25D Linux Foundation Course

04 – Managing Linux Files and Directories

Overview

- Understanding Linux file systems and the Filesystem Hierarchy Standard (FHS)
- Finding files in the Linux file system
- Managing directories from the command line
- Managing files from the command line
- Working with link files
- Finding content within files

Understanding Linux File Systems and the Filesystem Hierarchy Standard (FHS)

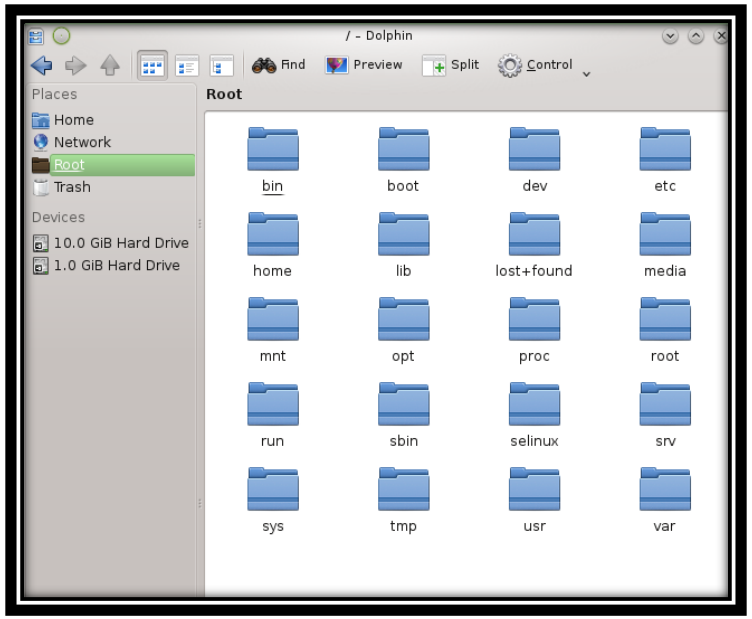
- The role of the Linux file system
- The hierarchical structure of the Linux file system
- Linux file types

The Role of the Linux File System

- The file system stores information on a storage device in such a manner that:
  - Data can be saved in a persistent manner
  - Data is organized and can be easily located
  - Data can be quickly retrieved for use at a later point in time
  - Data integrity is preserved

The Hierarchical Structure of the Linux File System

- Topmost directory is the / directory or root
- Below it are the subdirectories
- Specifications for the subdirectories are located in the Filesystem Hierarchy Standard



The Hierarchical Structure of the Linux File System

❑ Important directories in the Linux root directory:

Directory	Description
`/bin`	This directory contains the executable files necessary to manage and run the Linux system, including shells (such as bash) and file system management utilities such as cp and rm.
`/boot`	This directory contains your bootloader files, which are required to boot your system.
`/dev`	This directory contains special files that are used to represent the various hardware devices installed in the system.

- ❑ Character-oriented device files
  - These files are used for devices that send or receive data sequentially one character at a time, such as a printer, mouse, or tape drive.

- ❑ Block-oriented device files
  - These files are used for devices that manage data in blocks, such as floppy disks and hard drives.

The Hierarchical Structure of the Linux File System

- Hardware represented by files in the /dev directory:

Device	Device File in /dev
Floppy drive	/dev/fd0
Optical drive	/dev/scd0 or /dev/sr0
Serial port	/dev/ttyS0
Parallel port	/dev/lp0

- Physical hardware is addressed by applications and services in /dev:
  - Saving a file to disk? It goes through a file in /dev.
  - Sending a print job to a printer? It goes through a file in /dev.

The Hierarchical Structure of the Linux File System

❑ Important directories in the root directory (cont.):

Directory	Description
/etc	This directory contains text-based configuration files used by the system, as well as services running on the system. You can edit these files with a text editor to customize how Linux behaves.
/home	This directory contains subdirectories that serve as home directories for each user account on your Linux system.
/lib	This directory contains code libraries used by programs in /bin and /sbin. Your kernel modules are also stored in the modules subdirectory of /lib.
/media	This directory is used by some Linux distributions (such as openSUSE and Fedora) to mount external devices, including optical drives and USB drives.
/mnt	This directory is used by some Linux distributions to mount external devices, including optical drives and \USB drives.
/opt	This directory contains files for some programs you install on the system.
/proc	a pseudo–file system that is dynamically created whenever it is accessed. It's used to access process and other system information from the Linux kernel.

The Hierarchical Structure of the Linux File System

❑ Important files in the /etc directory:

File	Function
/etc/aliases	Contains a table used to redirect mail to local recipients.
/etc/exports	Configures file systems to be exported to remote NFS clients.
/etc/fstab	Lists the partitions and file systems that will be automatically mounted when the system boots.
/etc/ftpusers	Controls user access to the FTP service running on the system.
/etc/group	Contains local group definitions.
/etc/hosts	Contains a list of hostname-to–IP address mappings the system can use to resolve hostnames.
/etc/inittab	Contains configuration parameters for the init process.

❑ Continued on next slide…

The Hierarchical Structure of the Linux File System

❑ Important files in the /etc directory (cont.):

File	Function
/etc/init.d/	A subdirectory that contains startup scripts for services installed on the system. On a Fedora or Red Hat system, these are located in /etc/rc.d/init.d.
/etc/nsswitch.conf	Configures which services are to be used to resolve hostnames and to store users, groups, and passwords.
/etc/passwd	Contains your system user accounts.
/etc/resolv.conf	Specifies the DNS server and domain suffix used by the system.
/etc/services	Maps port numbers to named services on the system.
/etc/shadow	Contains encrypted passwords for your user accounts.
/etc/X11/	Contains X Window configuration files.

The Hierarchical Structure of the Linux File System

- Contents of the /proc subdirectory:

```
openSUSE:/ # cd /proc
openSUSE:/proc # ls
1     182   3162  487  91           dri           loadavg       stat
10    2     3163  488  922          driver        locks         swaps
102   20    3165  489  93           execdomains   mdstat        sys
103   21    3250  490  987          fb            meminfo       sysrq-trigger
11    22    3251  492  acpi         filesystems   misc          sysvipc
111   220   3275  494  buddyinfo    fs            modules       timer_list
114   221   3277  5    bus          interrupts    mounts        timer_stats
12    23    3284  537  cgroups      iomem         mtrr          tty
1214  24    33    567  cmdline      ioports       net           uptime
13    25    3354  571  config.gz    irq           pagetypeinfo  version
14    253   356   597  consoles     kallsyms      partitions    vmallocinfo
15    285   358   6    cpuinfo      kcore         sched_debug   vmstat
16    3     359   68   crypto       key-users     schedstat     zoneinfo
17    30    360   7    device-tree  kmsg          scsi
18    3008  3859  8    devices      kpagecount    self
180   31    3860  9    diskstats    kpageflags    slabinfo
181   3152  3861  90   dma          latency_stats softirqs
```

- The numbers for the directories are associated with process ID (PID) of the software process running on the system.

The Hierarchical Structure of the Linux File System

- The top utility can be used to display running processes:

```
Tasks:  72 total,   1 running,  71 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   1027172 total,   592152 used,   435020 free,    91284 buffers
KiB Swap:  1051644 total,        0 used,  1051644 free,   428360 cached

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  987 kdm       20   0  129412  48296  23552 S 0.331 4.702   0:07.34 kdm_greet
 3883 root      20   0    3632   1192    892 R 0.331 0.116   0:00.14 top
    1 root      20   0    5960   3312   2252 S 0.000 0.322   0:00.68 systemd
    2 root      20   0       0      0      0 S 0.000 0.000   0:00.00 kthreadd
    3 root      20   0       0      0      0 S 0.000 0.000   0:00.07 ksoftirqd+
    5 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 kworker/0+
    6 root      20   0       0      0      0 S 0.000 0.000   0:00.00 kworker/u+
    7 root      rt   0       0      0      0 S 0.000 0.000   0:00.04 migration+
    8 root      20   0       0      0      0 S 0.000 0.000   0:00.00 rcu_bh
    9 root      20   0       0      0      0 S 0.000 0.000   0:00.44 rcu_sched
   10 root      rt   0       0      0      0 S 0.000 0.000   0:00.05 watchdog/0
   11 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 khelper
   12 root      20   0       0      0      0 S 0.000 0.000   0:00.00 kdevtmpfs
   13 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 netns
   14 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 writeback
   15 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 kintegrit+
   16 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 bioset
   17 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 kblockd
   18 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 md
   20 root      20   0       0      0      0 S 0.000 0.000   0:00.00 khungtaskd
   21 root      20   0       0      0      0 S 0.000 0.000   0:00.00 kswapd0
   22 root      25   5       0      0      0 S 0.000 0.000   0:00.00 ksmd
   23 root      39  19       0      0      0 S 0.000 0.000   0:00.04 khugepaged
openSUSE:/proc #
```

- In the above example the top utility has a PID of 3883

The Hierarchical Structure of the Linux File System

- The PID directories within the /proc directory can be accessed as well as files within the PID directory:

```
openSUSE:/proc # cd 21
openSUSE:/proc/21 # ls -a
ls: cannot read symbolic link exe: No such file or directory
.           coredump_filter  latency     net            sched      syscall
..          cpuset           limits      ns             schedstat  task
attr        cwd              loginuid    oom_adj        sessionid  wchan
auxv        environ          maps        oom_score      smaps
cgroup      exe              mem         oom_score_adj  stack
clear_refs  fd               mountinfo   pagemap        stat
cmdline     fdinfo           mounts      personality    statm
comm        io               mountstats  root           status
openSUSE:/proc/21 # cat limits
Limit                     Soft Limit           Hard Limit           Units
Max cpu time              unlimited            unlimited            seconds
Max file size             unlimited            unlimited            bytes
Max data size             unlimited            unlimited            bytes
Max stack size            8388608              unlimited            bytes
Max core file size        0                    unlimited            bytes
Max resident set          unlimited            unlimited            bytes
Max processes             7847                 7847                 processes
Max open files            1024                 4096                 files
Max locked memory         65536                65536                bytes
Max address space         unlimited            unlimited            bytes
Max file locks            unlimited            unlimited            locks
Max pending signals       7847                 7847                 signals
Max msgqueue size         819200               819200               bytes
Max nice priority         0                    0
Max realtime priority     0                    0
Max realtime timeout      unlimited            unlimited            us
openSUSE:/proc/21 #
```

The Hierarchical Structure of the Linux File System

❑ Important directories in the root directory (cont.):

Directory	Function
/root	This directory is the root user's home directory.
/sbin	This directory contains important system management and administration files, such as fdisk, fsck, ifconfig, init, mkfs, shutdown, and halt.
/srv	This directory contains subdirectories where services running on the system (such as httpd and ftpd) save their files.
/sys	This directory contains information about the hardware in your system.
/tmp	This directory contains temporary files created by you or by the system.
/usr	This directory contains application files.
/var	This directory contains variable data, including your system log files.

The Hierarchical Structure of the Linux File System

Subdirectories of /usr	Contents
bin	Most of your executable programs
lib	Library files
lib64	64-bit library files
local	Locally installed software that you created yourself (used to prevent it from being overwritten during a system update)
sbin	System administration programs
share	Documentation and man page files

Subdirectory of /var	Contents
lib	Library files created by various services and applications running on the system
log	Log files from your system and from services running on the system
spool	Print queues

Types of Files Used by Linux

- There are several file types used in the Linux file system:

File Type	Description
Regular files	These files are similar to those used by the file systems of other operating systems—for example, executable files, word processing files, images, text files, and so on.
Links	These files are pointers that point to other files in the file system.
FIFOs	FIFO stands for First In, First Out. These are special files used to move data from one running process on the system to another. A FIFO file is basically a queue where the first chunk of data added to the queue is the first chunk of data removed from the queue. Data can only move in one direction through a FIFO.
Sockets	Sockets are similar to FIFOs in that they are used to transfer information between processes. Unlike FIFOs, however, sockets can move data bi-directionally.

Finding Files in the Linux File System

- Using find
- Using locate
- Using which
- Using whereis
- Using type

Using find

- The find utility can be used to search the file system:

```
openSUSE:~ # find / -name "*.log"
/var/log/Xorg.0.log
/var/log/YaST2/config_diff_2015_02_04.log
/var/log/YaST2/y2start.log
/var/log/YaST2/internet-test/ip_addr.log
/var/log/YaST2/internet-test/ip_route.log
/var/log/YaST2/internet-test/curl_0.log
/var/log/YaST2/config_diff_2013_11_06.log
/var/log/YaST2/mkinitrd.log
/var/log/kdm.log
/var/log/boot.log
/var/log/wpa_supplicant.log
/var/log/pbl.log
/var/log/pm-powersave.log
/var/log/alternatives.log
/usr/share/doc/packages/libjpeg-turbo/change.log
/home/student/.kde4/share/apps/kconf_update/log/update.log
/home/student/.kde4/share/apps/nepomuk/repository/main/data/virtuosobackend/sopr
ano-virtuoso.log
```

- In this example the find utility will locate and return all file extensions with .log starting in the root directory and then all subdirectories.

Using find

- In the below example a user named clannister used the find command to identify files owned by user astark in the /home directory.

```
openSUSE:/home/clannister # find /home -user "astark"
/home/astark
/home/astark/bin
/home/astark/.inputrc
/home/astark/.emacs
/home/astark/aryastuff
/home/astark/.local
/home/astark/.local/share
/home/astark/.local/share/systemd
/home/astark/.local/share/systemd/user
/home/astark/.xim.template
/home/astark/.viminfo
/home/astark/aryalist
/home/astark/.config
/home/astark/public_html
/home/astark/public_html/.directory
/home/astark/.fonts
/home/astark/.profile
/home/astark/.bash_history
/home/astark/.bashrc
/home/astark/.xinitrc.template
```

Using locate

- Ensure findutils-locate package is installed
  - It is in our version of SUSE
- locatedb
  - Index of all files in file system created in /var/log
  - Indexed each day
- updatedb command
  - Updates the locatedb manually

```
openSUSE:/ # locate aryalist
openSUSE:/ # updatedb
openSUSE:/ # locate aryalist
/home/astark/aryalist
```

Using which

- Used to display full path to a shell command or utility

```
openSUSE:~ # which locate
/usr/bin/locate
openSUSE:~ # which find
/usr/bin/find
openSUSE:~ # which ls
/usr/bin/ls
openSUSE:~ # which cat
/usr/bin/cat
```

Using whereis

- The whereis command locates the source code, binary files, and manual pages for specified files.
  - -b option returns the binary location
  - -m option returns the manual page location
  - -s option returns the source code location

```
openSUSE:~ # whereis -s cat
cat:openSUSE:~ # whereis -b cat
cat: /usr/bin/cat /bin/cat
openSUSE:~ # whereis -m cat
cat: /usr/share/man/man1/cat.1.gz /usr/share/man/man1p/cat.1p.gz
openSUSE:~ # whereis -s cat
cat:openSUSE:~ #
```

- There are other options available with whereis to assist in searches.

Using type

- The type command returns what type of command is executed when you enter it.
  - A command that is hard-coded into the shell itself

```
openSUSE:~ # type cd
cd is a shell builtin
```

  - An external command that is called by the shell

```
openSUSE:~ # type cat
cat is /usr/bin/cat
```

  - An alias

```
openSUSE:~ # type ls
ls is aliased to `_ls'
```

  - A function
    - New internal commands created by defining your own shell functions

Exercise 4-1: Using Linux Search Tools

Please open your Practical Exercise book to Exercise 4-1.

Time to Complete: 5 Minutes

Managing Directories from the Command Line

- Navigating the file system
- Viewing directory contents
- Creating new directories
- Copying, moving, and deleting directories

Navigating the File System

❑ Using the pwd Utility

- Print Working Directory

```
student@openSUSE:~> pwd
/home/student
```

- Useful especially if shell is not configured to display current directory

Navigating the File System

Using the cd Utility

- Change directories in the file system (relative path):

```
student@openSUSE:~> cd Documents
student@openSUSE:~/Documents>
```

- Using absolute path:

```
student@openSUSE:~> cd /var/log
student@openSUSE:/var/log>
```

- Next directory higher in the hierarchy:

```
student@openSUSE:/var/log> cd ..
student@openSUSE:/var>
```

- Two directories higher in the hierarchy and back to home:

```
student@openSUSE:/var/log> cd ../..
student@openSUSE:/> pwd
/
student@openSUSE:/> cd
student@openSUSE:~> _
```

Viewing Directory Contents

- Using the ls Utility
- Viewing contents of current directory:

```
student@openSUSE:~> ls
bin       Documents  Music     Public       Templates  Videos
Desktop   Downloads  Pictures  public_html  test.txt
```

- Viewing contents of another directory using the absolute path

```
student@openSUSE:~> ls /var/log
acpid              hp              messages           wpa_supplicant.log
alternatives.log   kdm.log         NetworkManager     wtmp
apparmor           krb5            news               Xorg.0.log
audit              lastlog         ntp                Xorg.0.log.old
boot.log           localmessages   pbl.log            YaST2
btmp               mail            pk_backend_zypp    zypp
cups               mail.err        pm-powersave.log
faillog            mail.info       samba
firewall           mail.warn       warn
```

Viewing Directory Contents

- Using ls Utility with the –a option
  - Will list all files to include hidden files in that directory

```
student@openSUSE:~> ls -a
.                .dmrc        .local       .vboxclient-draganddrop.pid
..               Documents    Music        Videos
.bash_history    Downloads    Pictures     .viminfo
.bashrc          .emacs       .profile     .Xauthority
bin              .esd_auth    Public       .xim.template
.cache           .fonts       public_html  .xinitrc.template
.config          .gtkrc-2.0   .skel        .xsession-errors
.dbus            .inputrc     Templates    .xsession-errors-:0
Desktop          .kde4        test.txt
```

Viewing Directory Contents

- **Using ls with the –l option**
  - **Will print a long listing of files and directories to the screen**

```
student@openSUSE:~> ls -l
total 44
drwxr-xr-x 2 student users 4096 Feb  4  2015 bin
drwxr-xr-x 2 student users 4096 Feb  4  2015 Desktop
drwxr-xr-x 2 student users 4096 Feb  4  2015 Documents
drwxr-xr-x 2 student users 4096 Feb  4  2015 Downloads
drwxr-xr-x 2 student users 4096 Feb  4  2015 Music
drwxr-xr-x 2 student users 4096 Feb  4  2015 Pictures
drwxr-xr-x 2 student users 4096 Feb  4  2015 Public
drwxr-xr-x 2 student users 4096 Feb  4  2015 public_html
drwxr-xr-x 2 student users 4096 Feb  4  2015 Templates
-rw-r--r-- 1 student users  180 Feb  4  2015 test.txt
drwxr-xr-x 2 student users 4096 Feb  4  2015 Videos
```

Viewing Directory Contents

- **Using ls with the –R option**
  - **Displays directory contents recursively (current and subdirectories)**
  - **May need to pipe this command with the more utility depending on the amount of files in all of the directories**

```
student@openSUSE:~> ls -R
.:
bin      Documents  Music     Public       Templates  Videos
Desktop  Downloads  Pictures  public_html  test.txt

./bin:

./Desktop:
kinfocenter.desktop     Office.desktop   SuSE.desktop
MozillaFirefox.desktop  Support.desktop

./Documents:

./Downloads:

./Music:

./Pictures:

./Public:

./public_html:

./Templates:

./Videos:
```

Exercise 4-2: Navigating the File System

Please open your Practical Exercise book to Exercise 4-2.

Time to Complete: 5 Minutes

Creating New Directories

Using the mkdir Utility

- Used to create new directories using relative or absolute paths:
  - Relative:

```
student@openSUSE:~> mkdir MyFiles
student@openSUSE:~> ls
bin       Documents  Music    Pictures  public_html  test.txt
Desktop   Downloads  MyFiles  Public    Templates    Videos
```

  - Absolute:

```
openSUSE:~ # mkdir /var/log/SuperAuditing
openSUSE:~ # ls /var/log
NetworkManager    apparmor  hp             mail.info        pm-powersave.log
SuperAuditing     audit     kdm.log        mail.warn        samba
Xorg.0.log        boot.log  krb5           messages         warn
Xorg.0.log.old    btmp      lastlog        news             wpa_supplicant.log
YaST2             cups      localmessages  ntp              wtmp
acpid             faillog   mail           pbl.log          zypp
alternatives.log  firewall  mail.err       pk_backend_zypp
```

Creating New Directories

Using the md alias

- The md alias will run the mkdir utility with the –p option
  - The –p option means create every directory in the command if it does not already exist

```
student@openSUSE:~> md ~/temp/backups/daily
student@openSUSE:~> ls -R
.:
bin       Documents  Music    Pictures  public_html  Templates  Videos
Desktop   Downloads  MyFiles  Public    temp         test.txt

./temp:
backups

./temp/backups:
daily

./temp/backups/daily:
```

Copying, Moving, and Deleting Directories

Using the cp Utility

- Copies entire directory structures from one location in the file system to another
  - -R is required to copy the entire structure

```
student@openSUSE:~/MyFiles> ls
File1  File2  File3  Folder1  Folder2  Folder3
student@openSUSE:~/MyFiles> ls ~/backup
student@openSUSE:~/MyFiles> cp -R ~/MyFiles/ ~/backup
student@openSUSE:~/MyFiles> ls ~/backup
MyFiles
student@openSUSE:~/MyFiles> ls ~/backup/MyFiles
File1  File2  File3  Folder1  Folder2  Folder3
```

- In this example the MyFiles directory along with directories and files within were copied to the backup directory residing in the home directory

Copying, Moving, and Deleting Directories

Using the mv Utility

- Works much like the cp utility
- copies specified directory from one location in the file system to another but deletes the original file

```
student@openSUSE:~> mv ~/backup /tmp
student@openSUSE:~> ls
bin       Documents  Music     Pictures  public_html  Templates  Videos
Desktop   Downloads  MyFiles   Public    temp         test.txt
student@openSUSE:~> ls /tmp
1682198942  kde-kdm      ksocket-kdm      virtuoso_w23608.ini
backup      kde-student  ksocket-student
student@openSUSE:~> ls /tmp/backup
MyFiles
student@openSUSE:~> ls /tmp/backup/MyFiles
File1  File2  File3  Folder1  Folder2  Folder3
```

- In this example the mv utility was used to move the backup directory and it's contents from the home directory to the /tmp directory. The ls commands were used to verify the move.

Copying, Moving, and Deleting Directories

Using the rmdir Utility

- deletes an existing directory
  - works only if the directory is empty

```
student@openSUSE:~> ls
bin       Documents  Music     Pictures  public_html  temp       test.txt
Desktop   Downloads  MyFiles   Public    StudentDocs  Templates  Videos
student@openSUSE:~> ls StudentDocs
student@openSUSE:~> rmdir StudentDocs
student@openSUSE:~> ls
bin       Documents  Music     Pictures  public_html  Templates  Videos
Desktop   Downloads  MyFiles   Public    temp         test.txt
student@openSUSE:~> ls temp
backups
student@openSUSE:~> rmdir temp
rmdir: failed to remove ■temp■: Directory not empty
student@openSUSE:~> _
```

- In this example the empty directory StudentDocs was deleted using the rmdir utility but the temp directory cannot be removed using the mkdir utility.

Copying, Moving, and Deleting Directories

Using the rm Utility

- can be used to delete a populated directory
  - -r option used to remove directories recursively

```
student@openSUSE:~> ls /tmp/
1682198942  kde-kdm      ksocket-kdm      virtuoso_wZ3608.ini
backup      kde-student  ksocket-student
student@openSUSE:~> ls /tmp/backup/
MyFiles
student@openSUSE:~> ls /tmp/backup/MyFiles
File1  File2  File3  Folder1  Folder2  Folder3
student@openSUSE:~> rm -r /tmp/backup
student@openSUSE:~> ls /tmp/
1682198942  kde-student  ksocket-student
kde-kdm     ksocket-kdm  virtuoso_wZ3608.ini
```

- In this example the backup directory residing in the tmp directory was removed including all of the subdirectories and files located within it (MyFiles and its contents) using the rm utility.

Managing Files from the Command Line

- Creating files
- Viewing file contents
- Deleting files
- Copying and moving files
- Determining the file type

Creating New Files

Using the touch Utility

- can be used to create new files

```
student@openSUSE:~> ls
bin       Documents  Music    Pictures  public_html  Templates  Videos
Desktop   Downloads  MyFiles  Public    temp         test.txt
student@openSUSE:~> touch superawesomefile
student@openSUSE:~> ls -l
total 52
drwxr-xr-x 2 student users 4096 Feb  4  2015 bin
drwxr-xr-x 2 student users 4096 Feb  4  2015 Desktop
drwxr-xr-x 2 student users 4096 Feb  4  2015 Documents
drwxr-xr-x 2 student users 4096 Feb  4  2015 Downloads
drwxr-xr-x 2 student users 4096 Feb  4  2015 Music
drwxr-xr-x 5 student users 4096 Nov 29 08:59 MyFiles
drwxr-xr-x 2 student users 4096 Feb  4  2015 Pictures
drwxr-xr-x 2 student users 4096 Feb  4  2015 Public
drwxr-xr-x 2 student users 4096 Feb  4  2015 public_html
-rw-r--r-- 1 student users    0 Nov 29 10:46 superawesomefile
drwxr-xr-x 3 student users 4096 Nov 29 08:49 temp
drwxr-xr-x 2 student users 4096 Feb  4  2015 Templates
-rw-r--r-- 1 student users  180 Feb  4  2015 test.txt
drwxr-xr-x 2 student users 4096 Feb  4  2015 Videos
```

- In this example a file named superawesomefile was created in the /home/student directory using touch

Viewing File Contents

- **cat**
  - Displays specified text file on screen
- **less**
  - Displays specified text file one page at a time
- **head**
  - Displays the first couple of lines of a text file on the screen
- **tail**
  - Displays the last couple of lines of a text file on the screen

Viewing File Contents

- cat /var/log/messages example:

```
r root.
2016-11-29T10:00:01.204599-07:00 openSUSE systemd[1]: Started Session 20 of user
 root.
2016-11-29T10:00:01.231441-07:00 openSUSE /USR/SBIN/CRON[4099]: pam_unix(crond:s
ession): session closed for user root
2016-11-29T10:15:01.244792-07:00 openSUSE /usr/sbin/cron[4148]: pam_unix(crond:s
ession): session opened for user root by (uid=0)
2016-11-29T10:15:01.253981-07:00 openSUSE systemd[1]: Starting Session 21 of use
r root.
2016-11-29T10:15:01.256430-07:00 openSUSE systemd[1]: Started Session 21 of user
 root.
2016-11-29T10:15:01.284030-07:00 openSUSE /USR/SBIN/CRON[4148]: pam_unix(crond:s
ession): session closed for user root
2016-11-29T10:30:01.293541-07:00 openSUSE /usr/sbin/cron[4170]: pam_unix(crond:s
ession): session opened for user root by (uid=0)
2016-11-29T10:30:01.295588-07:00 openSUSE systemd[1]: Starting Session 22 of use
r root.
2016-11-29T10:30:01.295838-07:00 openSUSE systemd[1]: Started Session 22 of user
 root.
2016-11-29T10:30:01.309096-07:00 openSUSE /USR/SBIN/CRON[4170]: pam_unix(crond:s
ession): session closed for user root
2016-11-29T10:45:01.323141-07:00 openSUSE /usr/sbin/cron[4192]: pam_unix(crond:s
ession): session opened for user root by (uid=0)
2016-11-29T10:45:01.332810-07:00 openSUSE systemd[1]: Starting Session 23 of use
r root.
2016-11-29T10:45:01.337010-07:00 openSUSE systemd[1]: Started Session 23 of user
 root.
2016-11-29T10:45:01.367925-07:00 openSUSE /USR/SBIN/CRON[4192]: pam_unix(crond:s
ession): session closed for user root
2016-11-29T10:45:45.503216-07:00 openSUSE kernel: [16357.788751] e1000: enp0s3 N
```

- One of many pages displayed on the screen
  - Scrolls without stopping to the end of the file

Viewing File Contents

- less cat /var/log/messages example:

```
2015-02-04T10:53:58.918399-07:00 linux rsyslogd: [origin software="rsyslogd" swV
ersion="7.4.4" x-pid="517" x-info="http://www.rsyslog.com"] start
2015-02-04T10:53:58.923512-07:00 linux systemd[1]: Starting Setup Virtual Consol
e...
2015-02-04T10:53:58.950752-07:00 linux systemd[1]: Starting Arbitrary Executable
 File Formats File System Automount Point.
2015-02-04T10:53:58.950809-07:00 linux systemd[1]: Set up automount Arbitrary Ex
ecutable File Formats File System Automount Point.
2015-02-04T10:53:58.950835-07:00 linux systemd[1]: Started Set Up Additional Bin
ary Formats.
2015-02-04T10:53:58.950867-07:00 linux systemd[1]: Mounting Debug File System...
2015-02-04T10:53:58.947597-07:00 linux kernel: [    0.000000] Initializing cgrou
p subsys cpuset
2015-02-04T10:53:58.960981-07:00 linux kernel: [    0.000000] Initializing cgrou
p subsys cpu
2015-02-04T10:53:58.961001-07:00 linux kernel: [    0.000000] Initializing cgrou
p subsys cpuacct
2015-02-04T10:53:58.961012-07:00 linux kernel: [    0.000000] Linux version 3.11
.6-4-default (geeko@buildhost) (gcc version 4.8.1 20130909 [gcc-4_8-branch revis
ion 202388] (SUSE Linux) ) #1 SMP Wed Oct 30 18:04:56 UTC 2013 (e6d4a27)
2015-02-04T10:53:58.961024-07:00 linux kernel: [    0.000000] e820: BIOS-provide
d physical RAM map:
2015-02-04T10:53:58.961382-07:00 linux kernel: [    0.000000] BIOS-e820: [mem 0x
0000000000000000-0x000000000009fbff] usable
2015-02-04T10:53:58.961397-07:00 linux kernel: [    0.000000] BIOS-e820: [mem 0x
000000000009fc00-0x000000000009ffff] reserved
2015-02-04T10:53:58.961408-07:00 linux kernel: [    0.000000] BIOS-e820: [mem 0x
00000000000f0000-0x00000000000fffff] reserved
2015-02-04T10:53:58.961419-07:00 linux kernel: [    0.000000] BIOS-e820: [mem 0x
/var/log/messages lines 1-15/4238 0%
```

- Prints the output to screen page by page
  - Stops after the first page (lines 1 to 15 in this example)

Viewing File Contents

- head /var/log/messages example:

```
openSUSE:~ # head /var/log/messages
2015-02-04T10:53:58.918399-07:00 linux rsyslogd: [origin software="rsyslogd" swV
ersion="7.4.4" x-pid="517" x-info="http://www.rsyslog.com"] start
2015-02-04T10:53:58.923512-07:00 linux systemd[1]: Starting Setup Virtual Consol
e...
2015-02-04T10:53:58.950752-07:00 linux systemd[1]: Starting Arbitrary Executable
 File Formats File System Automount Point.
2015-02-04T10:53:58.950809-07:00 linux systemd[1]: Set up automount Arbitrary Ex
ecutable File Formats File System Automount Point.
2015-02-04T10:53:58.950835-07:00 linux systemd[1]: Started Set Up Additional Bin
ary Formats.
2015-02-04T10:53:58.950867-07:00 linux systemd[1]: Mounting Debug File System...
2015-02-04T10:53:58.947597-07:00 linux kernel: [    0.000000] Initializing cgrou
p subsys cpuset
2015-02-04T10:53:58.960981-07:00 linux kernel: [    0.000000] Initializing cgrou
p subsys cpu
2015-02-04T10:53:58.961001-07:00 linux kernel: [    0.000000] Initializing cgrou
p subsys cpuacct
2015-02-04T10:53:58.961012-07:00 linux kernel: [    0.000000] Linux version 3.11
.6-4-default (geeko@buildhost) (gcc version 4.8.1 20130909 [gcc-4_8-branch revis
ion 202388] (SUSE Linux) ) #1 SMP Wed Oct 30 18:04:56 UTC 2013 (e6d4a27)
```

- Prints the first few lines of the text file to the screen

Viewing File Contents

- tail /var/log/messages example:

```
openSUSE:~ # tail /var/log/messages
2016-11-29T11:00:00.406777-07:00 openSUSE systemd-logind[494]: Removed session 1
5.
2016-11-29T11:00:01.382266-07:00 openSUSE /usr/sbin/cron[4220]: pam_unix(crond:s
ession): session opened for user root by (uid=0)
2016-11-29T11:00:01.391180-07:00 openSUSE systemd[1]: Starting Session 24 of use
r root.
2016-11-29T11:00:01.395492-07:00 openSUSE systemd[1]: Started Session 24 of user
 root.
2016-11-29T11:00:01.428765-07:00 openSUSE /USR/SBIN/CRON[4220]: pam_unix(crond:s
ession): session closed for user root
2016-11-29T11:00:04.318898-07:00 openSUSE login: pam_unix(login:session): sessio
n opened for user root by LOGIN(uid=0)
2016-11-29T11:00:04.321247-07:00 openSUSE systemd[1]: Starting Session 25 of use
r root.
2016-11-29T11:00:04.321513-07:00 openSUSE systemd[1]: Started Session 25 of user
 root.
2016-11-29T11:00:04.322675-07:00 openSUSE systemd-logind[494]: New session 25 of
 user root.
2016-11-29T11:00:04.325027-07:00 openSUSE login: ROOT LOGIN ON tty1
```

- Prints the last few lines of a text file to the screen

Deleting Files

- Using the rm Utility
  - Operates much the same as when removing directories
  - Be careful as it does not ask you twice about deleting a file

```
student@openSUSE:~> ls
bin       Documents  Music    Pictures  public_html       temp       test.txt
Desktop   Downloads  MyFiles  Public    superawesomefile  Templates  Videos
student@openSUSE:~> rm superawesomefile
student@openSUSE:~> ls
bin       Documents  Music    Pictures  public_html  Templates  Videos
Desktop   Downloads  MyFiles  Public    temp         test.txt
```

- In this example the superawesomefile residing in the /home/student directory was deleted via the rm utility.
  - Notice it did not ask for confirmation, rm just deleted it

Copying and Moving Files

- Use the cp and mv Utilities:

```
student@openSUSE:~> ls
bin       Documents  Music    Pictures  public_html  Templates  Videos
Desktop   Downloads  MyFiles  Public    temp         test.txt
student@openSUSE:~> cp test.txt /tmp
student@openSUSE:~> ls /tmp
1682198942  kde-student  ksocket-student  virtuoso_wZ3608.ini
kde-kdm     ksocket-kdm  test.txt
student@openSUSE:~> ls
bin       Documents  Music    Pictures  public_html  Templates  Videos
Desktop   Downloads  MyFiles  Public    temp         test.txt
student@openSUSE:~> mv test.txt ~/Pictures
student@openSUSE:~> ls
bin       Documents  Music    Pictures  public_html  Templates
Desktop   Downloads  MyFiles  Public    temp         Videos
student@openSUSE:~> ls Pictures
test.txt
```

- In this example the test.txt file is copied to the tmp folder and then moved to the Pictures folder. The ls utility is used to verify.

Determining the File Type

Using the file Utility

- Determines and displays a file's type

```
student@openSUSE:~> file handydandyfile
handydandyfile: empty
student@openSUSE:~> file homework.txt
homework.txt: empty
student@openSUSE:~> file /tmp/virtuoso_w23608.ini
/tmp/virtuoso_w23608.ini: ASCII text
student@openSUSE:~> file /var/log/mail.err
/var/log/mail.err: regular file, no read permission
```

In this example the file utility is used to display information about four different files

Working with Link Files

❑ Hard-Link

- file that points directly to the inode of another file
- stores basic information about a file
  - size, device, owner, and permissions

❑ Symbolic-Link

- points to another file in the file system
- has its own inode, because of this the pointer and the pointee in the file system can be easily identified

```
lrwxrwxrwx 1 root root        8 Nov  6  2013 vi -> /bin/vim
lrwxrwxrwx 1 root root        3 Nov  6  2013 view -> vim
-rwxr-xr-x 1 root root   116620 Sep 28  2013 vig_optimize
lrwxrwxrwx 1 root root        8 Nov  6  2013 vim -> /bin/vim
lrwxrwxrwx 1 root root        3 Nov  6  2013 vimdiff -> vim
```

Working with Link Files

- Using the ln utility to create links
  - -s option is used for symbolic links

```
student@openSUSE:~> ln -s /usr/share/gimp/ ~/gimp
student@openSUSE:~> ls-l
total 48
drwxr-xr-x 2 student users 4096 Nov 29 12:32 bin
drwxr-xr-x 2 student users 4096 Feb  4  2015 Desktop
drwxr-xr-x 2 student users 4096 Feb  4  2015 Documents
drwxr-xr-x 2 student users 4096 Feb  4  2015 Downloads
lrwxrwxrwx 1 student users   16 Nov 29 12:39 gimp -> /usr/share/gimp/
-rw-r--r-- 1 student users    0 Nov 29 11:34 handydandyfile
-rw-r--r-- 1 student users    0 Nov 29 11:35 homework.txt
drwxr-xr-x 2 student users 4096 Feb  4  2015 Music
drwxr-xr-x 5 student users 4096 Nov 29 08:59 MyFiles
drwxr-xr-x 2 student users 4096 Nov 29 11:20 Pictures
drwxr-xr-x 2 student users 4096 Feb  4  2015 Public
drwxr-xr-x 2 student users 4096 Feb  4  2015 public_html
drwxr-xr-x 3 student users 4096 Nov 29 08:49 temp
drwxr-xr-x 2 student users 4096 Feb  4  2015 Templates
drwxr-xr-x 2 student users 4096 Feb  4  2015 Videos
student@openSUSE:~> cd gimp
student@openSUSE:~/gimp> ls
2.0
student@openSUSE:~/gimp> pwd
/home/student/gimp
```

- In this example a symbolic link to the gimp directory in /usr/share/gimp to the /home/student directory

Exercise 4-3: Managing Files and Directories

Please open your Practical Exercise book to Exercise 4-3.

Time to Complete: 5 Minutes

Finding Content within Files

Using the grep Utility

- Search through a file for a particular text string
- Useful when you want to search a very large log file for a specific message
- Useful when you need to find a specific directive within a configuration file

```
openSUSE:~ # grep klogd /var/log/messages
2015-02-04T10:53:59.174375-07:00 linux kernel: [   32.857333] type=1400 audit(14
23072421.252:3): apparmor="STATUS" operation="profile_load" name="/sbin/klogd" p
id=387 comm="apparmor_parser"
2015-02-04T11:45:19.275267-07:00 openSUSE kernel: [   32.224952] type=1400 audit
(1423075509.550:3): apparmor="STATUS" operation="profile_load" name="/sbin/klog"
 pid=388 comm="apparmor_parser"
2015-02-04T15:41:56.579341-07:00 openSUSE kernel: [   31.781773] type=1400 audit
(1423089707.218:3): apparmor="STATUS" operation="profile_load" name="/sbin/klog"
 pid=387 comm="apparmor_parser"
2015-02-04T16:04:35.097235-07:00 openSUSE kernel: [   33.637876] type=1400 audit
(1423091064.923:3): apparmor="STATUS" operation="profile_load" name="/sbin/klog"
 pid=387 comm="apparmor_parser"
2015-02-04T16:11:25.432646-07:00 openSUSE kernel: [   31.807599] type=1400 audit
(1423091476.191:3): apparmor="STATUS" operation="profile_load" name="/sbin/klog"
 pid=389 comm="apparmor_parser"
2016-11-29T06:13:14.516036-07:00 openSUSE kernel: [   10.381153] type=1400 audit
(1480425192.928:3): apparmor="STATUS" operation="profile_load" name="/sbin/klog"
 pid=394 comm="apparmor_parser"
```

Finding Content within Files

- When working with grep at the command line, you can use the following options:
  - `-i` Ignores case when searching for the text

```
openSUSE:~ # grep -i KLOGD /var/log/messages
2015-02-04T10:53:59.174375-07:00 linux kernel: [   32.857333] type=1400 audit(14
23072421.252:3): apparmor="STATUS" operation="profile_load" name="/sbin/klogd" p
id=387 comm="apparmor_parser"
2015-02-04T11:45:19.275267-07:00 openSUSE kernel: [   32.224952] type=1400 audit
(1423075509.550:3): apparmor="STATUS" operation="profile_load" name="/sbin/klog"
```

  - `-l` Displays only the names of the files that contain the matching text
    - It doesn't display the actual matching line of text

```
openSUSE:~ # grep -l klogd /var/log/messages
/var/log/messages
```

Finding Content within Files

❑ grep options continued

- **–n Displays matching line numbers**

```
openSUSE:~ # grep -n klogd /var/log/messages
420:2015-02-04T10:53:59.174375-07:00 linux kernel: [   32.857333] type=1400 audi
t(1423072421.252:3): apparmor="STATUS" operation="profile_load" name="/sbin/klog
d" pid=387 comm="apparmor_parser"
1167:2015-02-04T11:45:19.275267-07:00 openSUSE kernel: [   32.224952] type=1400
audit(1423075509.550:3): apparmor="STATUS" operation="profile_load" name="/sbink
logd" pid=388 comm="apparmor_parser"
1877:2015-02-04T15:41:56.579341-07:00 openSUSE kernel: [   31.781773] type=1400
audit(1423089707.218:3): apparmor="STATUS" operation="profile_load" name="/sbink
logd" pid=387 comm="apparmor_parser"
2548:2015-02-04T16:04:35.097235-07:00 openSUSE kernel: [   33.637876] type=1400
audit(1423091064.923:3): apparmor="STATUS" operation="profile_load" name="/sbink
logd" pid=387 comm="apparmor_parser"
3210:2015-02-04T16:11:25.432646-07:00 openSUSE kernel: [   31.807599] type=1400
audit(1423091476.191:3): apparmor="STATUS" operation="profile_load" name="/sbink
logd" pid=389 comm="apparmor_parser"
3929:2016-11-29T06:13:14.516036-07:00 openSUSE kernel: [   10.381153] type=1400
audit(1480425192.928:3): apparmor="STATUS" operation="profile_load" name="/sbink
logd" pid=394 comm="apparmor_parser"
```

- **–r Searches recursively through subdirectories of the path specified**
- **–v Displays all lines that do not contain the search string**

Finding Content within Files

❑ Regular Expressions (used with egrep)

Metacharacter	Function	Example
*	Matches any number of any characters	**Myfile*** would match Myfile1, Myfile2, Myfiles, Myfiles23, etc.
.	Matches a single character	**Myfile.** would match Myfile1, Myfile2, Myfiles, but would not match Myfiles23
^	Matches an expression if it appears at the beginning of a line	**^server** would match any instance of "server" as long as it appears at the beginning of a line
$	Matches an expression if it appears at the end of a line	**server$** would match any instance of "server" as long as it appears at the end of a line.
|	Matches the expressions on either side of the pipe character	**Server|server** would match either "Server" or "server."
[nnn]	Matches any one character between the braces	**[xyz]** would match any one of "x," "y," or "z."
[^nnn]	Matches an expression that does not contain any one of the characters specified	**[^aei]** would not match the characters "a," "e," and "i."
[n-n]	Matches any single character in the range	**[1-5]** would match any number between 1 and 5.

Finding Content within Files

- egrep
  - Search file(s) for lines that match an extended expression (extended grep)
  - egrep is the same as `grep -E‘

```
student@openSUSE:~> egrep MyFiles*
egrep: MyFiles1: Is a directory
egrep: MyFiles2: Is a directory
egrep: MyFiles3: Is a directory
egrep: MyFilesa: Is a directory
egrep: MyFilesb: Is a directory
egrep: MyFilesc: Is a directory
```

```
student@openSUSE:~> egrep luv file[123]
file2:I luv vi it so much fun to use.
student@openSUSE:~> egrep love file[123]
file1:I love vi it is so much fun to use.
student@openSUSE:~> egrep nano file[123]
file3:I hate vi. Why can't we use nano?
```

```
student@openSUSE:~> egrep vi file[123]
file1:I love vi it is so much fun to use.
file2:I luv vi it so much fun to use.
file3:I hate vi. Why can't we use nano?
```

Finding Content within Files

- fgrep
  - Search file(s) for lines that match a fixed string
  - Syntax: fgrep <options> ...
  - fgrep is the same as `grep -F‘

```
student@openSUSE:~> fgrep Routers *.txt
notsogreat.txt:Routers are awesome!
student@openSUSE:~> fgrep Workstations *.c
oldapp.c:Workstations are awesome!
student@openSUSE:~> fgrep servers *.ini
working.ini:servers are awesome!
```

- In the above example the fgrep utility is used to search through three different file types for three different strings.

Exercise 4-4: Using grep

Please open your Practical Exercise book to Exercise 4-4.

Time to Complete: 5 Minutes

Summary

- ❑ Understanding Linux file systems and the Filesystem Hierarchy Standard (FHS)
- ❑ Finding files in the Linux file system
- ❑ Managing directories from the command line
- ❑ Managing files from the command line
- ❑ Working with link files
- ❑ Finding content within files

Questions

Questions?

Check on Learning

Question 1

Which of the following are roles of the Linux file system? (Choose two.)

A. Data availability due to an automated backup process.

B. Make data easily locatable.

C. Preserve data integrity.

D. Provide the user with a command-line interface.

E. Provide the user with a graphical user interface.

Check on Learning

Question 2

Which directory contains file system management utilities such as cp and rm?

A. /bin

B. /dev

C. /var

D. /usr

Check on Learning

Question 3

Which directory does the ~ character represent when used with file system commands?

A. /var

B. The current directory.

C. The current user's home directory.

D. The root user's home directory.

Check on Learning

Question 4

You need to generate a listing of files and directories within the /tmp directory, including files in subdirectories. Which command will do this?

A. ls /tmp

B. ls –l /tmp

C. ls –f /tmp

D. ls –R /tmp

Check on Learning

Question 5

You need to view the last few lines of the /var/log/messages file. Which is the best command to do this from the choices below?

A. tail /var/log/messages

B. cat /var/log/messages

C. head /var/log/messages

D. less /var/log/messages

Check on Learning

Question 6

You need to delete a directory named Stuff within your user's home directory. The Stuff directory has files in it, which of the below commands would delete the Stuff directory?

A. rmdir Stuff

B. rmdir ~/Stuff

C. rm Stuff

D. rm –r ~/Stuff

Check on Learning

Question 7

You need to copy the Pictures directory within you user's home directory to the /tmp directory. Given that Documents has files and subdirectories within it, which is the correct command to do accomplish this task?

A. cp ~/Pictures /tmp

B. cp –R ~/Pictures /tmp

C. cp ~/Pictures ~/tmp

D. cp –R ~/Pictures ~/tmp

Check on Learning

Question 8

You want to create a symbolic link in your home directory that will link the manual file to the /usr/share/doc/manual directory. Which of the below commands will accomplish this?

A. ln –s /usr/share/doc/manual ~/manual

B. ln /usr/share/doc/manual ~/manual

C. ln –s ~/manual /usr/share/doc/manual

D. ln ~/manual /usr/share/doc/manual

Check on Learning

Question 9

You need to find a file named payroll.txt somewhere in your Linux file system. Which of the below commands will accomplish this?

A. find / -name “payroll.txt”

B. find “payroll.txt”

C. find / -n payroll.txt

D. find –name “myfile.txt”

Check on Learning

Question 10

You need to find all entries in your /var/log/messages file that contain "scsi." Which of the below commands will accomplish this?

A. grep /var/log/messages "scsi"

B. grep –find "scsi" /var/log/messages

C. grep /var/log/messages scsi

D. grep scsi /var/log/messages