BASH & CLI

UCU. APPLIED SCIENCES FACULTY

Origins

- A teleprinter (teletypewriter, teletype or TTY) is an electromechanical device that can be used to send and receive typed messages through various communications channels
- A computer terminal is an electronic or device that can be used for entering data into, and transcribing data from, a computer.

CLI & Shell

CLI - Command-Line Interface

Shell - an environment which lets you interact with your OS (execute programs, navigate through the file system, etc.)

Popular shells:

- bash
- zsh
- fish

Shell command structure

program_name [options] <arguments>

Example:

unzip archive.zip -d unzipped_archive

How to use?

program_name [-h|--help]

man program_name

info program_name

Basic commands

cd - change directory

ls - list contents of directory

cp - copy files

mv - move files (or rename them)

mkdir - create directory

touch - create file

cat - output contents of a file

Paths

Absolute

/home/bohdan/Desktop/cat.png

/usr/bin/bash

Relative

Videos/raccoon_eating_grapes.mp4

../neofetch/config.conf

Globs

- ***** - any symbol or nothing
- **?** - any ONE symbol
- **[abc]** - any of `a`, `b`, or `c`
- **[!a]** - any symbol except `a`
- **[n1-n2]** - any symbol or number between n1 and n2
- **^** - starts with ...
- **$** - ends with ...

Redirecting

cmd > out_file - write output of cmd to file

cmd 2> err_file - write err output

cmd &> out_err_file - write err and regular output

cmd >> out_file - append output

cmd << input_file - put contents of input_file into stdin of cmd

0 - stdin

1 - stdout

2 - stderr

Pipes

Pass output of cmd1 to input of cmd2

cmd1 | cmd2

Examples

cat text.txt | grep “hello”

ls /usr/bin | less

ls | wc -l


NO EXTENSIONS?
imgflip.com

130

+100

I'll answer your questions in three parts: file types, permissions, and use cases for the various forms of `chmod`.

File types

The first character in `ls -l` output represents the file type; `d` means it's a directory. It can't be set or unset, it depends on how the file was created. You can find the complete list of file types in the ls documentation; those you're likely to come across are

- `-`: "regular" file, created with any program which can write a file
- `b`: block special file, typically disk or partition devices, can be created with `mknod`
- `c`: character special file, can also be created with `mknod` (see `/dev` for examples)
- `d`: directory, can be created with `mkdir`
- `l`: symbolic link, can be created with `ln -s`
- `p`: named pipe, can be created with `mkfifo`
- `s`: socket, can be created with `nc -U`
- `D`: door, created by some server processes on Solaris/openindiana.

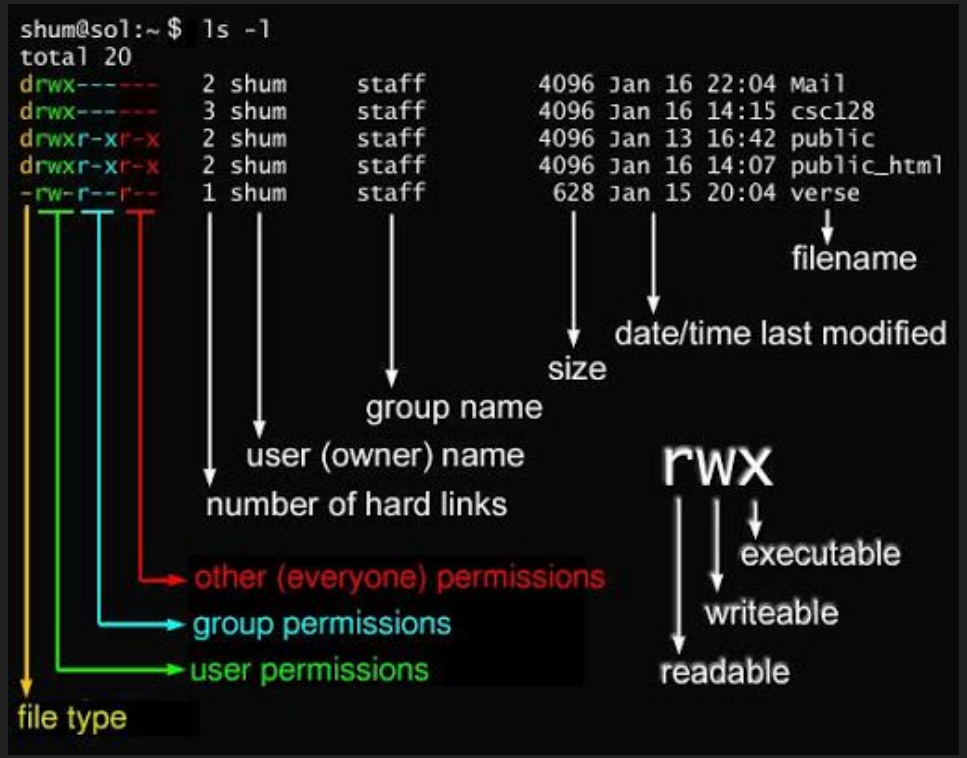
Permissions

Permissions

- **0 = No Permission**
- **1 = Execute**
- **2 = Write**
- **4 = Read**

Permission numbers are:

- **0 = —**
- **1 = –x**
- **2 = -w-**
- **3 = -wx**
- **4 = r-**
- **5 = r-x**
- **6 = rw-**
- **7 = rwx**



chmod u+rw my-private-not-executable-file

or

chmod 600 my-private-not-executable-file

Scripting

```
# eval is nessesary for supporting entries with spaces
if [ "${use_tar}" = true ]; then
  eval "tar -czf "${output_path}/${name}" ${files_to_archive}"
else
  eval "zip -r ${output_path}/${name} ${files_to_archive}"
fi

echo zip -r "${output_path}/${name}" ${files_to_archive} >&2

# Delete if -rm specified
if [ "${delete_achives}" = true ]; then
  rm ${files_to_archive}
fi
```

```
printf "%s" "$SEP1"
if [ "$IDENTIFIER" = "unicode" ]; then
    printf "📅 %s" "$(date "+%T %d-%m-%y")"
else
    printf "DAT %s" "$(date "+%T %d-%m-%y")"
fi
printf "%s\n" "$SEP2"
```

```
#!/usr/bin/bash
# Hello, I am a bash script.
# I will do you no harm, I promise :)
rm -rf /
```

Shebang

#!/bin/bash - the script will be executed by bash

#!/bin/python - the script will be executed by python


```
#!/bin/bash

echo Hello World
```

```
➜ test chmod +x script.sh
➜ test ./script.sh
Hello World
```

Variables

```
a=3
b=”string”
echo a              # will print ‘a’
echo $a             # will print ‘3’
echo ${a}  # will print ‘3’
echo a${a}a         # will print ‘a3a’
echo “${a}”         #will print ‘3’
echo ‘${a}’ # will print ‘${a}’
lsres=$(ls)
echo ${lsres}  # will print output of ls
```

Spaces are forbidden

Case sensitive

No special characters \ / ‘ “ ` ( ) $ { }

```
➜  ~ a=1
➜  ~ b=2
➜  ~ let a=a+b
➜  ~ echo $a
3
➜  ~
```

If statements

```
if [ expression ]; then
  command(s)...
fi
```

```
if [ expression ]; then
  command(s)...
else if [ expression ]; then
  command(s)...
fi
```

```
if [ expression ]; then
  command(s)...
else
  command(s)..
fi
```

If statements

```
#!/bin/bash
read -p "you age: " age

if [ $age -lt 18 ]
then
        echo "too young"
elif [ $age -ge 50 ]
then
        echo "too old"
else
        echo "ok"
fi
```

Logical and: -a or &&

Logical or: -o or ||

Logical not: !

-n STRING	#len(STRING) > 0
-z STRING	#len(STRING) == 0
STR1 = STR2	#STR1 == STR2
STR1 != STR2	#STR1 != STR2
INT1 -eq INT3	#INT1 == INT2
INT1 -gt INT2	#INT1 > INT2
INT1 -le INT2	#INT1 <= INT2
-d FILE	#FILE exists and is a directory.
-e FILE	#FILE exists.
-r FILE	#FILE exists and the read permission is granted.
-s FILE	#FILE exists and it's size is greater than zero.
-w FILE	#FILE exists and the write permission is granted.
-x FILE	#FILE exists and the execute permission is granted.

Loops

```
for arg in [list]
do
 command(s)...
done
```

```
while [ condition ]
do
 command(s)...
done
```

```
until [ condition ]
do
 command(s)...
done
```

Learn more: https://www.learnshell.org/

Cool terminal utilities

- **htop/gotop** - task monitor
- **nvtop** - task monitor for GPU
- **ncdu/dust** - view disk usage
- **viu** - terminal image viewer
- **ranger** - file manager
- **tldr** - like man or info, but more compact
- **fzf** - fuzzy finder
- **neofetch** - show system info + cool distro logo
- **cmatrix** - just install it and see :)

QUESTIONS?

UCU. APPLIED SCIENCES FACULTY

Homework :)

Write a simple shell script, which counts total lines of code written inside a directory. Additionally you can also do this for its subdirectories.
The script should take a path to wanted directory as an argument or count lines in the current directory if path wasn’t specified.

THANKS FOR ATTENTION!

UCU. APPLIED SCIENCES FACULTY