SECTION 7 Use the Command Line Interface to Administer the System

In this section of the workbook, you learn how to do the following:

Exercise 7-1 Manage User Accounts

In this exercise command line tools are employed to manager user accounts. Especially if there are many accounts to manage, the command line tools usually get the job done faster than YaST.

However, usually you have to use more than one tool, whereas in YaST everything is within one or two dialogs.

To manage user accounts, do the following:

1. Open a terminal window; then su to root (**su -**) with a password of **novell**.
2. Create a new local user by entering

   useradd -c "Tux Linux" -m tux
3. Verify that a home directory for tux was created by entering

   ls /home

SECTION 7 Use the Command Line Interface to Administer the System

In this section of the workbook, you learn how to do the following:

Exercise 7-1 Manage User Accounts

In this exercise command line tools are employed to manager user accounts. Especially if there are many accounts to manage, the command line tools usually get the job done faster than YaST.

However, usually you have to use more than one tool, whereas in YaST everything is within one or two dialogs.

To manage user accounts, do the following:

1. Open a terminal window; then su to root (**su -**) with a password of **novell**.
2. Create a new local user by entering

   useradd -c "Tux Linux" -m tux
3. Verify that a home directory for tux was created by entering

   ls /home

4. Verify that there is a entry for the tux user in /etc/shadow by entering

   cat /etc/shadow

   Notice the "!" in the second field, indicating that there is no password for tux. You did not use the option –p when creating tux, so there is no password set.

5. Add a password for the user tux by entering

   passwd tux

6. Enter the password **suse** twice.

7. Log out as root by entering

   exit

8. Log in as tux by entering

   su - tux

9. Enter the tux password (**suse**).

10. Change the password of the user tux by entering

    passwd

11. Enter the old password of the user tux (**suse**).

12. Try to change the password to novell by entering

    novell

    You receive a warning that the password is too simple.

13. Enter **d1g1t@l** as new password (twice).

14. Log out as user tux by entering

    exit

15. Switch to user root (**su -**) with a password of **novell.**

16. Delete the user tux by entering

    userdel -r tux

4. Verify that there is a entry for the tux user in /etc/shadow by entering

   cat /etc/shadow

   Notice the "!" in the second field, indicating that there is no password for tux. You did not use the option –p when creating tux, so there is no password set.

5. Add a password for the user tux by entering

   passwd tux

6. Enter the password **suse** twice.

7. Log out as root by entering

   exit

8. Log in as tux by entering

   su - tux

9. Enter the tux password (**suse**).

10. Change the password of the user tux by entering

    passwd

11. Enter the old password of the user tux (**suse**).

12. Try to change the password to novell by entering

    novell

    You receive a warning that the password is too simple.

13. Enter **d1g1t@l** as new password (twice).

14. Log out as user tux by entering

    exit

15. Switch to user root (**su -**) with a password of **novell.**

16. Delete the user tux by entering

    userdel -r tux

17. Verify that the home directory for tux has been removed by entering

    ls /home

18. Verify that there is no entry for tux in /etc/passwd by entering

    cat /etc/passwd

19. Close the terminal window.

(End of Exercise)

Exercise 7-2 Use ulimit

The program ulimit is useful when there are several users on a machine and you want to prevent them from giving each other a hard time by using too many of the available resources.

To use ulimit, do the following:

1. Perform the following steps:

```
tux@da10:~> echo "main() {for(;;)fork();}" > fork.c
tux@da10:~> gcc fork.c
```

   The program (a.out) merely serves demonstration purposes.

   This kind of program is referred to as fork bomb.

   The program continuously starts new instances of itself, making the computer virtually unusable due the multitude of processes - unless suitable precautions are taken before the program is started.

Do not execute this program on productive systems!

2. Set ulimit to 10.

17. Verify that the home directory for tux has been removed by entering

    ls /home

18. Verify that there is no entry for tux in /etc/passwd by entering

    cat /etc/passwd

19. Close the terminal window.

(End of Exercise)

Exercise 7-2 Use ulimit

The program ulimit is useful when there are several users on a machine and you want to prevent them from giving each other a hard time by using too many of the available resources.

To use ulimit, do the following:

1. Perform the following steps:

```
tux@da10:~> echo "main() {for(;;)fork();}" > fork.c
tux@da10:~> gcc fork.c
```

   The program (a.out) merely serves demonstration purposes.

   This kind of program is referred to as fork bomb.

   The program continuously starts new instances of itself, making the computer virtually unusable due the multitude of processes - unless suitable precautions are taken before the program is started.

Do not execute this program on productive systems!

2. Set ulimit to 10.

3. Start a.out.
4. Switch to another console and look at the process table by entering

   ps aux
5. Terminate a.out by pressing **Ctrl + c**.
6. Change the ulimit value.
7. Execute a.out again.
8. Observe the change in the processes by entering **ps aux** again.

   If the default ulimit value of 1023 is used, the computer will be virtually unusable following the execution of a.out.

   Often, the only thing you can do in such a case is to reboot the system.

(End of Exercise)

Exercise 7-3 Manage File Permissions and Ownership

File permissions and ownership is a subject any user on a Linux system needs to understand. For a system administrator this understanding is of crucial importance, as faulty permissions can have serious impact on the system security.

To manage file permissions and ownership, do the following:

1. As the user geeko open a terminal window (do not su to root).
2. Create two files:

   echo hello > perm_test1

   echo hello > perm_test2
3. Allow only the user geeko to read and write the file perm_test1 by entering

3. Start a.out.
4. Switch to another console and look at the process table by entering

   ps aux
5. Terminate a.out by pressing **Ctrl + c**.
6. Change the ulimit value.
7. Execute a.out again.
8. Observe the change in the processes by entering **ps aux** again.

   If the default ulimit value of 1023 is used, the computer will be virtually unusable following the execution of a.out.

   Often, the only thing you can do in such a case is to reboot the system.

(End of Exercise)

Exercise 7-3 Manage File Permissions and Ownership

File permissions and ownership is a subject any user on a Linux system needs to understand. For a system administrator this understanding is of crucial importance, as faulty permissions can have serious impact on the system security.

To manage file permissions and ownership, do the following:

1. As the user geeko open a terminal window (do not su to root).
2. Create two files:

   echo hello > perm_test1

   echo hello > perm_test2
3. Allow only the user geeko to read and write the file perm_test1 by entering

chmod 600 perm_test1

4. Verify that the change was made by entering

   ls -l

   Notice that geeko is the owner of the file, and that the only permissions assigned to the file are “rw” for the file owner.

   Also notice that others can read (r) the contents of the file perm_test2.

5. Remove the read permission for others of the file perm_test2 by entering

   chmod o-r perm_test2

6. Make sure that the permissions are correct by entering

   ls -l perm_test*

7. Su to root (**su -**) with a password of **novell**.

8. Create a file df by entering

   touch df

9. Verify that the file was created by entering

   ls -l df

   Notice that the owner of the file is root, and that the file group is also root.

10. Change the owner of the file df to nobody by entering

    chown nobody df

11. Change the group of the file df to nogroup by entering

    chgrp nogroup df

12. Make sure that the settings are correct by entering

    ls -l df

13. Log out as user root by entering

    exit

chmod 600 perm_test1

4. Verify that the change was made by entering

   ls -l

   Notice that geeko is the owner of the file, and that the only permissions assigned to the file are “rw” for the file owner.

   Also notice that others can read (r) the contents of the file perm_test2.

5. Remove the read permission for others of the file perm_test2 by entering

   chmod o-r perm_test2

6. Make sure that the permissions are correct by entering

   ls -l perm_test*

7. Su to root (**su -**) with a password of **novell**.

8. Create a file df by entering

   touch df

9. Verify that the file was created by entering

   ls -l df

   Notice that the owner of the file is root, and that the file group is also root.

10. Change the owner of the file df to nobody by entering

    chown nobody df

11. Change the group of the file df to nogroup by entering

    chgrp nogroup df

12. Make sure that the settings are correct by entering

    ls -l df

13. Log out as user root by entering

    exit

14. Close the terminal window.

(End of Exercise)

Exercise 7-4 Set Up and Configure Disk Quotas

Usually sooner or later the available storage space gets used up. While not a cure all, quotas are a means to prevent single users from using up the space, leaving too little for others to work effectively.

The purpose of this exercise is to practice setting and managing disk quotas.

To set up disk quotas, do the following:

1. From a terminal window, su to root (**su -**) with a password of **novell**.
2. Install the package qouta by entering **yast -i qouta** and inserting the SLES9 CDROMs when prompted.
3. View the disk quota configuration for user geeko by entering

   quota -vu geeko

   Notice that there are no quotas currently configured for geeko.

Use **df** to find out what partitions exist.

4. Add quota mount options to the partition /dev/hda2 (or the partition holding the directory / on your machine) by doing the following:
   a. Open the /etc/fstab file in the vi editor by entering

      vim /etc/fstab

   b. Edit the /dev/hda2 entry to reflect the following:

      /dev/hda2 / reiserfs defaults**,usrquota,grpquota** 1 2

   c. When you finish, save the file and exit by entering **:wq**.
5. Remount the file system so it that reads the changes in file /etc/fstab by entering

14. Close the terminal window.

(End of Exercise)

Exercise 7-4 Set Up and Configure Disk Quotas

Usually sooner or later the available storage space gets used up. While not a cure all, quotas are a means to prevent single users from using up the space, leaving too little for others to work effectively.

The purpose of this exercise is to practice setting and managing disk quotas.

To set up disk quotas, do the following:

1. From a terminal window, su to root (**su -**) with a password of **novell**.
2. Install the package qouta by entering **yast -i qouta** and inserting the SLES9 CDROMs when prompted.
3. View the disk quota configuration for user geeko by entering

   quota -vu geeko

   Notice that there are no quotas currently configured for geeko.

Use **df** to find out what partitions exist.

4. Add quota mount options to the partition /dev/hda2 (or the partition holding the directory / on your machine) by doing the following:
   a. Open the /etc/fstab file in the vi editor by entering

      vim /etc/fstab

   b. Edit the /dev/hda2 entry to reflect the following:

      /dev/hda2 / reiserfs defaults**,usrquota,grpquota** 1 2

   c. When you finish, save the file and exit by entering **:wq**.
5. Remount the file system so it that reads the changes in file /etc/fstab by entering

mount -o remount /

If you receive an error message "/ not mounted already, or bad option" check the contents of the /etc/fstab file. You might have misspelled the usrquota or grpquota option.

6. Run quotacheck to initialize the quota database by entering

   quotacheck -mavug

   You receive several status messages about old quota files. These indicate that this is a new quota database with no previous quota database files on the system.

7. Verify that the files aquota.user and aquota.groups exist in the directory / by entering

   ls -l /

8. Turn quotas on for all file systems that are mounted with these options by entering

   quotaon -av

9. Make the quota system persistent after reboot by entering

   chkconfig quota on

10. View the quota report by entering

    repquota -av

    The quotas are set by using the number of 1k blocks.

    Notice that root is the only user listed.

11. Set a quota for geeko of a soft limit of 200 MB and a hard limit of 300 MB on /dev/hda2 by entering

    edquota -u geeko

    The quota editor appears (the vi editor).

12. Enter the required ***soft limit*** and ***hard limit*** under the Soft and Hard columns for /dev/hda2 (press **Insert** twice to replace text).

mount -o remount /

If you receive an error message "/ not mounted already, or bad option" check the contents of the /etc/fstab file. You might have misspelled the usrquota or grpquota option.

6. Run quotacheck to initialize the quota database by entering

   quotacheck -mavug

   You receive several status messages about old quota files. These indicate that this is a new quota database with no previous quota database files on the system.

7. Verify that the files aquota.user and aquota.groups exist in the directory / by entering

   ls -l /

8. Turn quotas on for all file systems that are mounted with these options by entering

   quotaon -av

9. Make the quota system persistent after reboot by entering

   chkconfig quota on

10. View the quota report by entering

    repquota -av

    The quotas are set by using the number of 1k blocks.

    Notice that root is the only user listed.

11. Set a quota for geeko of a soft limit of 200 MB and a hard limit of 300 MB on /dev/hda2 by entering

    edquota -u geeko

    The quota editor appears (the vi editor).

12. Enter the required ***soft limit*** and ***hard limit*** under the Soft and Hard columns for /dev/hda2 (press **Insert** twice to replace text).

13. When you finish, press **Esc**; then enter **:wq**.
14. Run repquota to view the quota information about all configured users by entering

    repquota -av

    Notice that geeko is now listed with the soft limit and hard limit values you entered.
15. (Optional) If you finish early then set a quota for the users group of 500 MB for the soft limit, and 750 MB for the hard limit.
16. Close all open windows.

(End of Exercise)

Exercise 7-5 Execute RPM Package-Related Operations

The standard tool to install and remove software in SLES9 is YaST. YaST is a front end to the RPM package manager that does is responsible for the actual installation of the software.

The purpose of this exercise is to show you how you can use RPM directly to install and remove software.

To execute RPM package-related operations, do the following:

1. From a terminal window, su to root (**su -**) with a password of **novell**.
2. Insert *SLES 9 CD 3* in your CD-ROM drive.
3. List all files included in the not yet installed package gcal by entering

   rpm -qpl /media/cdrom/suse/i586/gcal-3.01-581.1.i586.rpm
4. Install the package gcal by entering

   rpm -ihv /media/cdrom/suse/i586/gcal-3.01-581.1.i586.rpm

13. When you finish, press **Esc**; then enter **:wq**.
14. Run repquota to view the quota information about all configured users by entering

    repquota -av

    Notice that geeko is now listed with the soft limit and hard limit values you entered.
15. (Optional) If you finish early then set a quota for the users group of 500 MB for the soft limit, and 750 MB for the hard limit.
16. Close all open windows.

(End of Exercise)

Exercise 7-5 Execute RPM Package-Related Operations

The standard tool to install and remove software in SLES9 is YaST. YaST is a front end to the RPM package manager that does is responsible for the actual installation of the software.

The purpose of this exercise is to show you how you can use RPM directly to install and remove software.

To execute RPM package-related operations, do the following:

1. From a terminal window, su to root (**su -**) with a password of **novell**.
2. Insert *SLES 9 CD 3* in your CD-ROM drive.
3. List all files included in the not yet installed package gcal by entering

   rpm -qpl /media/cdrom/suse/i586/gcal-3.01-581.1.i586.rpm
4. Install the package gcal by entering

   rpm -ihv /media/cdrom/suse/i586/gcal-3.01-581.1.i586.rpm

5. Remove the CD from your drive.
6. (Conditional) If the CD drive does not open, enter

   umount /media/cdrom

   Then try again.
7. Test the installation of the software package by entering

   gcal
8. List all files included in the installed package gcal by entering

   rpm -ql gcal
9. Remove the package gcal by entering

   rpm -e gcal
10. Verify that the package is no longer installed by entering

    rpm -ql gcal
11. Log out as root by entering

    exit
12. Close the terminal window.

(End of Exercise)

5. Remove the CD from your drive.
6. (Conditional) If the CD drive does not open, enter

   umount /media/cdrom

   Then try again.
7. Test the installation of the software package by entering

   gcal
8. List all files included in the installed package gcal by entering

   rpm -ql gcal
9. Remove the package gcal by entering

   rpm -e gcal
10. Verify that the package is no longer installed by entering

    rpm -ql gcal
11. Log out as root by entering

    exit
12. Close the terminal window.

(End of Exercise)