



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

# Κυριάκος Καρατζιάς

1<sup>η</sup> Εργασία στο μάθημα **Λειτουργικά Συστήματα**

Ταύρος, 10 Δεκεμβρίου 2023

Περιεχόμενα	
<b>Άσκηση 1</b>	<b>3</b>
Κώδικας	3
Ενδεικτικές εκτελέσεις (screenshots):	4
Γενικά Σχόλια/Παρατηρήσεις	4
Με δυσκόλεψε / δεν υλοποίησα	4
<b>Άσκηση 2</b>	<b>5</b>
Κώδικας	5
Ενδεικτικές εκτελέσεις (screenshots):	6
Γενικά Σχόλια/Παρατηρήσεις	6
Με δυσκόλεψε / δεν υλοποίησα	6
<b>Άσκηση 3</b>	<b>7</b>
Κώδικας	7
Ενδεικτικές εκτελέσεις (screenshots):	8
Γενικά Σχόλια/Παρατηρήσεις	8
Με δυσκόλεψε / δεν υλοποίησα	8
<b>Άσκηση 4</b>	<b>9</b>
Κώδικας	9
Ενδεικτικές εκτελέσεις (screenshots):	10
Γενικά Σχόλια/Παρατηρήσεις	10
Με δυσκόλεψε / δεν υλοποίησα	10
<b>Άσκηση 5</b>	<b>11</b>
Κώδικας	11
Ενδεικτικές εκτελέσεις (screenshots):	12
Γενικά Σχόλια/Παρατηρήσεις	12
Με δυσκόλεψε / δεν υλοποίησα	12
<b>Συνοπτικός Πίνακας</b>	<b>13</b>

# Άσκηση 1

## Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

if [ $# -ne 1 ]; then # Check for the right number of arguments
    echo "Please give the correct number of arguments!"
    echo "'e' for easy"
    echo "'m' for medium"
    echo "'h' for hard"
    exit 1
fi

case "$1" in
    e)      #if argument is e
        password=$(tr -dc '[:alnum:]' < /dev/urandom | head -c 8)
        ;;
    m)      #if argument is m
        password=$(tr -dc '[:alnum:].!#' < /dev/urandom | head -c 12)
        ;;
    h)      #if argument is h
        password=$(tr -dc '[:alnum:].!#@,?^$' < /dev/urandom | head -c 20)
        ;;
    *)      #if argument is something else
        echo "Wrong input!"
        exit 1
        ;;
esac

echo "$password is the generated password"
```

## Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
karatzias@~$ cd ~/code/semester_3/oper_sys/scripts/anafora1 && shellcheck it2022120_ex_1.sh
karatzias@~$ cd ~/code/semester_3/oper_sys/scripts/anafora1 && ./it2022120_ex_1.sh e
n3xTJj35 is the generated password
karatzias@~$ cd ~/code/semester_3/oper_sys/scripts/anafora1 && ./it2022120_ex_1.sh m
7vS9Ub.#gz0C is the generated password
karatzias@~$ cd ~/code/semester_3/oper_sys/scripts/anafora1 && ./it2022120_ex_1.sh h
W!SWlKXaa9PioZs0.rc. is the generated password
```

Παρατηρήσεις:

- Έλεγχος παραμέτρων χρήστη

```
karatzias@~$ cd ~/code/semester_3/oper_sys/scripts/anafora1 && ./it2022120_ex_1.sh wrongArgument
Wrong input!
```

Παρατηρήσεις:

## Γενικά Σχόλια/Παρατηρήσεις

Αρχικά ελέγχουμε αν ο χρήστης έδωσε σωστό argument με την χρήση ενός if. Μετά δημιουργούμε ένα switch το οποίο πέρνει τις 3(+1) περιπτώσεις. Η εντολή tr -dc διαγράφει όλους τους χαρακτήρες από το αρχείο που δίνουμε με ανακατεύθυνση εισόδου (dev/urandom/) οι οποίοι δεν περιέχονται στο σύνολο που της δίνουμε. Στην περίπτωση μας δίνουμε τους χαρακτήρες που περιέχονται στο [:alnum] (οι οποίοι είναι αλφαριθμητικοί χαρακτήρες) και ανάλογα από το argument που δίνουμε προσθέτουμε τους ανάλογους χαρακτήρες που θέλουμε. Επίσης με την χρήση της εντολής head πέρνουμε τον ανάλογο αριθμό χαρακτήρων. Τέλος εμφανίζουμε τον κωδικό με μήνυμα.

## Links που χρησιμοποίησα

<https://www.geeksforgeeks.org/tr-command-in-unix-linux-with-examples/>

# Άσκηση 2

## Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

while true; do

echo "Menu:"echo "1) Hostname"

echo "2) Kernel"

echo "3) CPU"

echo "4) Memory"

echo "5) Disk-Usage"

echo "6) Exit"

read -r -p "Choose a number between 1 and 6: " choice

case "$choice" in

1)

echo "Hostname: $(hostname)"

;;

2)

echo "Kernel: $(uname -r)"

;;

3)

echo "CPU: $(lscpu | grep "Model name" | awk -F: '{print $2}' | sed
's/^[\t]*//') "

;;

4)
```

```
echo "Memory: $(free -h | grep "Mem:" | awk '{printf "Total: %s Used: %s Free: %s\n", $2, $3, $4}')
```

```
;;
```

```
5)
```

```
echo "Disk Usage: $(df -h | grep "/dev/sda4" | awk '{printf "Total: %s Used: %s Free: %s\n", $2, $3, $4}')
```

```
;;
```

```
6)
```

```
echo "Exit"
```

```
break
```

```
;;
```

```
*)
```

```
echo "Wrong input!"
```

```
;;
```

```
esac
```

```
echo "====="
```

```
done
```

## Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 shellcheck it2022120_ex_2.sh
karatziask$~/code/semester_3/oper_sys/scripts/anafora1
```

```
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 ./it2022120_ex_2.sh
Menu:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
6) Exit
Choose a number between 1 and 6: 1
Hostname: karatziask
=====
Menu:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
6) Exit
Choose a number between 1 and 6: 2
Kernel: 6.2.0-39-generic
=====
Menu:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
6) Exit
Choose a number between 1 and 6: 3
CPU: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
=====
Menu:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
6) Exit
Choose a number between 1 and 6: 4
Memory: Total: 7,7Gi Used: 2,2Gi Free: 2,6Gi
=====
Menu:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
6) Exit
Choose a number between 1 and 6: 5
Disk Usage: Total: 191G Used: 36G Free: 146G
```

## Παρατηρήσεις:

- Έλεγχος παραμέτρων χρήστη

```
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 ./it2022120_ex_2.sh
Menu:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
6) Exit
Choose a number between 1 and 6: wrongInput
Wrong input!
```

Παρατηρήσεις:

## Γενικά Σχόλια/Παρατηρήσεις

Αρχικά το πρόγραμμα εμφανίζει το menu μέχρι ο χρήστης να επιλέξει μια σωστή επιλογή. Σε περίπτωση λάθους εμφανίζει ανάλογο μήνυμα. Η εντολή hostname εμφανίζει το hostname. Το uname -r εμφανίζει το kernel version. Η lscpu εμφανίζει πληροφορίες για την cpu. Χρησιμοποιούμε την grep για να πάρουμε την γραμμή που αφορά αυτό που θέλουμε, την awk για να πάρουμε την 2η γραμμή και την sed για να φιλτράρουμε την έξοδο. Η εντολή free εμφανίζει πληροφορίες σχετικά με την μνήμη του συστήματος. Χρησιμοποιούμε την grep για να πάρουμε την γραμμή που θέλουμε και την awk για να διαμορφώσουμε την έξοδο. Η df εμφανίζει πληροφορίες για την χρήση του δίσκου. Το -h το προσθέτουμε για να εμφανίσει human readable. Grep για να πάρουμε την γραμμή που θέλουμε και awk για την διαμόρφωση της εκτύπωσης.

## Links που χρησιμοποίησα

<https://tech.rochester.edu/tutorials/finding-the-computer-name-on-linux/>

<https://www.geeksforgeeks.org/df-command-linux-examples/>



# Άσκηση 3

## Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

if [ -z "$1" ]; then
echo " $0 [path-to-log-file]"
exit 1
fi

file="$1"

awk '{count[$1 " "$2]++} END {for (word in count) print count[word],
word}' "$file"
```

## Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 shellcheck it2022120_ex_3.sh
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 ./it2022120_ex_3.sh /home/karatziask/code/semester_3/oper_sys/scripts/anafora1/syslog
1613 Nov 28
996 Nov 30
948 Dec 1
594 Nov 29
518 Nov 26
398 Nov 27
```

Παρατηρήσεις:

- Έλεγχος παραμέτρων χρήστη

```
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 ./it2022120_ex_3.sh
./it2022120_ex_3.sh [path-to-log-file]
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 █
```

Παρατηρήσεις:

## Γενικά Σχόλια/Παρατηρήσεις

Αρχικά το πρόγραμμα ελέγχει μέσω του if αν δώθηκαν σωστά τα ορίσματα. Αποθηκεύει το argument (\$1) σε μια μεταβλητή file. Η awk δημιουργεί έναν πίνακα (count). Ο μετρητής αυξάνεται μέχρι να φτάσει στο τέλος της ημερομηνίας του αρχείου. Τέλος τα εκτυπώνει.

## Links που χρησιμοποίησα

<https://www.cyberciti.biz/tips/howto-linux-unix-write-to-syslog.html>

<https://www.geeksforgeeks.org/awk-command-unixlinux-examples/>

# Άσκηση 4

## Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

directory=$1

if [ -z "$directory" ]; then
echo "Please give a directory as argument"
echo "$0 [path-to-directory]"
exit 1
fi

echo "5 biggest files in directory $directory:"
find "$directory" -type f -exec du -h {} + | sort -rh | head -n 5 | awk
'{print $2}' | xargs -I {} basename {}

echo -e "\nFiles with more than one hard link:"
find "$directory" -type f -links +1 -exec stat --printf="%n\n" {} \; |
xargs -I {} basename {}

echo -e "\nFiles without read permission:"
find "$directory" -type f ! -readable -exec stat --printf="%n\n" {} \; |
xargs -I {} basename {}
```

## Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
karatzias@~$ cd /code/semester_3/oper_sys/scripts/anafora1 && shellcheck it2022120_ex_4.sh
karatzias@~$ cd /code/semester_3/oper_sys/scripts/anafora1 && ./it2022120_ex_4.sh /home/karatzias/pwt
5 biggest files in directory /home/karatzias/pwt:
file_without_read_permission.txt
e.txt
d.txt
c.txt
b.txt

Files with more than one hard link:
aa.txt
aaa.txt
a.txt

Files without read permission:
file_without_read_permission.txt
```

Παρατηρήσεις:

- Έλεγχος παραμέτρων χρήστη

```
karatzias@~$ cd /code/semester_3/oper_sys/scripts/anafora1 && ./it2022120_ex_4.sh
Please give a directory as argument
./it2022120_ex_4.sh [path-to-directory]
karatzias@~$ cd /code/semester_3/oper_sys/scripts/anafora1 &&
```

Παρατηρήσεις:

## Γενικά Σχόλια/Παρατηρήσεις

Αρχικά αναθέτω το path σε μια μεταβλητή, κάνω τους απαραίτητους ελέγχους σε περίπτωση που δεν δώθηκε directory σαν argument.

Η εντολή find αναζητά από το directory όλα τα files, η du υπολογίζει το μέγεθος των αρχείων και εμφανίζει τα αποτελέσματα. Sort & head ταξινομούμε και πέρνουμε τα πρώτα 5 (μεγαλύτερα). Awk & xargs χρησιμοποιούνται για την εκτύπωση.

Find "\$directory" -type f -links +1 Βρες αρχεία στον φάκελο που έχουν περισσότερους από ένα σύνδεσμο. -exec stat --printf="%n\n" {} \; Για κάθε αρχείο που βρέθηκε, εμφάνισε το όνομά του (χωρίς τη διαδρομή). | xargs -I {} basename {} Πάρε τα ονόματα των αρχείων και εμφάνισε μόνο τα ονόματα τους χωρίς τις διαδρομές.

find "\$directory" -type f ! -readable Βρες όλα τα αρχεία στον καθορισμένο φάκελο που δεν μπορούν να διαβαστούν από τον τρέχοντα χρήστη. -exec stat --printf="%n\n" {} \; Για κάθε αρχείο που βρίσκεται, εκτέλεσε την εντολή stat για να εμφανίσει πληροφορίες σχετικά με το αρχείο, εκτυπώνοντας το όνομά του με μια αλλαγή γραμμής. | xargs -I {} basename {} Πάρε τα ονόματα των αρχείων από το stat και εφαρμόστε την εντολή basename για να εμφανιστεί μόνο το όνομα του αρχείου, αφαιρώντας τη διαδρομή.

## -Links που χρησιμοποίησα

<https://www.oreilly.com/library/view/linux-pocket-guide/9780596806347/re39.html>

<https://www.geeksforgeeks.org/xargs-command-unix/>

# Άσκηση 5

## Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

if [ $# -ne 1 ]; then
echo "please give a website as argument."
echo "$0 [website]"
exit 1
fi

web=$1

expiration_date=$(echo | openssl s_client -servername "$web" -connect
"$web":443 2>/dev/null | openssl x509 -noout -dates | grep "notAfter" |
awk -F=' ' '{print $2}')

if [ -z "$expiration_date" ]; then
echo "Unable to get date of ssl expires for website $web."
exit 1
fi

expiration_epoch=$(date -d "$expiration_date" +%s)
current_epoch=$(date +%s)

if [ "$current_epoch" -ge "$expiration_epoch" ]; then
echo "SSL has expired for website $web Date: $expiration_date."
else
echo "Date of website $web ssl expires is $expiration_date."
fi
```

## Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 shellcheck it2022120_ex_5.sh
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 ./it2022120_ex_5.sh www.google.com
Date of website www.google.com ssl expires is Feb 12 08:09:46 2024 GMT.
karatziask$~/code/semester_3/oper_sys/scripts/anafora1
```

Παρατηρήσεις:

- Έλεγχος παραμέτρων χρήστη

```
karatziask$~/code/semester_3/oper_sys/scripts/anafora1 ./it2022120_ex_5.sh
please give a website as argument.
./it2022120_ex_5.sh [website]
karatziask$~/code/semester_3/oper_sys/scripts/anafora1
```

Παρατηρήσεις:

## Γενικά Σχόλια/Παρατηρήσεις

Αρχικά ελέγχει αν έχει δοθεί ακριβώς ένα όρισμα (το όνομα της ιστοσελίδας). Αν όχι τότε εμφανίζει ένα μήνυμα και τερματίζει. Αποθηκεύει το όνομα της ιστοσελίδας από το όρισμα. Στη συνέχεια, χρησιμοποιεί την εντολή openssl για να συνδεθεί στην ιστοσελίδα και να ανακτήσει την ημερομηνία λήξης του SSL. Μετατρέπει τις ημερομηνίες για να γίνει η σύγκριση. Συγκρίνει την τρέχουσα ημερομηνία με την ημερομηνία λήξης του SSL πιστοποιητικού και εμφανίζει κατάλληλο μήνυμα ανάλογα με το αν το SSL έχει λήξει ή όχι.

## Links που χρησιμοποίησα

<https://www.tutorialspoint.com/unix/unix-shell-functions.htm>

<https://www.geeksforgeeks.org/practical-uses-of-openssl-command-in-linux/>