

OPS 102
INTRODUCTION TO OPERATING SYSTEMS

File Systems

Tiayyba Riaz

Topics

- File globbing
- Redirection the output
- Piping multiple commands

File Globbing

- File globbing is a feature provided by shell.
- By using special characters called wildcards, we can write a generic name that Shell will expand into specific names.
- A wildcard is a symbol with special meanings and it 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 the wildcards before it carries out the command.
- So you type something and it is expanded into something else before shell executes it. Look at this example!

```
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 it Work?

- When the enter key is pressed, shell automatically expands * into the names of all the files and directories in the current working directory before executing echo command.
- The echo command never received "*" as argument, it only received its expanded result.
- Wild cards can be used with most commands such as `ls`, `rm`, `cp`
- Example:
`rm *.pdf` deletes all pdf files in the current directory.

File Globbing: Wild Cards

- Bash has three types of wild cards:
 - * **star** –Represents zero or more characters.
 - ? **question mark** –Represents any single character
 - [] **square brackets**–Represents one character from a list given in brackets.

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` and no path is given, the shell will use filenames in the current directory.

Command	Interpretation
<code>*.pdf</code>	This means any name followed by <code>.pdf</code>
<code>ls *.pdf</code>	This command will list all files having <code>.pdf</code> extension. Example: <code>myfile.pdf</code> , <code>cities.pdf</code> , <code>123.pdf</code>
<code>rm img*.jpg</code>	This command will delete all files starting with the word <code>"img"</code> and having <code>.jpg</code> extension. Examples: <code>img001.jpg</code> , <code>imgface.jpg</code> , <code>img500.jpg</code>

File Globbing: question mark ?

- Question mark ? is interpreted by shell as a sign to generate filenames by replacing question mark (?) with exactly one character.

Command	Interpretation
<code>ls File?.pdf</code>	This command will list all files starting with the word “ File ” and having one more character and ending with “.pdf”. For example; Filea.pdf, File1.pdf, File2.pdf, FileC.pdf However, File12.pdf will not be listed, why?
<code>rm img?.jpg</code>	This command will delete all files starting with “ img ” having one more character and “.jpg” extension. img0.jpg, img2.jpg if exist will be deleted, however img50.jpg will not be deleted.

File Globbing: square brackets []

- The square bracket [] represent a character class.
- Shell matches any one character included between square brackets.
- The order in this list between the brackets is not important.

Command	Interpretation
<code>ls File[123].pdf</code>	This command will only list "File1.pdf", "File2.pdf" and "File3.pdf" if they exist. It will not list "File123.pdf" if it exists, why?
<code>rm img[012].jpg</code>	This command will only delete all the files whose name starts with the word "img" followed by either 0 or 1 or 2 and ending with .jpg. Examples: img0.jpg, img1.jpg and img2.jpg

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

Text Editors

- A text editor allows users to create, modify and save editing changes of text files.
- Editing text files is an everyday activity for both programmers as well as administrators on a Unix and Linux system.
 - Edit System configuration files
 - Write Scripts and programs
 - Write/edit documentation
 - Develop web pages
- It is important to learn to be able to use a text editor in order to install, configure and run network services.

Text Editors

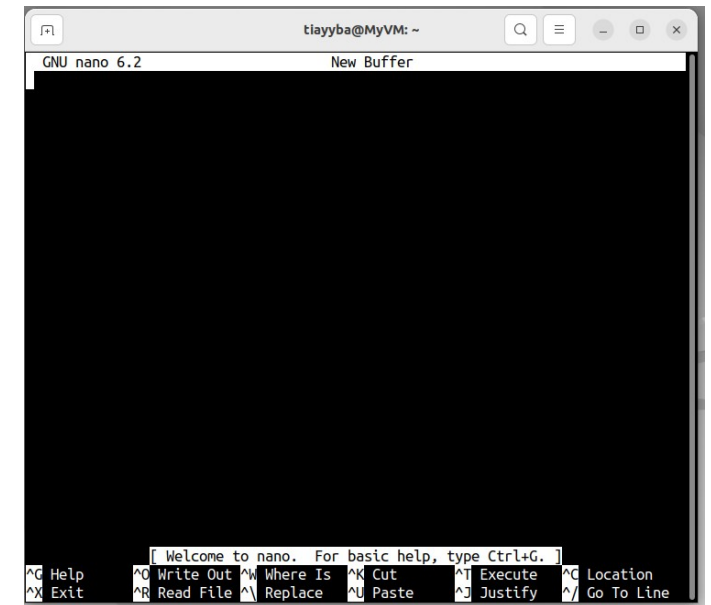
- There are a number of editors available in Linux and Windows.
- It is useful to expose yourself to a multiple text editors and then use one that you feel most comfortable working with.
- Following are some well known text editors:
 - VIM
 - Nano
 - Gedit
 - Notepad
 - Notepad++
 - Sublime
- A list of editors is given [here](#)
- You will learn some basics of nano editor in next slides (you should try other text editors as well). It is also a default editor for many Linux distributions

Text Editors

- Line breaks is an important concept in text editors. Traditionally Windows uses Carriage Return + Line Feed (CR/LF) as line break whereas Linux uses Line Feed character only as line break.
- This may cause some formatting issues when text files are transferred between Windows and Linux.
- A file moved from Windows to Linux might show some empty lines whereas a file moved from Linux to Windows may miss the line-breaks.
- Some editors can detect the later issue and can add line-breaks

Nano

- The Nano text editor is an easy to use text editor.
- Type the command `nano` and the editor will open for you.
- Nano editing commands typically consist of the `^` symbol which represents `<ctrl>` key, followed by another character (command).
- For nano you don't need to memorize the necessary commands, they are shown at the bottom of the nano screen



Nano (Cont)

- Some basic commands include:
 - ^G - to get help at any time
 - ^R - open a file
 - ^O - save a file
 - ^W - find
 - ^\ - replace
 - ^X - exit
- Instructor note: please demonstrate the use of these commands