UNIX

Lecture 2: Basic Commands (continued), Shell Metacharacters

Filipe Vasconcelos[1]
[1]ESME, Lille, filipe.vasconcelo@esme.fr

ESME
PARIS•BORDEAUX•LILLE•LYON

Outline

Intended Learning Outcomes

ILO2

Recognize, describe, and use basic Bash commands, or quickly find documentation about these commands.

Class GitHub Repository

File sharing through GitHub

```
$ git clone https://github.com/EsmeEngineeringSchool/unix-etudiant
$ cd unix-etudiant/
```

If git is not installed... `$ sudo apt-get install git`

Outline

Command `echo`

echo

- Usage: Display a string of characters or the value of a variable.
- Simplified syntax: `echo [-n] [string]`
- Option: `-n`: no final line break.
- Example: `echo This is an example`.
- Note: Useful for displaying a variable or an environment variable (e.g., `echo $PATH`).

Command `echo`

First example

```
$ echo "Hello World!"
```

```
Hello World!
$
```

Without a line break

```
$ echo -n "Hello World!"
```

```
Hello World!  $
```

Special characters

```
$ echo -e "\t Hello \n World!"
```

```
    Hello
 World!
```

Command `cat`

cat: **cat**enate

- Usage: Concatenate and display the content of one or more files.
- Simplified syntax: `cat [-n] [file]`
- Option: `-n`: line numbering.
- Examples:
  - `cat file1.txt`: To view a file.
  - `cat file1.txt file2.txt`: To view two concatenated files.

Command `more`

more

- Usage: Display the content of a file page by page.
- Simplified syntax: `more [file]`
- Remarks:
  - `<ENTER>`: display the next line.
  - `<SPACEBAR>`: next page.
  - `<q>`: quit.
  - Enhanced version: `less`

Command `file`

file

- Usage: Determine the type of a file.
- Simplified syntax: `file [file]`
- Option: `-z`: allows inspecting compressed files.
- Example: `file /bin/bash`.

Command `alias`

alias

- Usage: Create a synonym for a command.
- Simplified syntax: `alias synonym='command'`
- Example: `alias ll='ls -l'`
- Note: It's possible to group aliases in a configuration file `~/.bash_aliases`.

Command `exit`

exit

- Usage: Running this command ends the current user session or closes the terminal or user's terminal connection.
- Simplified syntax: `exit`
- Note: Equivalent to `Ctrl+D`

Command history?

history

- Usage: Display the most recent commands executed, preceded by a number.
- Simplified syntax: `history [n_last_commands]`
- Option: `[-c]`: clear history
- Note: The history is stored in the configuration file `~/.bash_history`.

Command `wc`

wc: **w**ord **c**ount

- Usage: Display the number of lines, words, characters, or bytes in a file.
- Simplified syntax: `wc file`
- Options:
  - `-l`: number of lines
  - `-w`: number of space-separated words
  - `-m`: number of characters
  - `-c`: number of bytes

Example

```
$ echo This is a file > file.txt; wc file.txt
```

Output `1 4 20 file.txt`

- Why 20?

Command `find`

find

- Usage: Search for a file in a directory hierarchy. It's possible to search for files whose names match a pattern containing shell metacharacters.
- Simplified syntax: `find path -name name`
- Example: `find . -name "*.txt"` Search in the current directory `./` for files with the extension `.txt`.

Outline

Shell Metacharacters

- They allow you to construct generic character strings.
  - `*`: denotes any string of any characters
  - `?`: denotes any single character
  - `[..]`: denotes characters within brackets, defined by enumeration or by a range
  - `debian, linux`: denotes exactly one whole word from the given options
- Examples:
  - `[Aa]`: denotes characters `A` or `a`
  - `[0-9a-zA-Z]`: denotes any alphanumeric character
  - `b*`: denotes all words starting with the letter `b`
  - `*b*`: denotes all words that contain at least one `b`
  - `b?`: denotes all words with two letters, where the first letter is "b"
- Note: `[!0-9]` denotes all characters except digits.

Shell Metacharacters

Shell metacharacters are often used with basic commands

- `mv /directory/*.pdf /other/part`: move all files with the `pdf` extension from `directory` to `/other/part`.
- `ls t?t?`: display files like `tatb`, `toto`, etc.
- `ls t[oi]t[oi]`: display only files `toto`, `titi`, `toti`, `tito`.
- `find /usr/bin -name '*.sh'`: search for `sh` files in the directory `/usr/bin`.