



Εργαστήριο 6

Ασκήσεις: Προγραμματισμός Κελύφους Bash

- 1) Γράψετε ένα πρόγραμμα στο κέλυφος Bash το οποίο ζητά από το χρήστη για το όνομα του αρχείου ή ενός καταλόγου και τυπώνει αν είναι κανονικό αρχείο ή κατάλογος. Στο τέλος εκτελέστε μια εντολή ls με το long listing option πάνω στο δοθέν (από το χρήστη) όνομα.

Παράδειγμα εκτέλεσης

```
b103ws10:/home/faculty/csp5pa1>./ask1.sh
Enter the file path: ./EPL421
./EPL421 is a directory
total 1024
drwxr-xr-x  14 csp5pa1 tspecial    4096 Sep 22 15:52 AS1
drwxr-xr-x 109 csp5pa1 tspecial    8192 Nov 21 15:53 AS2
-rw-r--r--   1 csp5pa1 tspecial 681970 Dec  4 18:23
EPL131.zip
```

SOLUTION

```
#!/bin/bash
read -p "Enter the file path: " FILE
if [ -f "$FILE" ]; then
    echo "$FILE is a regular file"
elif [ -d "$FILE" ]; then
    echo "$FILE is a directory"
else
    echo "$FILE is another type of file"
fi
ls -l $FILE
```

- 2) Τροποποιήστε το προηγούμενο script έτσι ώστε να δέχεται το όνομα αρχείου ή καταλόγου σαν όρισμα γραμμής εντολής (command line argument) αντί να το ζητά από το χρήστη.

Παράδειγμα εκτέλεσης

```
b103ws10:/home/faculty/csp5pa1>./ask1.sh ./EPL421
./EPL421 is a directory
total 1024
drwxr-xr-x  14 csp5pa1 tspecial    4096 Sep 22 15:52 AS1
drwxr-xr-x 109 csp5pa1 tspecial    8192 Nov 21 15:53 AS2
-rw-r--r--   1 csp5pa1 tspecial 681970 Dec  4 18:23
EPL131.zip
```

SOLUTION

```
#!/bin/bash
FILE=$1
if [ -f "$FILE" ]; then
    echo "$FILE is a regular file"
elif [ -d "$FILE" ]; then
```



```

    echo "$FILE is a directory"
else
    echo "$FILE is another type of file"
fi
ls -l $FILE

```

- 3) Τροποποιήστε το προηγούμενο script έτσι ώστε να δέχεται απεριόριστο αριθμό ορισμάτων γραμμής εντολής (unlimited command line argument).

SOLUTION

```

#!/bin/bash
FILES=$*
for FILE in $FILES
do
    if [ -f "$FILE" ]; then
        echo "$FILE is a regular file"
    elif [ -d "$FILE" ]; then
        echo "$FILE is a directory"
    else
        echo "$FILE is another type of file"
    fi
    ls -l $FILE
done

```

- 4) Γράψτε ένα πρόγραμμα στο κέλυφος Bash που υλοποιεί και καλεί μια συνάρτηση που βρίσκει και αναθέτει σε τοπική μεταβλητή τον αριθμό των αρχείων και καταλόγων που βρίσκονται στον τρέχοντα κατάλογο και στο τέλος τυπώνει τη τιμή της μεταβλητής.

SOLUTION

```

#!/bin/bash

function file_count() {
    NUMBER_OF_FILES=$(ls -l | wc -l)
    # OR
    # NUMBER_OF_FILES=`ls -l | wc -l`
    echo "$NUMBER_OF_FILES"
}

file_count

```

- 5) Τροποποιήστε το προηγούμενο script έτσι ώστε να τυπώνει το μήνυμα “Small number” αν ο αριθμός ≤ 20 , “Medium number” αν είναι μεταξύ 21 και 30 και “High number” αν είναι μεγαλύτερος 30.

SOLUTION

```

#!/bin/bash

function file_count() {
    NUMBER_OF_FILES=$(ls -l | wc -l)

```



```
if [ $NUMBER_OF_FILES -le 20 ]; then
    echo "Small number"
elif [ $NUMBER_OF_FILES -le 30 ]; then
    echo "Medium number"
else
    echo "High number"
fi
}

file_count
```

- 6) Γράψτε ένα πρόγραμμα στο κέλυφος Bash που ψάχνει σε όλα τα αρχεία .c του τρέχοντος καταλόγου, βρίσκει και τυπώνει αυτό που έχει το πιο μεγάλο αριθμό γραμμών.

Παράδειγμα εκτέλεσης

```
b103ws10:/home/faculty/csp5pa1>./ask6.sh
server.c has 159 lines
```

SOLUTION

```
#!/bin/bash

max=0

for file in `ls *.c`
do
    num=`cat $file | wc -l`
    if [ $num -gt $max ]; then
        max=$num
        filename=$file
    fi
done

echo "$filename has $max lines"
```