Preparation for the Labrary 4

The enviranmental variable PATH.
First Bash Scripts.

- Standard commands using by PATH.
- Chmod command
- Let’s write our first script in bash!
- Variables in bash
- Command substitution
- Arithmetics in Bash

1. Linux Fundamental directories

~ = Your home directory

/ = Root directory (Root is all powerful. Root access is restricted to system administrator)

/bin = binaries and other executable programms

/etc = system configuration files

/home = home directories

/usr = user related programs

/var = variable data

/usr/local = User applications can be installed in usr/local or /apt

2. The environmental variable PATH

The environmental variable PATH controls the command search path.

> **echo $PATH** = shows list of directories in which there are standard programs.

When we use, for example, the command **cat,** the system is looking for the program named cat in all the directories, defined in PATH.

Linux Commands

1. **which** file_name = find the location of the file_name

Example:

```
> which cat
/usr/bin/cat
```

Actually, when we write

```
> cat foo.dat
```

the program named cat which is located in the path /usr/bin/cat is running. Other way to run it is simply to write the full path of the running program:

```
>/usr/bin/cat foo.dat
```

2.chmod – change permissions of the files

EXAMPLE

```
> ls -l foo
drwxrwxrwx    1 root root 505 Nov 04 22:30  foo
```

d = type (**d** for directories and **–** for files)

d	**rwx**	**rwx**	**rwx**
(Type)	(User permissions)	(Group permissions)	(Others permissions)

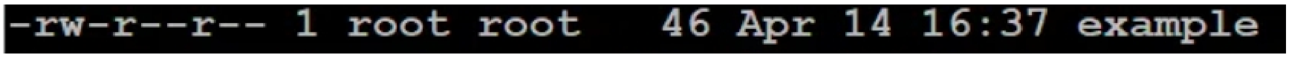
r = permission to read the file

w = permission to write to the file

x = permission to execute the file

- = no permission to read/write/execute

EXAMPLE



```
-rw-r--r-- 1 root root    46 Apr 14 16:37 example
```

File,

User: can read and write

Group: can read only

Others: can read only

----

* ugoa - user/group/other/all

*+-= - Add/Sub/Equal

* rwx - read/write/execute

EXAMPLES

>chmod g+rwx example   (**Add** the permissions for the group to be rwx)

>chmod g-r example   (Remove the permission to read for the group)

>chmod u+rwx **,** g-w example (Add rwx to user and remove w for group)

>chmod **a=r** example (Permission only to read for **all**)

Numerically based permissions

	r	w	x	
	0	0	0	Value for OFF
	1	1	1	Binary ON
	4	2	1	ON Based on 10

Octal	Binary	String
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

EXAMPLES

>chmod +700 example (Means -rwx------)

>chmod +755 example (Means -rwxr-xr-x)

ShortCuts

1. Reusing the previous commands.
To reuse the command which was used before, press few times the button ↑.

2. Completion of commands using arrows.
Write the command and then the name of the file and then press.

Tab ⇥

The filename will be completed, if possible.

Bash

1. Each file is in the format name**.sh**

2. The file strats with **#!/bin/bash.**
A shebang (#!) is a special line at the beginning of a script that tells the operating system which interpreter to use when executing the script. This line is the first line of a dash and starts with "#!" followed by the path to the interpreter.

3. Variables in bash can be defined by **VAR=”val”**,where val is a <u>string.</u>
Attention: No spaces here!! The name cannot start with a number!!
<u>Examples:</u>

```
MY_VAR=”I write Bash”
MY1Var = “2”
my123Var=”2+3”
```

<u>Invalid name</u>: 123myVAR=”2+3”

4. Command substitution can be done using
$(command) or $**(**command1| command2|command3**)** or `command` or `command1| command2|command3`, where the button ` can be found above the Tab button.

5. Arithmetics can be done using $**((**name1 op name 2**)).**
Four op are possible *in integers*: +,-,*,/.

6. To read input from the STDIN, use the command
read -p “Text_asking_input” NAME_OF_VAR

<u>Example 1 of file MyFirstBash.sh</u>

```
#!/bin/bash
VAR1=”boo likes docu”
VAR2=”2”
VAR3=”3”
VAR4=”drama”

echo “VAR1=$VAR1”
echo “VAR2=$VAR2, VAR3=$VAR3”
echo “and together it means $VAR1$VAR4”
echo “$VAR2+$VAR3 = $(($VAR2+$VAR3))”
```

<u>Running:</u>

```
>chmod 777 MyFirstBash.sh
> ~/Lab4/MyFirstBash.sh
```

Example 2 of file MySecondBash.sh

```
#!/bin/bash
echo “$(cat MyFirstBash.sh | tail+3 |head -2)”
echo “my present working directory is `pwd`”
echo “the directory $(pwd) contains the following files: $(ls)”
```

Running:

```
>chmod +755 MySecondBash.sh
>  ./MySecondBash
```

Example 3 of file ReadInput.sh

```
#!/bin/bash
read -p "Enter your input:" INPUT_VAR
echo "your input values = $INPUT_VAR"
read -p "Enter 3  input values:" VAR1 VAR2 VAR3
echo "your input values = $VAR1 $VAR2 $VAR3"
echo “the sum of these values is $(($VAR1+$VAR2+$VAR3))”
```