

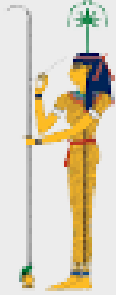
Linux

Linux Course



Mostafa Nabieh

وزارة الاتصالات
وتكنولوجيا المعلومات



MANNING



وزارة الاتصالات
وتكنولوجيا المعلومات
MINISTRY OF COMMUNICATIONS
AND INFORMATION TECHNOLOGY



AI TALENTS
HUB



UDACITY



PLURALSIGHT



YOUR SPACE TO LEARN
FUTURE SKILLS



Mostafa Nabieh





Mostafa Nabieh

DAY 3 CONTENTS

- Shell Scripting Basics
- Filters, Pipes, and variables
- Useful Features of the Bash Shell



What is a script?

- Script: list of commands interpreted by a scripting language
- Commands can be entered interactively or listed in a text file
- Scripting languages are interpreted at runtime
- Scripting is slower to run, but faster to develop

```
[root@tecmint-rhel8:~]# vi if_statement.sh
[root@tecmint-rhel8:~]#
[root@tecmint-rhel8:~]# bash if_statement.sh
Enter the score
40
Work Harder!
[root@tecmint-rhel8:~]# bash if_statement.sh
Enter the score
35
Work Harder!
[root@tecmint-rhel8:~]# bash if_statement.sh
Enter the score
20
Work Harder!
[root@tecmint-rhel8:~]# _
```



Shell scripts and the 'shebang'

- Shell script - executable text file with an interpreter directive
- Aka 'shebang' directive:

```
#!/interpreter [optional-arg]
```

- “interpreter” - path to an executable program
- “optional-org” - single argument string



Example - 'Shebang' directives

- Shell script directive

```
#!/bin/sh
```

```
#!/bin/bash
```

- Python script directive

```
#!/usr/bin/env python3
```



“Hello world” Script

- Create the shell script:

```
mostafa@mostafa-VirtualBox:~/Desktop$ touch hello-world.sh
mostafa@mostafa-VirtualBox:~/Desktop$ echo '#! /bin/bash' >> hello-world.sh
mostafa@mostafa-VirtualBox:~/Desktop$ echo 'echo hello world' >> hello-world.sh
```



“Hello world” Script

- Make it executable:

```
mostafa@mostafa-VirtualBox:~/Desktop$ ls -l hello-world.sh
-rw-rw-r-- 1 mostafa mostafa 30 10:57 18 لک hello-world.sh
mostafa@mostafa-VirtualBox:~/Desktop$ chmod +x hello-world.sh
mostafa@mostafa-VirtualBox:~/Desktop$ ls -l hello-world.sh
-rwxrwxr-x 1 mostafa mostafa 30 10:57 18 لک hello-world.sh
```



“Hello world” Script

- Run your bash script:

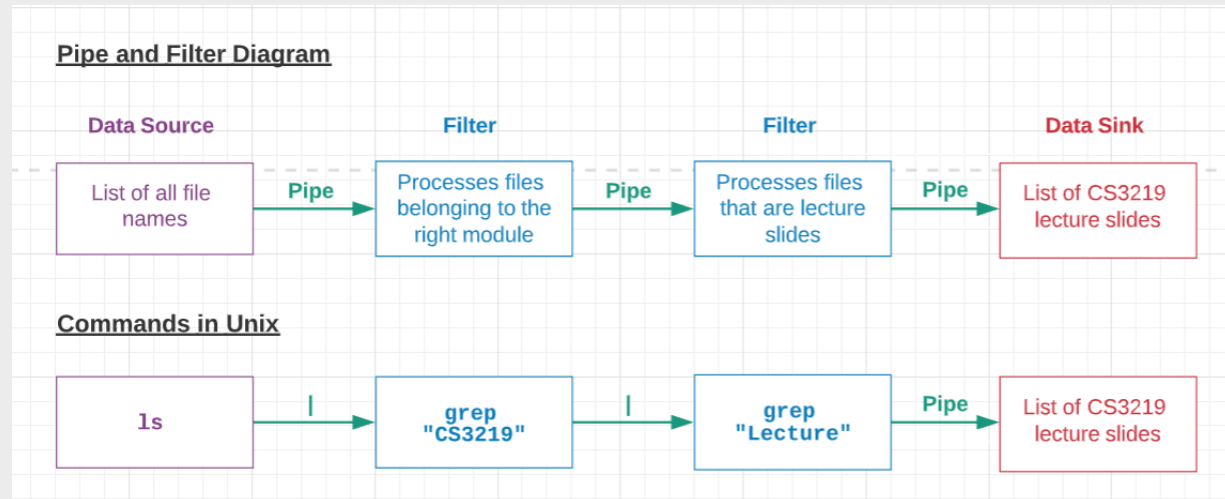
```
mostafa@mostafa-VirtualBox:~/Desktop$ ./hello-world.sh  
hello world
```



Pipes and filters

Filters are shell commands, which:

- Take input from standard input
- Send output to standard output
- Transform input data into output data
- Examples are wc, cat, more, head, sort,
- Filters can be chained together



Pipes and filters

Pipe command - |

- For chaining filter commands

```
command1 | command2
```

- Output of command 1 is input of command 2
- Pipe stands for pipeline

```
mostafa@mostafa-VirtualBox:~/Desktop$ ls | sort -r
Mostafa
hello-world.sh
EE
course
Anaconda3-2018.12-Linux-x86_64.sh
```



Shell variables

- Scope limited to shell
- Set – list all shell variables

```
mostafa@mostafa-VirtualBox:~/Desktop$ set | head -4
BASH=/usr/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:expand_aliases:extglob:extquote
:force_ignores:globasciiranges:histappend:interactive_comments:progcomp:promptva
rs:sourcpath
BASH_ALIASES=()
BASH_ARGC=([0]="0")
```



Defining shell variables

```
var_name=value
```

- No spaces around '='

```
unset var_name
```

- Deletes var_name

```
mostafa@mostafa-VirtualBox:~/Desktop$ greeting='Hello'
mostafa@mostafa-VirtualBox:~/Desktop$ echo $greeting
Hello
mostafa@mostafa-VirtualBox:~/Desktop$ SecondWord='World'
mostafa@mostafa-VirtualBox:~/Desktop$ echo $SecondWord
World
mostafa@mostafa-VirtualBox:~/Desktop$ unset SecondWord
```



Environment variable

- Extended scope

```
export var_name
```

- env - list all environment variables

```
mostafa@mostafa-VirtualBox:~/Desktop$ export greeting  
mostafa@mostafa-VirtualBox:~/Desktop$ env | grep 'gree'  
greeting=Hello
```



Metacharacters

- `#` - precedes a comment
- `;` - command separator
- `*` - filename expansion wildcard
- `?` - single character wildcard in filename expansion

```
mostafa@mostafa-VirtualBox:~$ # Hello
mostafa@mostafa-VirtualBox:~$ echo 'Hello';whoami
Hello
mostafa
mostafa@mostafa-VirtualBox:~$ ls /bin/ba*
/bin/baobab  /bin/base64  /bin/basenc  /bin/bashbug
/bin/base32  /bin/basename /bin/bash
mostafa@mostafa-VirtualBox:~$ ls /bin/?ash
/bin/bash  /bin/dash
```



Quoting

- \ - escape special character interpretation
- " " - interpret literally, but evaluate metacharacters
- ' ' - interpret literally

```
mostafa@mostafa-VirtualBox:~$ echo "\$1 each"
$1 each
mostafa@mostafa-VirtualBox:~$ echo "$1 each"
each
mostafa@mostafa-VirtualBox:~$ echo '$1 each'
$1 each
```



I/O redirection

- Input/Output, or I/O redirection, refers to a set of features used for redirecting
 - > - Redirect output to file
 - >> - Append output to file
 - 2> - Redirect standard error to file
 - 2>> - Append standard error to file
 - < - Redirect file contents to standard input

```
mostafa@mostafa-VirtualBox:~/Desktop$ echo 'Hello1' > eg.txt
mostafa@mostafa-VirtualBox:~/Desktop$ echo 'Hello1' >> eg.txt
mostafa@mostafa-VirtualBox:~/Desktop$ Hello
Command 'Hello' not found, did you mean:
  command 'hello' from snap hello (2.10)
  command 'hello' from deb hello (2.10-2ubuntu4)
  command 'hello' from deb hello-traditional (2.10-5)
  command 'jello' from deb jello (1.5.2-1)
See 'snap info <snapname>' for additional versions.
mostafa@mostafa-VirtualBox:~/Desktop$ Hello 2> eg.txt
```

```
mostafa@mostafa-VirtualBox:~/Desktop$ grep 'Hello' < eg.txt
Command 'Hello' not found, did you mean:
Command 'Hello' not found, did you mean:
```



Command substitution

- Replace command with its output
\$ (command) or 'command'
- Store output of 'pwd' command in 'here':

```
mostafa@mostafa-VirtualBox:~/Desktop$ here=$(pwd)
mostafa@mostafa-VirtualBox:~/Desktop$ echo $here
/home/mostafa/Desktop
```



Command line arguments

- Program arguments specified on the command line
- A way to pass arguments to a shell script
- Usage:

```
./MyBashScript.sh arg1 arg2
```



Batch vs. concurrent modes

- Batch mode:
Commands run sequentially

```
command1; command2
```

- Concurrent mode:
Commands run in parallel

```
command1 & command2
```



Thank
you