Lab 9
Date, whoami commands.
Environmental variables.
Background commands.
Processes commands: ps and kill.
Exit codes.
Recursion.

Read carefully the instructions before performing!!!
Submission in pairs : 27/01/25 for the groups of Monday
30/01/25 for the groups of Thursday

Linux Commands

1. date = shows the current date

2. whoami = displays the username of the current user

3. hostname = displays the hostname of the current user

4. Environmental Variables

- **printenv** = show the list of environmental variables.

- **Examples**: PWD,OLDPWD,PATH, LOGNAME,HOME,...

- **Create env. Var.: > export VARNAME=”VarValue” (No spaces here)**

- Delete env. Var.: >unset VARNAME

Example:

> export HOME=”/home”

> printenv HOME (or echo $HOME)

> export MYH=”/home/tamar” *(Create MYH)*

> unset MYH *(Delete MYH)*

5. ps = shows the processes running on your Linux system.

ps -a = List all ruining processes for all users.

6. **kill -9** PID = kill the process PID

7. command & = Background process= a process that is started from a terminal session and then runs independently.

Example

>cat while.sh

#!/bin/bash

```
i=1
while [ $i -gt 0 ]
do
        let i++
done
```

>ps

```
  PID TTY          TIME         CMD
2682280 pts/13   00:00:00 bash
2758036 pts/13   00:00:00 ps
```

> ./while.sh &

>ps

```
  PID TTY          TIME         CMD
2682280 pts/13   00:00:00 bash
```

2758122 pts/13   00:00:04 **while.sh**

```
2758261 pts/13   00:00:00 ps
```

>**kill -9 2758122**

>ps

```
 PID TTY          TIME         CMD
2682280 pts/13   00:00:00 bash
2758036 pts/13   00:00:00 ps
```

8. exit = the exit status is an integer number. A command which exits with a zero (0) exit status has succeeded. A non-zero (1-255) exit status indicates failure.

The special shell variable **$?** in bash is used to get the exit status of the last command.

Example

> ping -c 1 “google.com”

>echo $?

0

> ping -c 1 “google234.com”

>echo $?

2

Example exitFile.sh

```
#!/bin/bash
HOST="google.com"
ping -c 1 $HOST
if [ "$?" -eq "0" ]
then
        echo "$HOST reachable"
else
        echo "$HOST unreachable"
fi
```

--

List of the exit codes:

Exit code	Description
0	Successful exit without errors
1	One or more generic errors encountered upon exit
2	Incorrect usage, such as invalid options or missing arguments
126	Command found but is not executable
127	Command not found, usually the result of a missing directory in $PATH variable
128+N	Command encountered fatal error (was forcefully terminated manually or from an outside source). The `N` tells us which signal was received (see example below)
130	Command terminated with signal 2 (SIGINT) (ctrl+c on keyboard). 128+2
143	Command terminated with signal 15 (SIGTERM) (kill command). 128+15

Exit command in Bash

We use exit to exit from a Bash script. We can call exit anywhere in the script, either inside a function or outside a function.This stops the execution of the script immediately. It may take an integer argument. In that case, the script exits with the exit status set to the integer argument. If we don't pass an argument to exit, it exits with the exit status set to the exit status of the last command executed before exit.

Example exitFile2.sh

!/bin/bash

##############################

#exit outside of function

echo "This line will be printed"

exit 0

```
#exit 126
echo “But this line will not be printed”
############################
func(){
        echo "my name is $1"
        exit 12
}
func AVIV
echo "The function is over, but we are still here"
```

Recursion in Bash

Example recursion.sh:

```
#!/bin/bash
recurse()
{
   if [ $1 -lt 5 ]
   then
      echo $1
      recurse $(($1 + 1))
   fi
}
recurse $1
```

Procedures

1.Create directory Lab9.

2. Write a script **grepN.sh** which implements the linux command **grep -n** using while command.

Note: grep command can not be used in this script.

3. Write a script **fact.sh** which contains a recursion function **fact.** The script receives a positive number **n,** calculates and prints **n!**.

4. Write a script **long.sh** which runs in the background the script fact.sh with n=20. And then runs infinite for loop.

Next, in the shell show the list of the currently running processes (send this list to the **out** file) and kill the script long.sh. Then send again the list of the currently running processes to **out** file.

5. Write a script **filesInPATH.sh** that receives string PERMITIONS and for each directory in the PATH env. var. prints a number of files which have given permissions.

For example,

> ./filesInPATH “-rwxr-xr-x” will print number of files which have -rwxr-xr-x permission in all the directories defined in PATH.

6. Write a script **details.sh** which will print the date when the program is running, following by the name of the user who runs this program and the name of the host.

Then the script will show the list of the alias and of the env. variables defined for this user.

Finally, the script will check if the last command used in the script was executed correctly (hint: use an exit code) and print an appropriate message.

Note: Run this script using > source details.sh

7. Write script **bestPlayer.sh**, using the following steps:

a). Write a function **search** that accepts a string and file names and prints all the lines appearing in these files that contain the requested string.

```
> search Blue scene35.txt
scene35.txt : 37 : LAUNCELOT: Blue.
scene35.txt : 55 : GALAHAD: Blue.  No yel--  Auuuuuuuugh!
> search swallow scene*.txt
scene1.txt : 50 : GUARD #1: But then of course African swallows are not migratory.
scene1.txt : 54 : GUARD #2: Wait a minute -- supposing two swallows carried it together?
scene35.txt : 63 : BEDEMIR: How do know so much about swallows?
> search cow farms
farms/animals/animals.txt : 8 : cow
farms/farm1.txt : 2 : cow Betsy
farms/farm1.txt : 3 : slim cow Dazy
farms/farm1.txt : 4 : fat cow Burger
farms/farm1.txt : 5 : two cows Dartsy & Teo
farms/farm2.txt : 2 : cow Leni
farms/farm2.txt : 4 : cow Oreo
```

b). Given a file called **football.txt** that contains data on goals scored in football matches.

Each line in the file indicates

- the name of the player; the number of goals he scored in the match held on a certain date; the name of the team he played for and the name of the opposing team.

For example,

football.txt

```
Alon Miz. 2 23/10/93 Macabi-Haifa Macabi-Tel-Aviv
Izak Zoh. 1 12/11/93 Macabi-Tel-Aviv Hapoel-Beer-Sheva
Ronen Ha. 3 27/12/93 Hapoel-Tel-Aviv Macabi-Tel-Aviv
Reuven A. 2 12/11/93 Macabi-Haifa Hapoel-Tel-Aviv
Eyal Ber.    1 20/11/93 Macabi-Haifa Macabi-Tel-Aviv
Izak Zoh. 1 12/11/93 Macabi-Tel-Aviv Hapoel-Haifa
Alon Miz. 2 26/10/93 Macabi-Haifa Beitar-Jerusalem
Izak Zoh. 2 12/12/93 Macabi-Tel-Aviv Macabi-Hiafa
Alon Miz. 2 23/12/93 Macabi-Haifa Macabi-Pet-Tikva
Ronen Ha. 3 27/11/93 Hapoel-Tel-Aviv Macabi-Haifa
```

Write a function called **player** which will receive as a parameter the name of a player and will print all the lines with his name from the football.txt file and the sum of the number of goals he scored.

For example, for the above file the out put will be:

```
> player "Alon Miz."
Alon Miz. 2 23/10/93 Macabi-Haifa Macabi-Tel-Aviv
Alon Miz. 2 26/10/93 Macabi-Haifa Beitar-Jerusalem
Alon Miz. 2 23/12/93 Macabi-Haifa Macabi-Pet-Tikva
Total number of goals: 6
```

c). Write a function to rank players named **best_player** which will receive a list of player names on the command line and print the name of the player who scored the most goals.

If there are several players who scored the most goals, the names of all players will be printed. Each player's name and the number of goals scored must be printed.

```
> best_player "Alon Miz." "Izak Zoh." "Ronen Ha." "Reuven A."
Alon Miz. 6
Ronen Ha. 6
```

Notes to a)-c):

- Assume that the existing input file is correct.

- Above file football.txt is given only as an example.

- Do not use temporary files in your solution!

Submission:
Create one compressed file named Linux_Lab9_id1_id2.tar,
where id1 and id2 must be changed to the id numbers of the two partners.

This file will include the files **grepN.sh, fact.sh, long.sh, out, filesInPATH.sh, details.sh, bestPlayer.sh.**

Submit only the file Linux_Lab9_id1_id2.tar in Moodle and only by one of the partners.

Appeals:

Can be submitted by e-mail to Elad mail@eladhuttner.net within a week from the date of publication of the grades.