Hands-on Projects

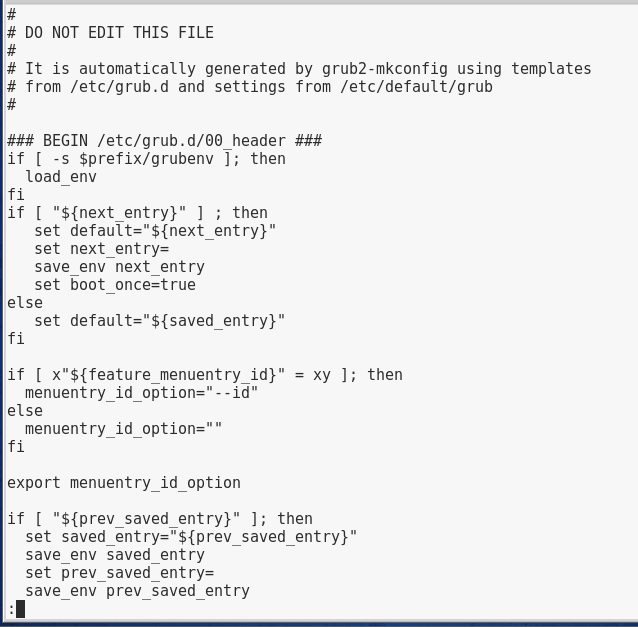
These projects should be completed in the order given. The hands-on projects presented in this chapter should take a total of three hours to complete. The requirements for this lab include:

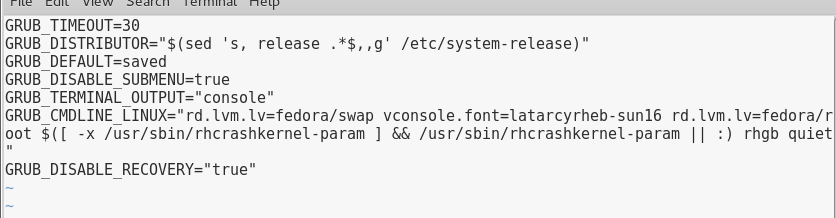
* A computer with Fedora Linux installed according to Hands-on Project 2-1 and Ubuntu Server Linux installed according to Hands-on Project 6-1.

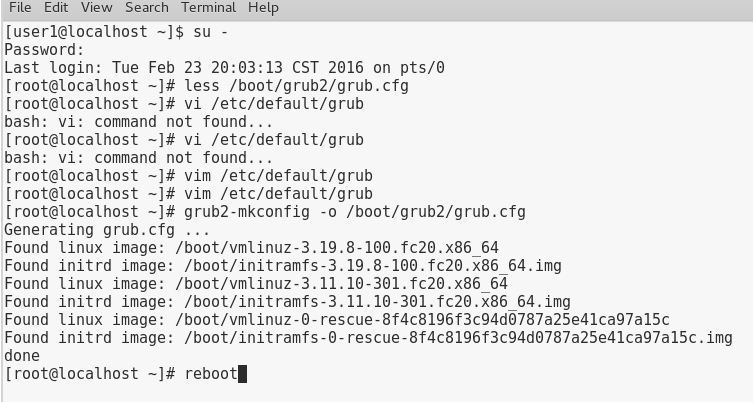
# Project 8-1

In this hands-on project, you use and configure the GRUB2 boot loader on Fedora 20.

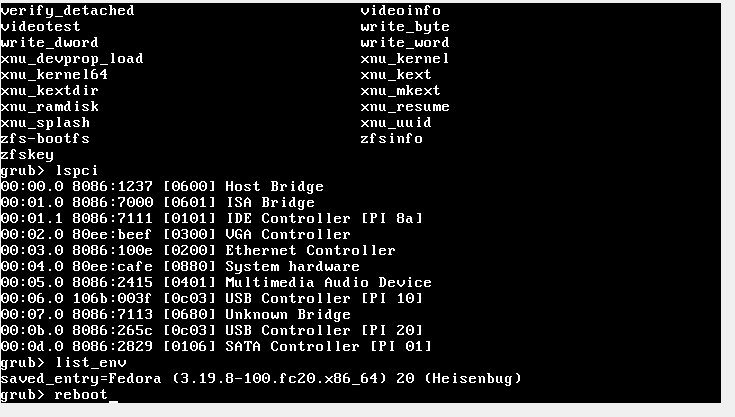
1. Boot your Fedora Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, type **less /boot/grub2/grub.cfg** and press Enter. Review the contents of the grub.cfg file. Press q to exit less.
4. At the command prompt, type **vim /etc/default/grub** and press Enter to edit the GRUB2 configuration file. Change the value of **GRUB\_TIMEOUT** to **30**. Save your changes and quit the vi editor.
5. At the command prompt, type **grub2-mkconfig -o /boot/grub2/grub.cfg** and press Enter to rebuild the GRUB2 configuration file with your change.
6. Reboot your system by typing **reboot** and pressing Enter. At the GRUB2 boot screen, view the available options. You should notice at the bottom right the timeout is now 30 seconds. Next, press **c** to obtain a command prompt.
7. At the **grub>** prompt, type **help** and press Enter to see a list of available commands. Next, type **lspci** and press Enter to see a list of detected PCI devices. Following this, type **list\_env** and press Enter to view the variables present. Normally, there should be a single variable called **saved\_entry** that lists the default OS that is booted by GRUB2, although on some systems you may not see this variable. Type **reboot** and press Enter to reboot your system.
8. At the GRUB2 boot screen, press **e** to edit your configuration.
9. Locate the line that starts with the word “linux” and navigate to the end of this line (the last two keywords on this line should be rhgb and quiet). Add the word single after the word quiet and press F10 to boot your modified configuration. You are now in single user mode.
10. Supply your root password of LNXrocks! when prompted.
11. **Provide screenshot(s) of steps 2 through 10.**

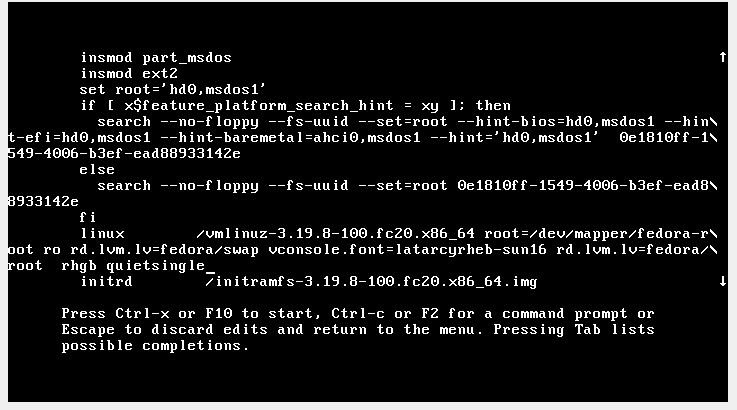
****

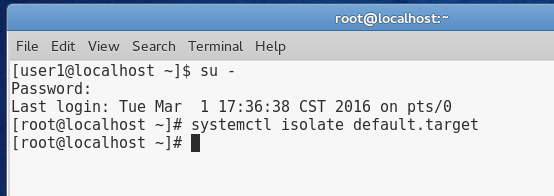
****

****

****

****

****

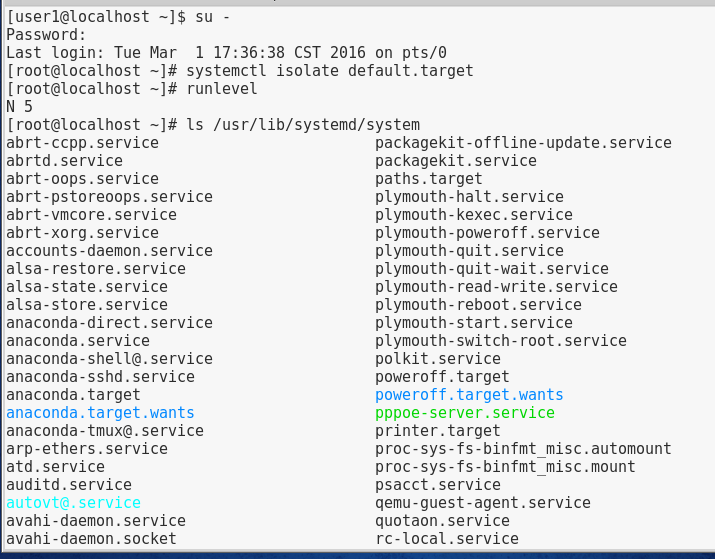
****

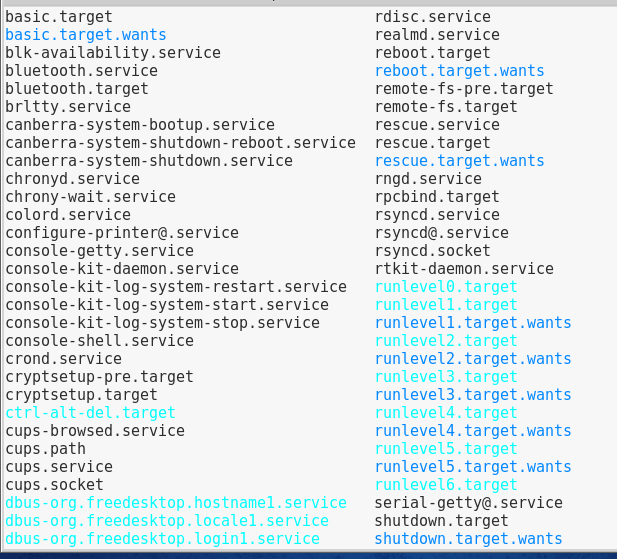
1. At the command line type **systemctl isolate default.target** to load to the default graphical.target.

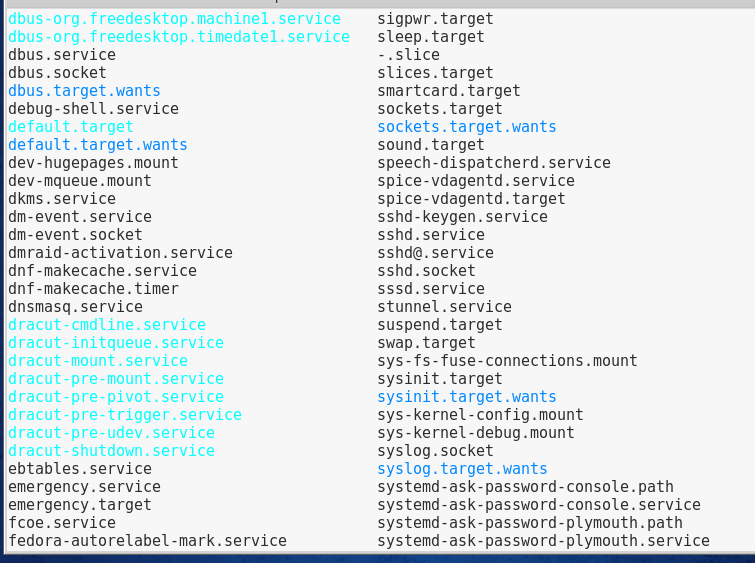
# Project 8-2

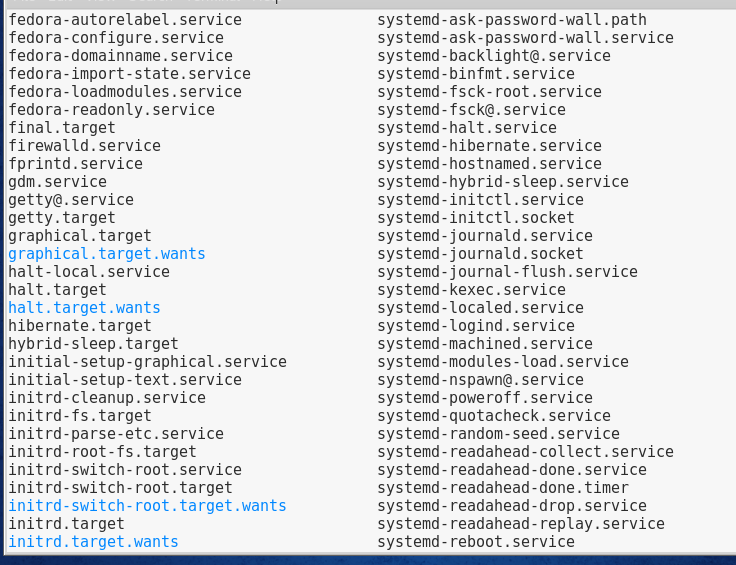
In this hands-on project, you explore and configure the system initialization process on Fedora 20.

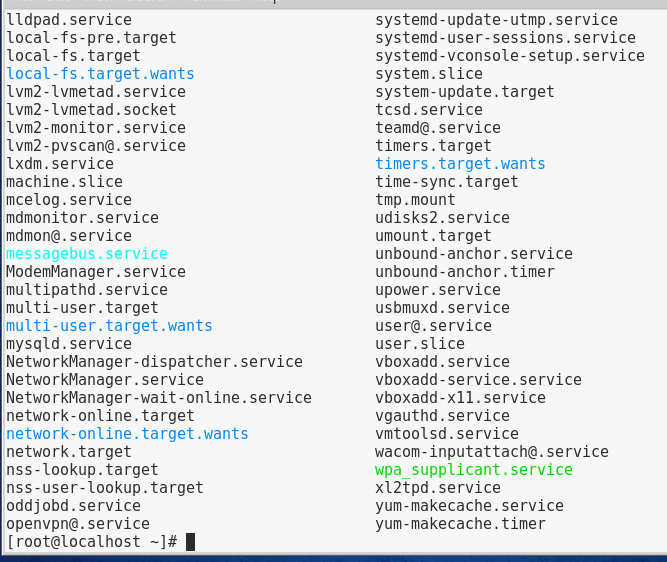
1. Boot your Fedora Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, type **runlevel** and press Enter. Your current runlevel should be 5 (graphical.target) and your previous runlevel should be 1 (rescue.target).
4. At the command prompt, type **ls /usr/lib/systemd/system** and press Enter. These are all the available units for Systemd.
5. At the command prompt, type **ls /etc/rc.d** and press Enter.
6. At the command prompt, type **ls /etc/rc.d/init.d** and press Enter. These are the available SysV init daemons.
7. At the command prompt, type **chkconfig --list netconsole** and press Enter. Note that the service is set to off for all available runlevels.
8. At the command prompt, type **chkconfig --level 23 netconsole on** and press Enter to configure the netconsole daemon to start in runlevels 2 and 3. Next, type **ls /etc/rc.d/rc[23].d** and press Enter. You will see that netconsole is configured to start.
9. At the command prompt, type **init 3** and press Enter to switch to runlevel 3 (multiuser.target). Note that you are on tty1 and the gdm is not loaded. You may need to press enter to have the terminal display. Log in to the terminal using the user name of **root** and the password of **LNXrocks!**.
10. **Provide screenshot(s) of steps 3 through 9.**

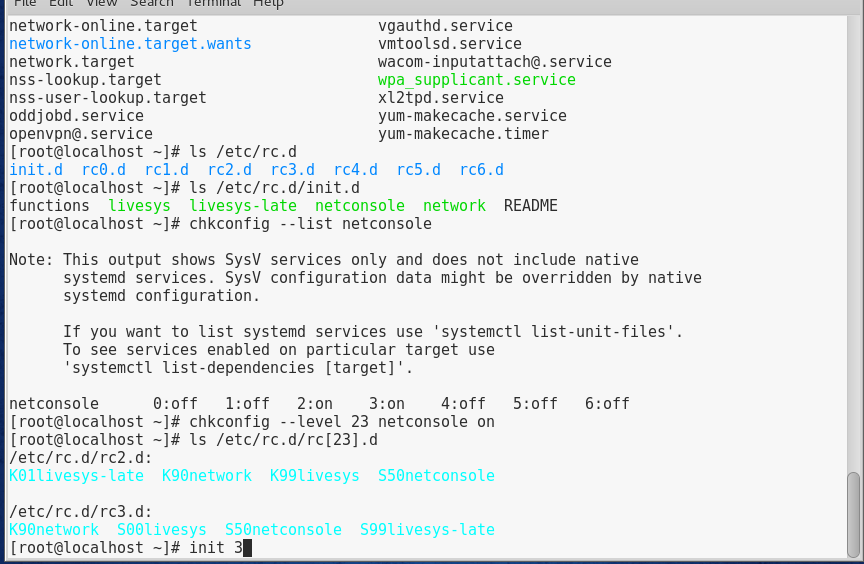
****

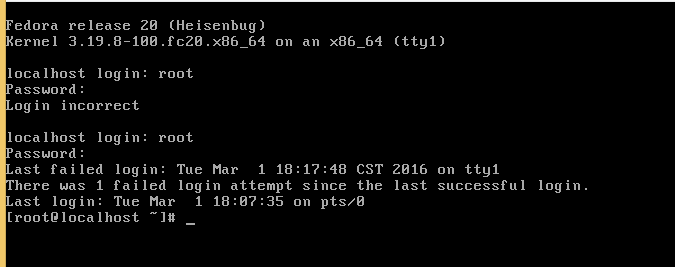
****

****

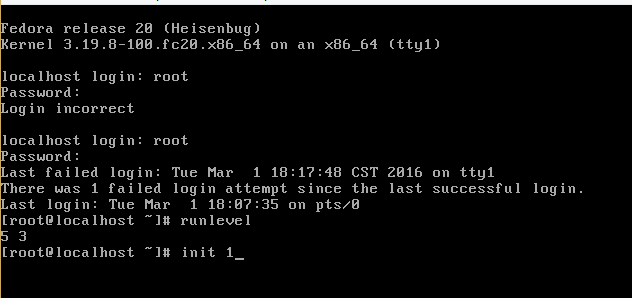
****

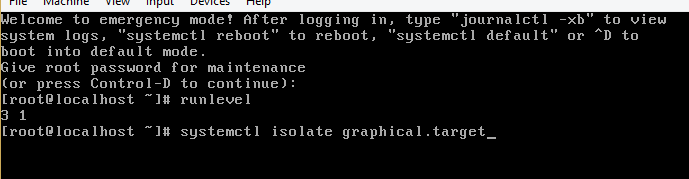
****

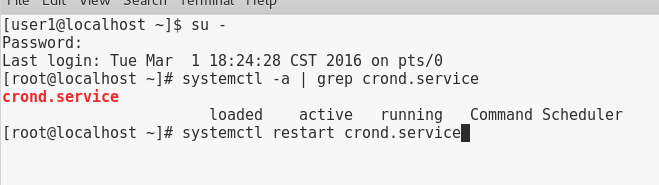
****

****

1. Next, type runlevel and press Enter. Review the output.
2. At the command prompt, type **init 1** and press Enter to switch to single user mode (rescue.target). Supply the **root** password of **LNXrocks!** when prompted.
3. Next, type **runlevel** and press Enter. Review the output.
4. At the command prompt, type **systemctl isolate graphical.target** and press Enter to switch to runlevel 5 (graphical.target). Note that the gdm is loaded. Log in to the desktop environment open a terminal and become **root** using the password of **LNXrocks!**.
5. At the command prompt, type **systemctl -a | grep crond.service** and press Enter. Is the Systemd cron daemon running?
6. At the command prompt, type **systemctl restart crond.service** and press Enter to restart the Systemd cron daemon.
7. At the command prompt, type **systemctl disable crond.service** and press Enter to prevent the system from starting the cron daemon in your current runlevel/target. Note that the existing symbolic link in the crond.service rc script is removed. Why was this link from the /etc/systemd/system/multi-user.target.wants directory instead of the /etc/systemd/system/graphical.target.wants directory?
8. At the command prompt, type **systemctl enable crond.service** and press Enter to start the cron daemon in your current runlevel/target. Was the symbolic link recreated?
9. At the command prompt, type **service netconsole start** and press Enter. Note that Systemd started the UNIX SysV netconsole daemon using the systemctl command because Systemd is backwards compatible with UNIX SysV.
10. **Provide screenshot(s) of steps 11 through 19.**

****

****

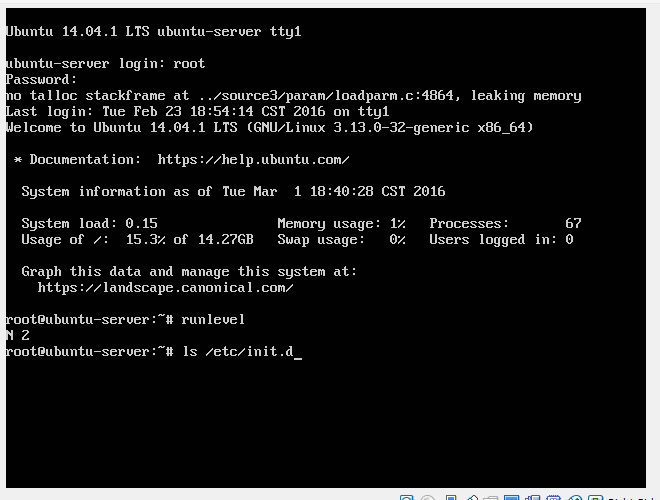
****

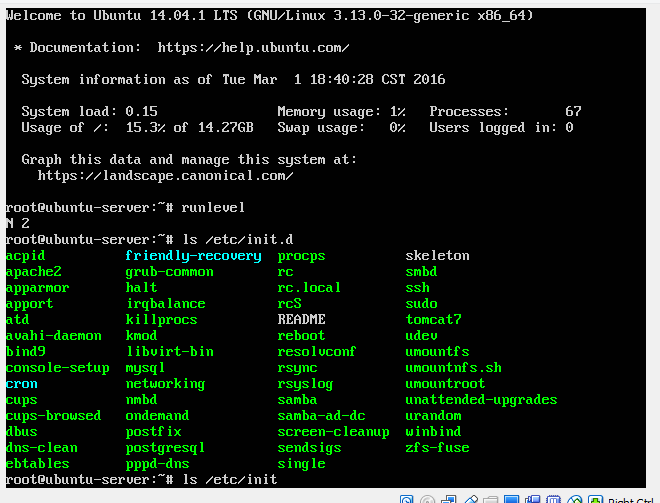
****

# Project 8-3

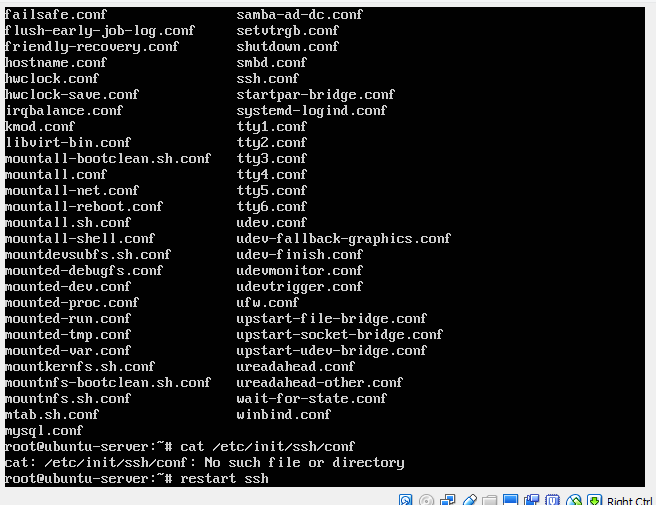
In this hands-on project, you explore and configure the system initialization process on Ubuntu Server 14.04..

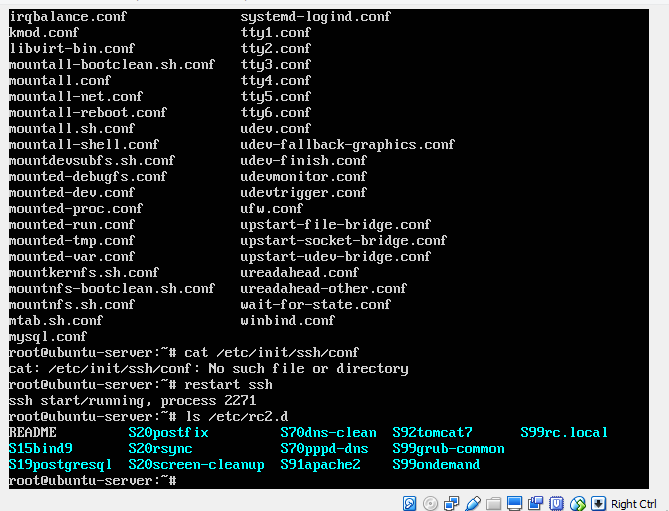
1. Login to your Ubuntu Server as **root** and provide **LNXrocks!** as the password.
2. At the command prompt, type **runlevel** and press Enter. Review your current and last runlevel.
3. At the command prompt, type **ls /etc/init.d** and press Enter. Review the contents.
4. At the command prompt, type **ls /etc/init** and press Enter. Review the contents.
5. At the command prompt, type **cat /etc/init/ssh.conf** and press Enter. In which runlevels is the ssh daemon started? At the command prompt, type **restart ssh** and press Enter.
6. At the command prompt, type **ls /etc/rc2.d** and press Enter. Are there any traditional UNIX SysV daemons started in your current runlevel? Is the postgresql daemon started before or after the apache2 daemon?
7. At the command prompt, type **restart postgresql** and press Enter. Why did you receive an error message?
8. At the command prompt, type **/etc/init.d/postgresql restart** and press Enter. Did the postgresql daemon restart?
9. At the command prompt, type **update-rc.d -f postgresql** remove and press Enter to remove the symbolic links that start the postgresql daemon. Which runlevels was the postgresql daemon originally started in?
10. At the command prompt, type **update-rc.d postgresql** defaults and press Enter to configure the symbolic links to start the postgresql daemon in runlevels 2 through 5.
11. **Provide screenshot(s) of steps 2 through 10.**

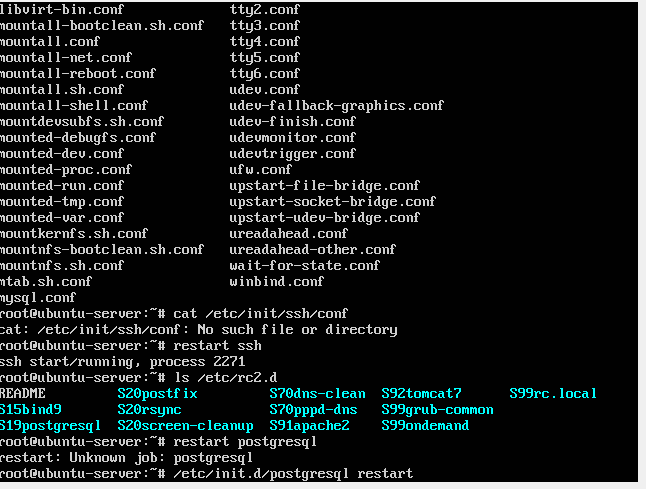
****

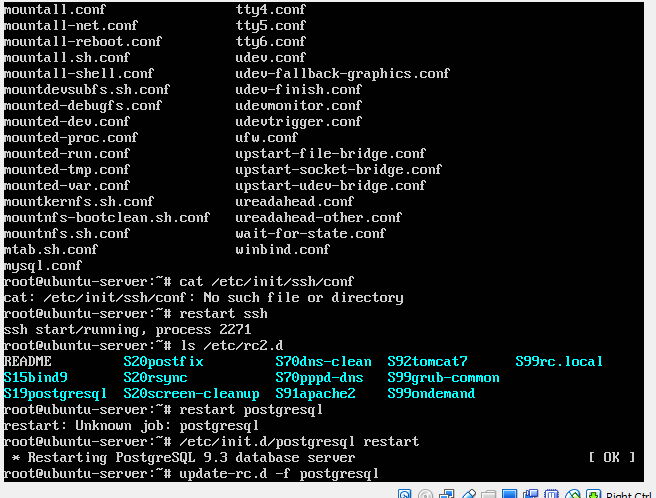
****

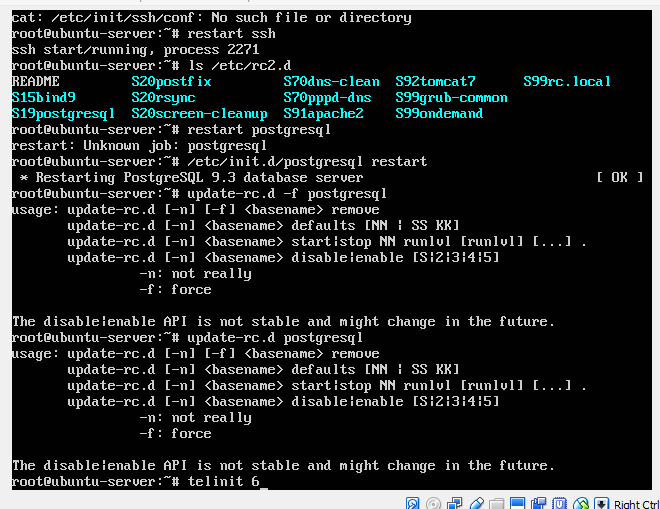
****

****

****

****

****

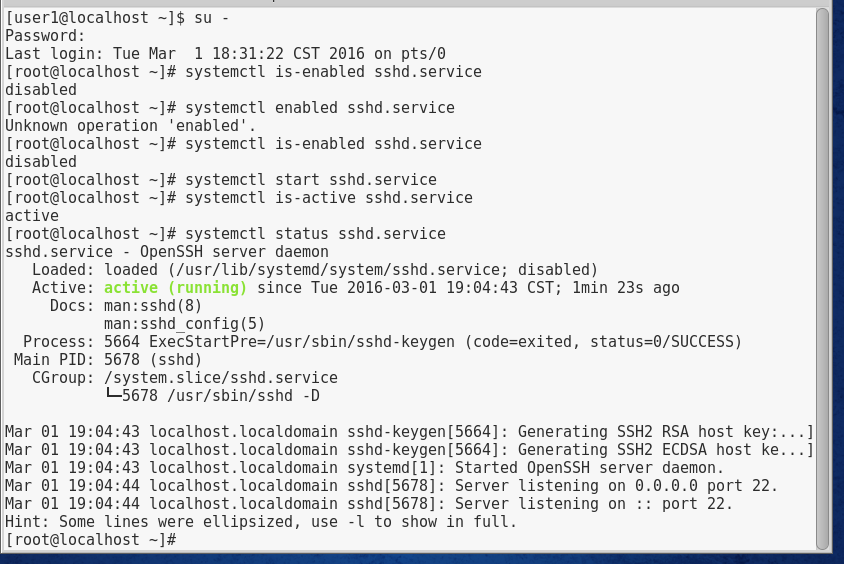
****

1. At the command prompt, type **telinit 6** and press Enter to reboot your system.
2. Once your system has rebooted, log into tty1 using the user name of **root** and the password of **LNXrocks!**.

# Project 8-4

In this hands-on project, you configure the ssh service on Fedora 20.

1. Boot your Fedora Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, type **systemctl is-enabled sshd.service** and press Enter. You should see that the ssh daemon is disabled.
4. At the command prompt, type **systemctl enabled sshd.service** and press Enter. You will see the symlink created to have ssh start upon system boot.
5. At the command prompt, type **systemctl is-enabled sshd.service** and press Enter. You should see that the ssh daemon is enabled.
6. At the command prompt, type **systemctl is-active sshd.service** and press Enter. You should see that the ssh daemon is inactive.
7. At the command prompt, type **systemctl start sshd.service** and press Enter. At the command prompt, type **systemctl is-active sshd.service** and press Enter. The service is now active since we started it.
8. At the command prompt, type **systemctl status sshd.service** and press Enter. Review the output.
9. **Provide screenshot(s) of steps 3 through 8.**

****