

# Lab 01: Scripting the Shell & Filtering Text

## Readings

The readings for Lab 05 are:

1. [Shell Scripting Tutorial](#)
  - Read from [1. Introduction](#) through [13. Functions](#) and then about [exit codes](#) and [trap](#).

*You will need this information to complete this reading assignment.*

2. [The Linux Command Line](#):
  - Chapter 16 - Networking
  - Chapter 17 - Searching For Files
  - Chapter 18 - Archiving And Backup
3. [The Linux Command Line](#):
  - Chapter 19 - Regular Expressions
  - Chapter 20 - Text Processing
4. [RegexOne](#)

**Optional** Resources:

- [The Linux Command Line](#) (Part 4: Writing Shell Scripts)
- [Bash Guide](#)
- [Shell programming with bash: by example, by counter-example](#)
- [Introduction to Linux - Chapter 10. Networking](#)
- [Slackware Linux Essentials - Chapter 13 Basic Network Commands](#)
- [Regular Expressions - User Guide](#)
- [Using Grep & Regular Expressions to Search for Text Patterns in Linux](#)
- [The Basics of Using the Sed Stream Editor to Manipulate Text in Linux](#)
- [Advanced Bash Scripting - Text Processing Commands](#)
- [The Unix School - Awk & Sed](#)
- [Sed - An Introduction and Tutorial](#)
- [Grep - An introduction to grep and egrep](#)
- [USEFUL ONE-LINE SCRIPTS FOR SED](#)

## Part 01: Scripting the Shell

TL;DR

The focus of this Lab 05 - part 01 is to introduce [shell scripting](#) in [bash](#) and some basic [networking](#).

## Questions

Given the following output of `ls -l`:

```
total 8.0K
-rw-r--r-- 1 pbui pbui 23 Jan 18 15:39 README.md
-rw-r--r-- 1 pbui pbui 155 Jan 25 01:15 exists.sh
```

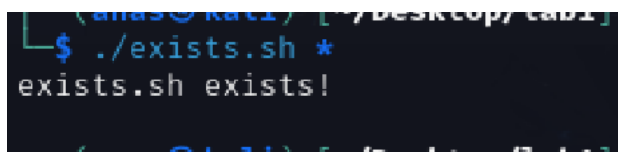
And the following script, `exists.sh`:

```
#!/bin/sh

if test -e "$1"; then
    echo "$1 exists!"
else
    echo "$1 does not exist!"
fi
```

In your `Lab01.docx` file, answer the following questions:

1. How would you run the script even though it is not executable? `sh + filename` or `bash+filename`
2. How would you make this script executable? Using `chmod +x exists.sh`
3. Once this script is executable, how would you run it directly? `./exists.sh`
4. What is the purpose of the line `#!/bin/sh`? tell the operating system to start execute the script using the `#!/bin/sh` shell.
5. What is the output of the script if you run it with the arguments `*`? Check every file in the directory if exists.



```
(anas@kali) [~/Desktop/lab1]
$ ./exists.sh *
exists.sh exists!
(anas@kali) [~/Desktop/lab1]
```

6= first argument in the script example : `./exists.sh`. (file name) this is `$1`

6. What is the `$1` that appears in the script?
7. What does `test -e "$1"` do?

Check if the the

arguments 1 exist

8. What does this script do?

The script checks if the entered file (arguments) exists in the directory or not.

Write a new version of `exists.sh` with the following modifications:

1. Use `[]` instead of `test` for the conditional.

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

```
if [ -e "$1" ]; then
```

```
    echo "$1 exists!"
```

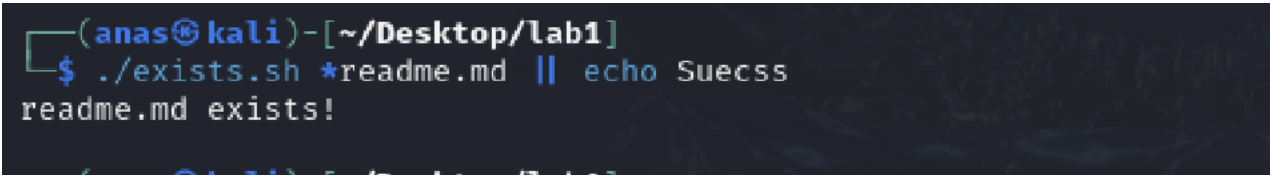
```
else
```

```
    echo "$1 does not exist!"
```

```
fi
```

```
if [ -e "$1" ]; then
```


2. This will check first argument (file name) if it exists will return `name_file + exists!`



```
(anas@kali)-[~/Desktop/lab1]
$ ./exists.sh *readme.md || echo Success
readme.md exists!
```

3. Return an error code if one of the tests fails.

In this case the file does not exist in the directory



```
(anas@kali)-[~/Desktop/lab1]
$ ./exists.sh ASDF.txt
ASDF.txt does not exist!
```

4. Display an error message and exit with an error if no arguments are given.

This part of bash `[ -z "$1" ]` check if first argument is empty

```
#!/bin/sh
if [ -z "$1" ]; then
    echo "Error No file argument provided." >&2
    exit 1
fi

(anas@kali)-[~/Desktop/lab1]
$ ./exists.sh
Error No file argument provided.
```

## Testing

To verify the correctness of your `exists.sh` script, you should be able to reproduce the following:

```
$ ls -l                                # List files in reading02 directory

total 8.0K

-rw-r--r-- 1 pbui pbui 23 Jan 18 15:39 README.md

-rwxr-xr-x 1 pbui pbui 254 Jan 28 18:02 exists.sh

$ ./exists.sh * && echo Success        # Run script and check error code

exists.sh exists!

README.md exists!

Success

$ ./exists.sh * ASDF || echo Success  # Run script and check error code

exists.sh exists!

README.md exists!

ASDF does not exist!

Success
```

## Part 02: Filtering Text

## TL;DR

The focus of this Lab 05 – part 02 is to introduce **regular expressions** and revisit **filters** and **pipelines**.

## Questions

In your **Lab01.docx** file, describe what command(s) you would use to accomplish the following:

1. Convert all the input text to upper case:

```
(anas@kali)-[~/Desktop/lab1]
$ echo "ALL your base are belong to us" | tr '[:lower:]' '[:upper:]'
ALL YOUR BASE ARE BELONG TO US
```

- 2-

```
(anas@kali)-[~/Desktop/lab1]
$ echo "monkeys love bananas" | sed 's/monkeys/gorillaz/'
gorillaz love bananas
```

- 3-

```
(anas@kali)-[~/Desktop/lab1]
$ echo "    monkeys love bananas" | sed 's/^[ \t]*//'
monkeys love bananas
```

- 4-

```
(anas@kali)-[~/Desktop/lab1]
$ cat /etc/passwd | grep "^root:" | cut -d':' -f7
/usr/bin/zsh
```

- 5-

```
(anas@kali)-[~/Desktop/lab1]
$ sed -E 's#/#bin/(bash|csh|tcsh)#/usr/bin/python#g' /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

- 6-

```
(anas@kali)-[~/Desktop/lab1]
$ cat /etc/passwd | grep ':[4][0-9]*7:'

(anas@kali)-[~/Desktop/lab1]
```

```
$ echo "All your base are belong to us" | ...
```

```
ALL YOUR BASE ARE BELONG TO US
```

2. Find and replace all instances of `monkeys` to `gorillaz`:

```
$ echo "monkeys love bananas" | ...
```

```
gorillaz love bananas
```

3. Remove any leading whitespace from a string of text:

```
$ echo "  monkeys love bananas" | ...
```

```
monkeys love bananas
```

4. Parse the `/etc/passwd` file for the shell of the root user:

```
$ cat /etc/passwd | ...
```

```
/bin/bash
```

**Hint:** You may need to read up on the format of `/etc/passwd`

5. Find and replace all instances of `/bin/bash`, `/bin/csh`, and `/bin/tcsh` to `/usr/bin/python` in `/etc/passwd`:

```
$ cat /etc/passwd | ... | grep python
```

```
root:x:0:0:root:/root:/usr/bin/python
```

```
mysql:x:27:27:MySQL Server:/var/lib/mysql:/usr/bin/python
```

```
xguest:x:500:501:Guest:/home/xguest:/usr/bin/python
```

```
condor:x:108172:40:Condor Batch System:/afs/nd.edu/user37/condor:/usr/bin/python
```

```
lukew:x:522:40:Luke Westby temp access:/var/tmp/lukew:/usr/bin/python
```

6. Find all the records in `/etc/passwd` that have a number that begins with a `4` and ends with a `7`:

```
$ cat /etc/passwd | ...
```

```
rtkit:x:499:497:RealtimeKit:/proc:/sbin/nologin
```

qpidd:x:497:495:Owner of Qpidd Daemons:/var/lib/qpidd:/sbin/nologin

uuid:x:495:487:UUID generator helper daemon:/var/lib/libuuid:/sbin/nologin

mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin

7. Given two text files, show all the lines that are present in both files.

```
(anas@kali)-[~/Desktop/lab1]
$ comm -12 <(sort file1.txt) <(sort file2.txt)
hi
```

8. Given two text files, show which lines are different.

```
(anas@kali)-[~/Desktop/lab1]
$ diff file1.txt file2.txt
1c1
< this is file 1
—
> this is file 2
```

## Submission

To submit your assignment, please upload your work to the **Lab01** folder in your **assignments Blackboard** repository. Your **Lab01** folder should only contain the following files:

- **Lab01.pdf**
- **exist.sh**