File and Directory Operations

OPS102 Week 3 Class 1

Tiayyba Riaz/John Sellens

January 21, 2025

Seneca Polytechnic

Outline

Files in Linux/Unix

Files, Files, Files

- Data is saved in files
- In Linux/Unix we *really* like text files
- For data, presentations, configuration, logs, and more
- The system and shell provide “easy” ways to deal with files
- More about file details next week

Learning About Commands

Manual Labour

- There is extensive documentation readily available on Linux/Unix systems
- Less so on Windows, though commands often provide help e.g. `"dir /?"`
- The ”man” (manual) command provides access to most documentation
- Man pages are divided into sections – see `"man man"`

File and Directory Operations

Topics

- Creating and removing directories
- Moving files and directories
- Copying files/directories
- Creating and deleting files
- Working with file contents

Important File Management Skills

- Create files and directories
- Read the contents of files
- Copy files for backup purposes
- Move or rename incorrectly spelled filenames
- View text file contents without the danger of editing or corrupting those files.
- Remove files
- Check for differences between a couple of files
- Obtain information regarding the status of a file and information regarding the file's contents

We have learned to do these operations in a GUI, now we will learn how to do them on a command line.

File and Directory Operations

Linux	Windows	Usage
mkdir	mkdir	Create directories
mv	move	Move/Rename files/directories
-	rename	Rename files/directories
cp	copy	Create a copy of files/directories
rm	del	Remove files/directories
rmdir	rmdir	Remove empty directories
rm -r	deltree	Recursive directory removal
touch	-	Create empty file/update time
-	copy nul: file.txt	Create empty file

Directory Operations

- Recall the `"cd"` command - change directory
    - There’s also `"pushd"` and `"popd"`
    - These are shell commands (or system library `"chdir()"`)
- `"mkdir"` creates one or more directories
    - The `"-p"` ensures the path/parents exist
- `"rmdir"` removes one or more empty directories
    - Recursive remove – `"rm -r"` – removes non-empty directories

File Operations

- Create files with a text editor
    - Or output from program. output redirection (next week), etc
- `"touch"` will create an empty file (limited utility) or change the file's timestamp
- Copy and move – `"cp"` and `"mv"` – mostly do what you expect
    - One or more sources to a destination
    - Destination can be an existing directory
    - `"mv"` also renames – moves to a new name
- Remove – `"rm"` – removes files or with `"-r"` it removes directories recursively

Working with Text Files

Linux/Unix systems have many tools for working with text files, Windows less so.

Linux	Windows	Usage
cat	copy file con: type	Display the contents of file all at once on screen
more, less	more	Display the contents of file one screen at a time
head, tail	-	Display the beginning or end of file
file	-	Determine the type of file

In Windows you would typically need add-on programs for most of these. Or WSL: Windows Subsystem for Linux.

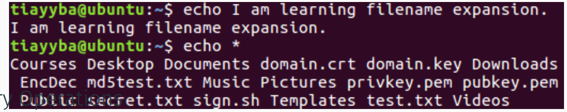
Working with Text Files (cont'd)

Linux	Windows	Usage
sort	-	Sort the lines of file
uniq	-	Display identical consecutive lines only once
cut	-	Remove undesired columns from your data in file
tr	-	Translate/replace the occurrences of characters
grep	findstr	Find specific lines in a file
find	-	Find files matching specific criteria in the filesystem
diff	-	Show the differences between two files

File Globbing

File Globbing

- File globbing is a feature provided by the shell.
- By using special characters called wildcards, we can write a generic name that the shell will expand into the specific matching names.
- A wildcard is a symbol with a special meaning that can be used to substitute for one or more characters.
- When you type a command and press the enter key, bash performs file name expansion on any wildcards on the command line before it executes the command.
- So you type a short form and it is expanded into the full list of matching files (or directories) before the shell executes the command. e.g.



```
tiayyba@ubuntu:~$ echo I am learning filename expansion.
I am learning filename expansion.
tiayyba@ubuntu:~$ echo *
Courses Desktop Documents domain.crt domain.key Downloads
 EncDec md5test.txt Music Pictures privkey.pem pubkey.pem
 Public secret.txt sign.sh Templates test.txt Videos
```

How Does Globbing Work?

- When the enter key is pressed, the shell automatically expands `"*"` into the names of all the files and directories in the current working directory before executing the echo command.
- The echo command never receives `"*"` as an argument, it only receives the result of the filename expansion.
- Wildcards can be used with any command such as `ls`, `rm`, `cp`, etc.
- Example: `"rm *.pdf"` deletes all pdf files in the current directory.
- ”glob” is short for ”global” (or so says wikipedia) and was originally a separate command, or so says `"man 7 glob"`
  - https://en.wikipedia.org/wiki/Glob_(programming)

File Globbing: Wildcard Characters

- The bash shell (like most shells) recognizes 3 types of globbing
    - Windows command has more limited globbing features
- An asterisk `"*"` (or star) represents zero or more characters
- A question mark `"?"` represents exactly one character (any character)
- A set of square brackets represents any one character from the list inside the brackets
    - e.g. `"[pdq]"` represents a `p`, `d`, or `q`.
    - e.g. `"[a-m]"` represents a lower case letter from the range `a` through `m`.
    - e.g. `"[a-zA-Z]"` represents any single letter.

File Globbing: asterisk *

- The asterisk `"*"` is interpreted by the shell to generate filenames by matching the asterisk to any combination of characters (even none).
- When `"*"` is used with the command ls (or any command) and no path is given, the shell will use filenames in the current directory.

Pattern	Interpretation
`*.pdf`	This expands to all file or directory names that end in `.pdf`
`ls *.pdf`	Lists all files (or directories) with the extension `.pdf` e.g. myfile.pdf, cities.pdf, 123.pdf
`rm img*.jpg`	Delete all files with names starting "img" and ending ".jpg" e.g. img001.jpg, imgface.jpg, img500.jpg

File Globbing: question mark ?

- The question mark `"?"` is interpreted by the shell to generate filenames by matching the question mark to any exactly one character (any character).

Pattern	Interpretation
`ls File?.pdf`	Lists all files (or directories) with names starting with `File`, followed by any one character, and then ending with `.pdf` e.g. Filea.pdf, File1.pdf, File2.pdf, FileC.pdf But not File12.pdf – why?
`rm img?.jpg`	Delete all files with names starting `img`, followed by one more character, and ending `.jpg` e.g. img0.jpg and img2.jpg would be deleted But not img50.jpg

File Globbing: square brackets `[]`

- A set surrounded by square brackets `[]` is called a character class.
- It matches any one of the characters contained in the class.
- The class may include ranges; order within the class is not important.

Pattern	Interpretation
`ls File[123].pdf`	List File1.pdf, File2.pdf and File3.pdf (if they exist). It will not list File123.pdf (if it exists) – why?
`rm img[012].jpg`	Delete files that start with img, followed by either 0, 1, or 2, and ending with .jpg. Examples: img0.jpg, img1.jpg and img2.jpg

File Globbing: square brackets `[]` and `!`

- If the first character in a character class is an exclamation mark `!` then the class is inverted.
- i.e. The character class will match any character that is not listed in the class.
- For example
    - `[!a-z]` matches any character that is not a lower case letter.
    - `[!0-9]` matches any character that is not a digit.

Test Yourself

The command `"rm *123??.jpg"` will delete which of the files from the following list?

- Image1230.jpg
- City12345.jpg
- Book12391.jpg
- Pic123me.jpg
- Img123you.jpg