit process to become the 1st process with PID=1, which gets executes in users space during the linux start a process

Controlling systemd on a Remote Machine

In addition to controlling the systemd system and service manager locally, the systemctl utility also allows you to interact with systemd running on a remote machine over the SSH protocol. Provided that the sshd service on the remote machine is running, you can connect to this machine by running the systemctl command with the --host or -H command line option:

systemctl --host user_name@ host_name command

*Replace user_name with the name of the remote user, host_name with the machine's host name, and command with any of the systemctl commands described above. Note that the remote machine must be configured to allow the selected user remote access over the SSH protocol.

14) Write a note on Unit Files and its structure in Linux.

A unit file contains configuration directives that describe the unit and define its behavior. Several systemctl commands work with unit files in the background.

unit_name.type_extension

- Unit files can be supplemented with a directory for additional configuration files. For example, to add custom configuration options to sshd . service, create the sshd . service.d /custom.conf file and insert additional directives there.
- Many unit file options can be set using the so called unit specifiers – wildcard strings that are dynamically replaced with unit parameters when the unit file is loaded.

Unit files typically consist of three sections:

- [Unit] — contains generic options that are not dependent on the type of the unit. These options provide unit description, specify the unit's behavior, and set dependencies to other units
- [unit type] — if a unit has type-specific directives, these are grouped under a section named after
- [Install] — contains information about unit installation used by systemctl enable and disable commands,