COURSE TECHNOLOGY
CENGAGE Learning™

Linux Operations and Administration

Ahmad AlRjoub
ahmadrj@ksu.edu.sa

Chapter Three
Managing Files and Directories

Objectives

- Describe the Linux file system and the Filesystem Hierarchy Standard
- Navigate the Linux directory structure
- Manage files and directories in Linux

An Overview of the Linux Directory Structure

- File system
  - The way files are stored and organized to simplify access to data
- Linux has only one root directory
  - All files and subdirectories are placed under the root directory in a treelike structure
- Filesystem Hierarchy Standard (FHS)
  - Specifies requirements and guidelines for file and directory placement in UNIX-like operating systems

An Overview of the Linux Directory Structure (cont'd.)

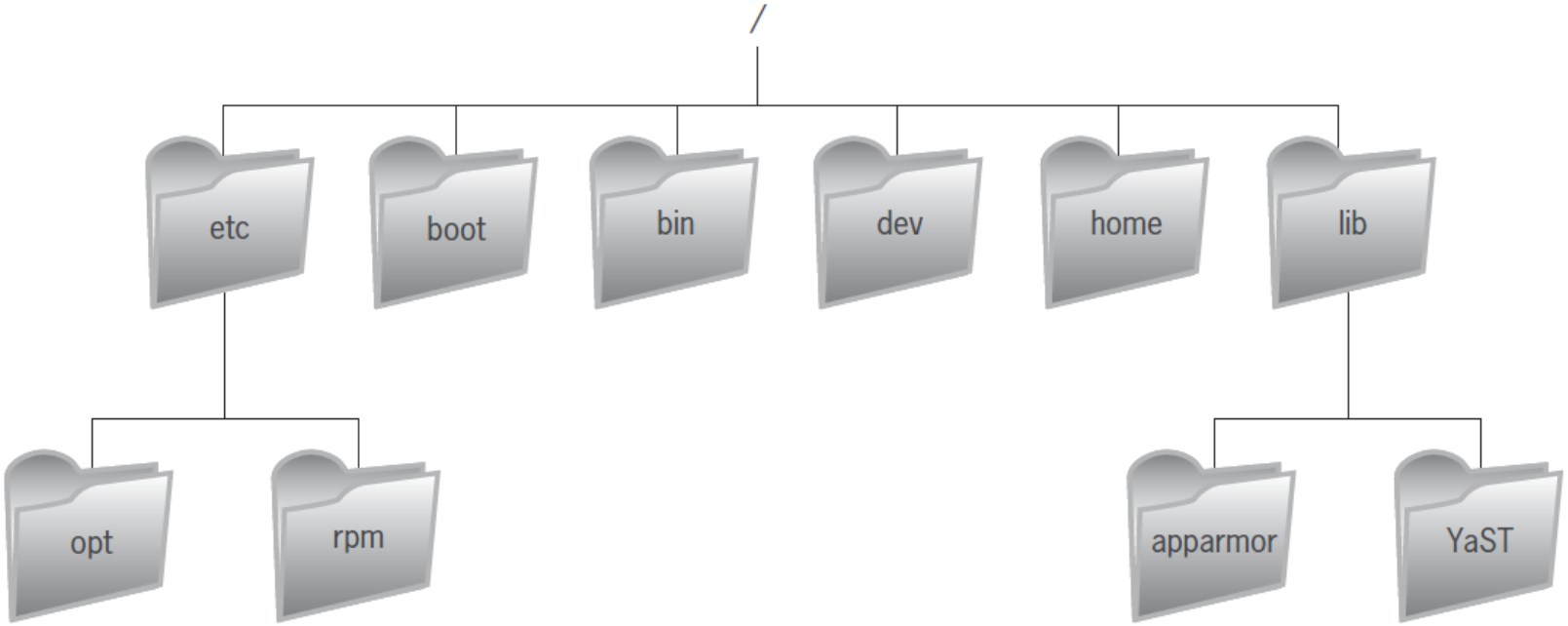


Figure 3-2 The Linux directory structure

An Overview of the Linux Directory Structure (cont'd.)

- Being able to find critical configuration files quickly cuts down on troubleshooting time
- File systems based on FHS have two distinctions:
  - Shareable versus unshareable files
  - Variable versus static files
- Shareable file
  - Can be stored on one machine and used by multiple users on other machines
- Unshareable file
  - Can't be accessed by multiple users

An Overview of the Linux Directory Structure (cont'd.)

- Static files
  - Don't change on their own
- Variable files
  - Usually found in the `/var` directory
  - Can change on their own
- Table 3-1
  - Linux directories defined by the FHS

Directory	Description
/bin	Contains binary commands that can be used by system administrators, users, and scripts; this directory shouldn't contain subdirectories and can be accessed in single user mode
/boot	Contains the Linux kernel and static files needed to boot the computer
/dev	Contains device files, such as the CD/DVD-ROM drive
/etc	Contains static configuration files, which are also unshareable files, meaning they're local to the machine
/home	An optional directory that might not be included in all Linux distributions; in openSUSE, it's the user's home directory
/lib	Contains shared libraries that are loaded when a program starts
/media	Contains the mount point for removable media
/mnt	Empty by default, but administrators can use it to mount other resources, such as CD/DVD-ROM drives
/opt	Contains static shareable add-on software packages
/root	Contains the recommended home directory for the root user; not all Linux distributions use it, but it's used in openSUSE
/sbin	Contains system binaries used by the system administrator
/srv	Contains data files for services
/tmp	Contains temporary files that system administrators should delete whenever the system is booted
/usr	Contains shareable, read-only applications and files
/var	Contains variable data files, such as log files

Table 3-1 Directories defined by the FHS

Navigating the Linux Directory Structure

- Most Linux servers are installed without a graphical environment
  - Administrators need to learn how to navigate without a GUI
- To open a terminal window:
  - KDE—Click the K menu button (the green circle) at the lower left and click Terminal
  - GNOME—Click Computer, More Applications to open the Application Browser, and then click the GNOME Terminal icon in the System Groups category

Changing Directories

- `pwd` (print working directory) command
  - Displays the directory you're currently working in

```
~> pwd
/home/dustin
```

- `~>`
  - Command-line prompt
  - Indicates where to enter commands
  - Varies depending on the shell

Changing Directories (cont'd.)

- `cd` (change directory) command
  - Switch to other directories
  - Can add a command-line argument
    - Information entered after a command to include specific instructions

```
~> cd /bin
/bin> pwd
/bin
```

- `~` (tilde) symbol
  - Represents the user's home directory
  - Directory where the user has full permission to store files

Changing Directories (cont'd.)

- Example: User's current directory is `/bin`
  - `~` symbol is used to change to the user's home directory

```
/bin> cd ~
~> pwd
/home/isaiah
```

- Can use the `~` symbol to specify another user's home directory
  - `~> cd ~jasmine`

Pathnames in Linux

- Absolute path method
  - States the full pathname starting from root (/)
- Relative path method
  - Specifies the pathname starting from the current directory

```
~/Desktop> cd ../../../etc
~/etc> cd ..
~> cd home/jake
```

Pathnames in Linux (cont'd.)

- `. .` (two dots)
  - Navigate to a directory above your current directory with the relative method

```
~/Desktop> cd ..
/jake> pwd
/home/jake
```

The BASH Command Completion Feature

- Finish partially typed commands
- Press the Tab key to enable
- Enter enough characters for this feature to work
  - Particularly if more than one subdirectory begins with the same letter

The BASH Command Completion Feature (cont'd.)

- Activity 3-1: Using the `pwd` and `cd` Commands
  - Practice navigating the Linux directory structure

Viewing Files and Directories

- `ls` command
  - Lists files and subdirectories in the current directory
  - Use arguments to specify other directories

```
~> ls
bin Download
Desktop Documents
```

Viewing Files and Directories (cont'd.)

- Options
  - Modify the way a command is carried out
  - Syntax: `command -options argument`
  - Must include hyphen before the first option you use

```
~> ls -l
total 548
drwxr-xr-x 2 sarah users 4096
2012-03-06 20:01 bin
drwxr-xr-x 2 sarah users 4096
2012-03-09 09:42 Desktop
```

Viewing Files and Directories (cont'd.)

- `-a` option
  - Displays all files including hidden files

```
~> ls -a
. .cache .local bin Desktop
Documents Download .. file1
```

- Table 3-2
  - Most common options for `ls`

Viewing Files and Directories (cont'd.)

Option	Description
`-a`	Lists all files, including hidden files
`-F`	Appends a special character to each filename to represent the file type, such as * for an executable file and / for a subdirectory
`-h`	Stands for "human-readable" format, which shows file sizes in megabytes or gigabytes, for example, instead of in bytes
`-i`	Displays the inode number (discussed later in "Creating Links") for each file
`-l`	Changes the display from a column format to a long list
`-R`	Stands for recursive, meaning the `ls` command is repeated for all subdirectories
`--help`	Lists all options available with a command

Table 3-2 Options with the `ls` command

Examining the `ls -l` Command

- Output of the `ls -l` command contains important information in eight separate columns
- Example: `drwxr-xr-x 2 natalie users 4096 2012-02-06 20:01 Desktop`
- Columns:
  - File type: “d,” which stands for a directory
  - File permissions: displayed for three categories: user, group, and other
  - Hard links: number of hard links associated with the file

Examining the `ls -l` Command (cont'd.)

- Owner—user owner of the file
- Group—file's group owner
- File size—in bytes by default
- Modification time—timestamp showing when the file was last modified
- Filename—name of the file

Examining the `ls -l` Command (cont'd.)

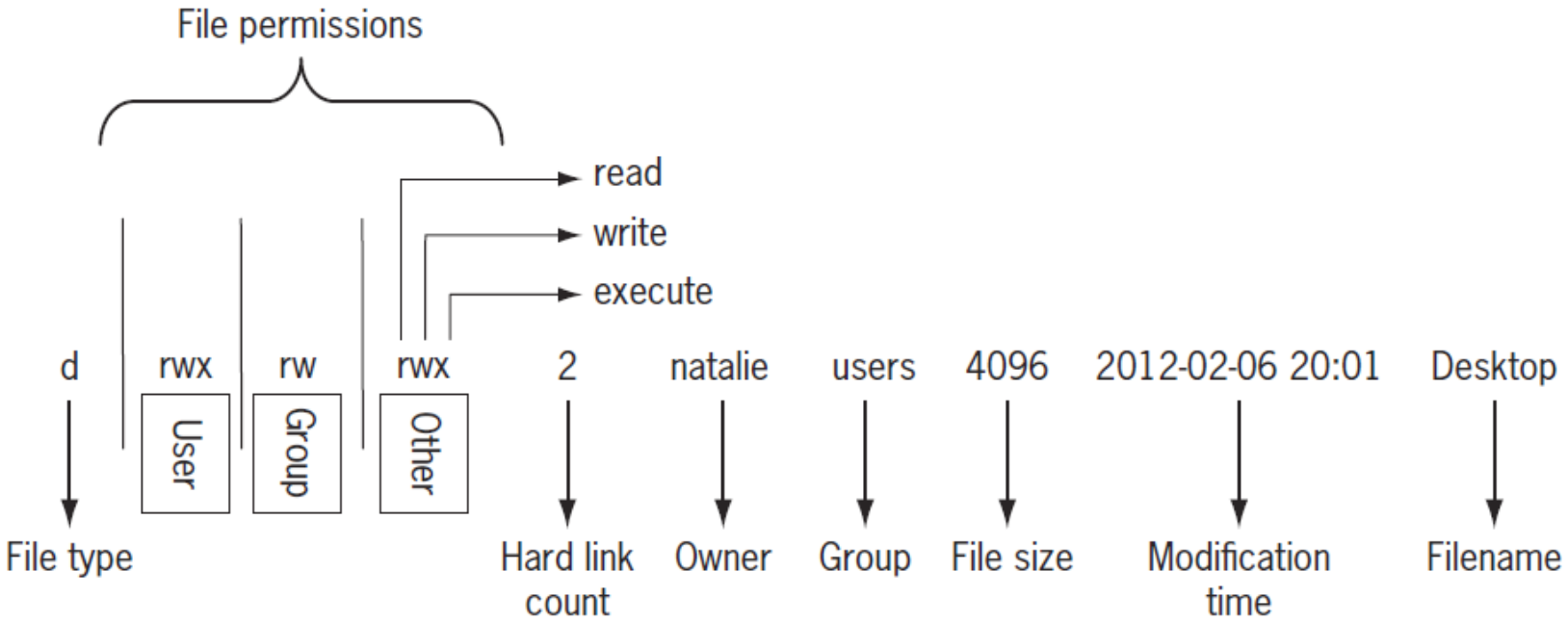


Figure 3-3 A guide to the `ls -l` command's output

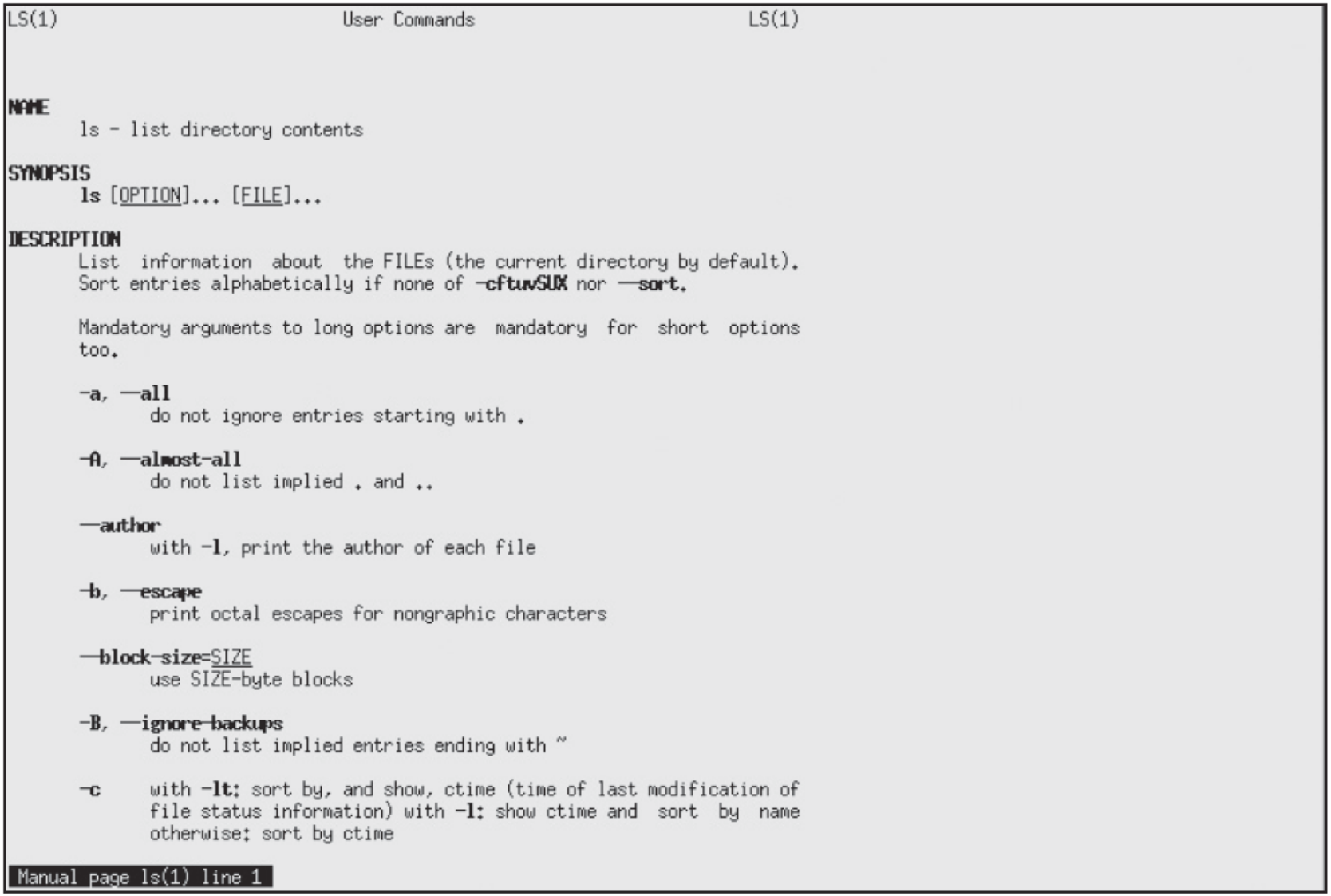
Examining the `ls -l` Command (cont'd.)

- Activity 3-2: Using the `ls` Command
  - Use commands for viewing Linux files and directories

Getting Help

- Man (manual) pages
  - Documentation files that describe Linux shell commands, executable programs, system calls, special files

```
man ls
Man: find all matching manual
pages
* ls (1)
ls (1p)
Man: What manual page do you
want?
```



```
LS(1)                          User Commands                          LS(1)



NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort.

       Mandatory arguments to long options are  mandatory  for  short  options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print octal escapes for nongraphic characters

       --block-size=SIZE
              use SIZE-byte blocks

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification of
              file status information) with -l: show ctime and  sort  by  name
              otherwise: sort by ctime

Manual page ls(1) line 1
```

Figure 3-4 Excerpt from the man page for the `ls` command

Getting Help (cont'd.)

Section	Description	Examples
1	Executable programs or shell commands	`man ls`, `man pwd`
2	System calls, which are system requests that programs make to the kernel	`man kill`, `man read`
3	Library calls (to access functions in program libraries)	`man xcrypt`, `man stdin`
4	Special files, such as the floppy disk, that are usually found in `/dev`	`man fd`, `man tty`
5	File formats and conventions	`man passwd`, `man hosts`
6	Games	`man tetravex`, `man AisleRiot`
7	Macro packages and conventions	`man man (7)`, `man gruff (7)`
8	System administration commands	`man yast`, `man suseconfig`

Table 3-3 Man page section

Navigating Man Pages

- Table 3-4
  - Lists ways to navigate man pages
- Activity 3-3: Working with Man Pages
  - Find and navigate man pages for any Linux command

Navigating Man Pages (cont'd.)

Action	Function
Press f or the spacebar	Move forward one window at a time.
Press b or backspace	Move backward one window at a time.
Press h	Open the help page.
Press / (forward slash)	Enter a string of text to search for in the man page.
Press n	Repeat the previous search.
Press N	Repeat the previous search in the reverse direction.

Table 3-4 Methods of navigating man pages

Using Wildcards

- Wildcard
  - Represents letters and characters used to specify a filename for searches
  - Linux administrators use wildcards to:
    - Navigate to directories faster
    - Move or delete a group of files
    - Locate files based on a portion of their filenames
- Table 3-5
  - Describes wildcards used in Linux

Using Wildcards (cont'd.)

Wildcard	Description
*	Matches zero or more characters in a filename
?	Matches any one character in a filename
[acf]	Matches one of multiple characters in a filename; in this example, a, c, or f
[a-f]	Matches one of a range of characters in a filename; in this example, any character from a through f
[!a-f]	Matches filenames that don't contain a specified range of characters; in this example, filenames that don't contain a through f

Table 3-5 Wildcards

Using Wildcards (cont'd.)

- `*` wildcard
  - Represents zero or more characters

```
ls fi*
file1 file2
```

- `?` wildcard
  - Represents only one character

```
~/newdirectory> ls file?
file1 file2
```

Using Wildcards (cont'd.)

- Activity 3-4: Using Wildcards
  - Use wildcards to search for files and directories

Managing Files and Directories

- Linux administrator should know how to use the command line to:
  - Create files and directories
  - Move files in and out of directories
  - Delete and copy files and directories
  - View file and directory contents

Creating and Deleting Directories and Files

- Directories
  - Essential for keeping files organized in the Linux file system
- Each directory has its own permissions assigned
- Commands to work with directories and files:
  - `mkdir`
  - `touch`
  - `rm`
  - `rmdir`

Creating Directories

- `mkdir` (make directory) command
  - Create directories

```
~> mkdir Studynotes
~> ls
bin Documents Music Public
Studynotes Videos
Desktop Download Pictures
public_html Templates
```

Creating Files

- Several ways to create a file in Linux
- `touch` command
  - Create a new empty file
  - Update the timestamp of an existing file

```
~/Studynotes> touch chapter1
~/Studynotes> ls -l
total 4
-rw-r--r-- 1 keith users 0 2012-
03-19 13:29 chapter1
```

- Use wildcards to update multiple files at the same time

Deleting Directories and Files

- Deleting directories can be dangerous
  - No confirmation message in Linux
- Most Linux users log in with an account that has limited permissions
  - Switch to the root user only to do administrative tasks
- `rm` command
  - Remove files and directories
- `rmdir` command
  - Remove empty directories

Deleting Directories and Files (cont'd.)

```
1. ~/Studynotes> cd Math
2. ~/Studynotes/Math> ls
3. ~/Studynotes/Math> cd ..
4. ~/Studynotes> pwd
5. /home/andrea/Studynotes
6. ~/Studynotes> rmdir Math
7. ~/Studynotes> ls -l
8. total 24
drwxr-xr-x 2 andrea users 4096 2012-03-20 09:27 English
drwxr-xr-x 2 andrea users 4096 2012-03-20 09:27 Week1
drwxr-xr-x 2 andrea users 4096 2012-03-20 09:27 Week2
```

Deleting Directories and Files (cont'd.)

- Remove a directory that isn't empty
  - Error message: `rmdir: failed to remove 'Math': Directory not empty`
- Activity 3-5: Creating and Deleting Directories and Files
  - Use Linux commands for creating and deleting directories and files

Moving, Renaming, and Copying Files

- `mv` command
  - Rename files
  - Move files from one directory to another
- Syntax: `mv filename new location`
- Example: move the `notes` file to another directory and rename it `Newnotes` at the same time:
  - `~/Studynotes/Week1> mv notes ~/Studynotes/Week2/newnotes`

Moving, Renaming, and Copying Files (cont'd.)

- `cp` (copy) command
  - Copies files and directories
- Syntax: `cp filename new location`
- Example: `~/Studynotes/Week2> cp newnotes ~/Studynotes/Week1`
- Activity 3-6: Renaming, Moving, and Copying Files
  - Use Linux commands for renaming, moving, and copying files

Creating Links

- Learn how to create links with the `ln` command
- Learn about inodes

Inodes

- Inode
  - Data structure that stores all information about a file except the actual data and filename
- Inode number
  - Like an address
  - References an entry in the inode table
    - List of inodes for all files on a Linux partition
    - Table entry points to the data’s location on the disk

Inodes (cont'd.)

- `ls -il` command
  - View a file's inode number

```
~/Math> ls -il
total 4
3327 -rw-r--r-- 1 dustin users 0 2012-03-21
15:46 algebra
```

Hard Links

- Hard links
  - Files that point to data on the hard drive
- Create a file
  - Automatically linked to the actual data stored on a partition
  - Assigned an inode number referencing this data
- Create a hard link:
  - `~/Math/Week1/Calculus> ln notes ~/Math`
- Delete files
  - Data isn’t deleted until the last link is deleted

Symbolic (Soft) Links

- Symbolic links
  - Also called soft links
  - Special types of files that point to other files instead of pointing to data on the hard drive
  - Do not share the same inode number
- Benefit of creating a symbolic link
  - Link files that are on separate partitions or even different computers

Symbolic (Soft) Links (cont'd.)

```
~/Sports> ln -s football baseball
~/Sports> ls -il
total 4
3935 lrwxrwxrwx 1 edward users 8 2012-03-22 10:11
baseball -> football
3934 -rw-r--r-- 1 edward users 0 2012-03-22 10:10
football
```

- Activity 3-7: Working with Links
  - Identify inode numbers and create hard and symbolic links

Switching Users and Becoming Root

- Every user must have a username and password and belong to a primary group
- Can switch to a different user account while staying in the same terminal window
- `su` (switch user) command
  - Enables one user to become another user temporarily

```
1. ~> su jasmine
2. Password:
3. jasmine@client:/home/dustin> pwd
4. /home/dustin
5. jasmine@client:/home/dustin> exit
6. exit
7. ~> su
8. Password:
9. client:/home/dustin # pwd
10. /home/dustin
11. client:/home/dustin # exit
12. exit
13. ~>
```

- Activity 3-8: Switching Users
  - Switch users without logging off the computer

Summary

- Nearly all major Linux distributions follow the Filesystem Hierarchy Standard (FHS)
- Commands:
  - `pwd` displays current directory
  - `cd` changes directory
  - `ls` lists contents of a directory
  - `man` shows help files
  - `mkdir` creates a directory
  - `rm` removes a file or directory
  - `mv` moves and renames files

Summary (cont’d.)

- `cp` copies files
- `ln` makes hard and symbolic links
- `su` changes the current user temporarily