|  |
| --- |
| Bahrain Polytechnic |
| Unix Systems |
| Lab Session 8 |
|  |
|  |
|  |

|  |
| --- |
|  |

# Lab session 8 – Scripting 2

## Introduction

The lab work consists of all practical tasks which must be submitted via Moodle.

You must submit the following:

* A single zip file, uploaded to Moodle.

**Note: Each Lab session is worth 1% of your final mark**

### **Learning Outcomes Assessed**

The following learning outcomes are being assessed in this lab session:

* Use the command-line on a UNIX system
* Manage a Linux server system (including files, processes, users)

## Lab 8 – Practical tasks

Create a script findBigFiles that will take as an **argument**:

* 1. A size in MB that is considered to be a big file.

The script will return the following information for the **current working directory** ($pwd):

1. The list of large files
2. The number of files found
3. The date and time of the search

**Breakdown of Tasks**

1. Add comments (script name, author, date and purpose) at the start of your script as appropriate
2. Create a function **usage** that will print out the correct usage as shown below

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USAGE: $SCRIPT\_NAME [Number\_Of\_Meg\_Bytes]

EXAMPLE: $SCRIPT\_NAME 5

Will Find Files Larger Than 5 Mb in and below the Current Directory...

EXITING...

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function usage

{

echo “\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”

echo “USAGE: $SCRIPT\_NAME [Number\_Of\_Meg\_Bytes]”

echo “EXAMPLE: $SCRIPT\_NAME 5”

echo “Will Find Files Larger Than 5 Mb in and below the Current Directory...”

echo “EXITING...”

echo “\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”

}

1. In the main part of your script, test the number of arguments so that if the script does not have 1 argument, it prints the usage and exits.

**Hint:**

To exit the script (i.e. if the number of arguments is incorrect), use the command *exit 1*  
if[$# -ne 1]

then

usage

exit 1

fi

1. In order to allow the user to exit using CTRL+C, we need to add the following code.

In the FUNCTION section:

function trap\_exit

{

echo -e “\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”

echo -e “\n\n EXITING ON A TRAPPED SIGNAL...”

echo -e “\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n”

}

function trap\_exit

{

echo -e “\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”

echo -e “\n\n EXITING ON A TRAPPED SIGNAL...”

echo -e “\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n”

}

At the start of the MAIN section:

# Set a trap to exit. REMEMBER - CANNOT TRAP ON kill -9

trap ‘trap\_exit; exit 2’ 1 2 3 15

#before the if statement add this (Q3)

# Set a trap to exit. REMEMBER - CANNOT TRAP ON kill -9

trap ‘trap\_exit; exit 2’ 1 2 3 15

**Note** the use the command *exit 2* to indicate a forced exit

1. Define the following variables and assign to them the appropriate value

**DATESTAMP** is the current date/time in the following format **“%h %d,%Y,%T”**

**SEARCH\_PATH** is the current directory (Top-level directory to search)

#DATASTAMP = $(date -u + ”format”)

DATASTAMP = $(date -u + ”%h%d,%Y,%T”)

SEARCH\_PATH = $(pwd)

1. Display the following message on screen

Searching for Files Larger Than xMb starting in:

/home/A2009XXXX

Please Standby for the Search Results...

Be sure to use the correct variables and parameters.

echo “Searching for Files Larger Than $1 Mb starting in:”

echo “$SEARCH\_PATH”

echo “Please Standby for the Search Results...”

1. **Searching for files**

Use the find command to find all the files greater than xMB (the size of MB is passed from an argument) in your current directory.

**Hint:** in the man find out about –type, -size and –print

Redirect the output to a file (For example: results.out)

find -H $SEARCH\_PATH -type f -size +“$1”c > temp1.txt

find 🡪 command

-type 🡪 what is it?

F 🡪 file type

-size 🡪 choose size

$1🡪 comes from the user

C 🡪 bytes

1. Test the size of the file to find out if there is anything inside of it

**Hint:** -s checks the size of a file

If the file is empty, display the following info on screen:

Nofiles were found that are larger than xMB

Exiting...

if[ -s temp.txt]

then

echo “Number of files found: $num\_lines”

else

echo “Nofiles were found that are larger than $1 MB”

echo “Exiting...”

If the file is not empty, then the file contains one line per file.

* Count the number of lines in the file and display as follows:

Number of files found: 17

num\_lines = $(cat temp1.txt | wc -l)

then call it (Q8)

1. The output should display as follows:

Searching for Files Larger Than 1Mb Starting in:

/home/david/unit\_8

Please Standby for the Search Results...

Date and time of Search: Dec12,2011,15:32:58

Number of Files Found: 3

These files were found:

/home/david/unit\_8/CPU.JPG

/home/david/unit\_8/findbigfiles

/home/david/unit\_8/findbigfiles\_new

Echo “Date and time of Search: $DATESTAMP”

echo “Date and time of Search: $DATASTAMP”

if statement

number of lines

echo number of files

echo files found

cat temp1.txt | more

rm temp1.txt

1. Zip the script and upload it to Moodle

Everything in one script

//////////////////////////////////////////////////

Funcyion usage

Function type\_exit

Datastamp

Seach\_path

File\_size

If statement 🡪for usage

Echos

Find

If statement 🡪 dose the file exist

Echos

cat

Else 🡪 found nothing

Delete the file

////////////////////////////////////////////

## Lab8 – What you need to submit

Name the tar file *Lab8.tar.gz* and submit this to Moodle by the end of the week.