



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

Καλαματιανός Χρήστος

1^η Εργασία στο μάθημα **Λειτουργικά Συστήματα**

Τάυρος, 22 Δεκεμβρίου 2023

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

Άσκηση 1

Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash
generate_password() {
    local length=$1
    local characters=$2
    local password
    password=$(head /dev/urandom | tr -dc "$characters" | head -c
"$length")
    echo "$password"
}
echo "Επιλέξτε τον τύπο του κωδικού:"
echo "1) Εύκολος (8 χαρακτήρες, αλφαριθμητικά αγγλικά)"
echo "2) Μέτριος (12 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#)"
echo "3) Δύσκολος (20 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#@,?^$)"
read -p "Επιλέξτε αριθμό (1, 2 ή 3): " choice
case $choice in
    1)
        password=$(generate_password 8 "A-Za-z")
        ;;
    2)
        password=$(generate_password 12 "A-Za-z. !#")
        ;;
    3)
        password=$(generate_password 20 "A-Za-z. !#@, ?^$")
        ;;
    *)
        echo "Μη έγκυρη επιλογή. Έξοδος."
        exit 1
        ;;
esac
echo "Ο τυχαίος κωδικός είναι: $password"
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_1.sh
Επιλέξτε τον τύπο του κωδικού:
1) Εύκολος (8 χαρακτήρες, αλφαριθμητικά αγγλικά)
2) Μέτριος (12 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#)
3) Δύσκολος (20 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#@,?^$)
Επιλέξτε αριθμό (1, 2 ή 3): 1
Ο τυχαίος κωδικός είναι: PcqnXXQW
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_1.sh
Επιλέξτε τον τύπο του κωδικού:
1) Εύκολος (8 χαρακτήρες, αλφαριθμητικά αγγλικά)
2) Μέτριος (12 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#)
3) Δύσκολος (20 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#@,?^$)
Επιλέξτε αριθμό (1, 2 ή 3): 2
Ο τυχαίος κωδικός είναι: XeNuWxuUkGum
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_1.sh
Επιλέξτε τον τύπο του κωδικού:
1) Εύκολος (8 χαρακτήρες, αλφαριθμητικά αγγλικά)
2) Μέτριος (12 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#)
3) Δύσκολος (20 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#@,?^$)
Επιλέξτε αριθμό (1, 2 ή 3): 3
Ο τυχαίος κωδικός είναι: MjBhdX^$^U,efp,e$hFj
paul@paul-VirtualBox:~/Desktop/1η εργασία$
```

Παρατηρήσεις:

Η πρώτη εκτέλεση εκτελεί με τον εύκολο τρόπο

Η δεύτερη εκτέλεση εκτελεί με τον μέτριο τρόπο

Η τρίτη εκτέλεση εκτελεί με τον δύσκολο τρόπο

- Έλεγχος παραμέτρων χρήστη

```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_1.sh
Επιλέξτε τον τύπο του κωδικού:
1) Εύκολος (8 χαρακτήρες, αλφαριθμητικά αγγλικά)
2) Μέτριος (12 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#)
3) Δύσκολος (20 χαρακτήρες, αλφαριθμητικά αγγλικά και .!#@,?^$)
Επιλέξτε αριθμό (1, 2 ή 3): 4
Μη έγκυρη επιλογή. Έξοδος.
paul@paul-VirtualBox:~/Desktop/1η εργασία$
```

Παρατηρήσεις:

Στην εκτέλεση αυτή δεν μας έδωσε το αποτέλεσμα που περιμέναμε, καθώς του δώσαμε λάθος είσοδο και μας ενημέρωσε που έχουμε κάνει λάθος.

Γενικά Σχόλια/Παρατηρήσεις

Αρχικά ορίζουμε την συνάρτηση **generate_password**. Η οποία δέχεται δύο παραμέτρους, το μήκος του κωδικού και τους χαρακτήρες που μπορούν να χρησιμοποιηθούν. Στην συνέχεια χρησιμοποιεί **/dev/urandom** για τη δημιουργία τυχαίων χαρακτήρων, επιλέγει τους χαρακτήρες με βάση τους καθορισμένους χαρακτήρες και το μήκος. Τέλος επιστρέφει τον τυχαίο κωδικό.

Το script θα εμφανίζει τις διαθέσιμες επιλογές στον χρήστη, θα κάνει ανάγνωση του αριθμού που αντιστοιχεί στο επίπεδο δυσκολίας.

Για την κατασκευή του τυχαίου κωδικού χρησιμοποιούμε τη συνάρτηση **generate_password** για τη δημιουργία του κωδικού.

Αποθηκεύει τον κωδικό στην μεταβλητή password και στο τέλος εμφανίζει τον τυχαίο αριθμό.

Με δυσκόλεψε / δεν υλοποίησα

Με δυσκόλεψε η δομή του προγράμματος όπου και κατέληξα στην παραπάνω έκδοση

Links που χρησιμοποίησα

Για την κατανόηση της dev/urandom :

<https://askubuntu.com/questions/192203/how-to-use-dev-urandom>

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

Άσκηση 2

Κώδικας

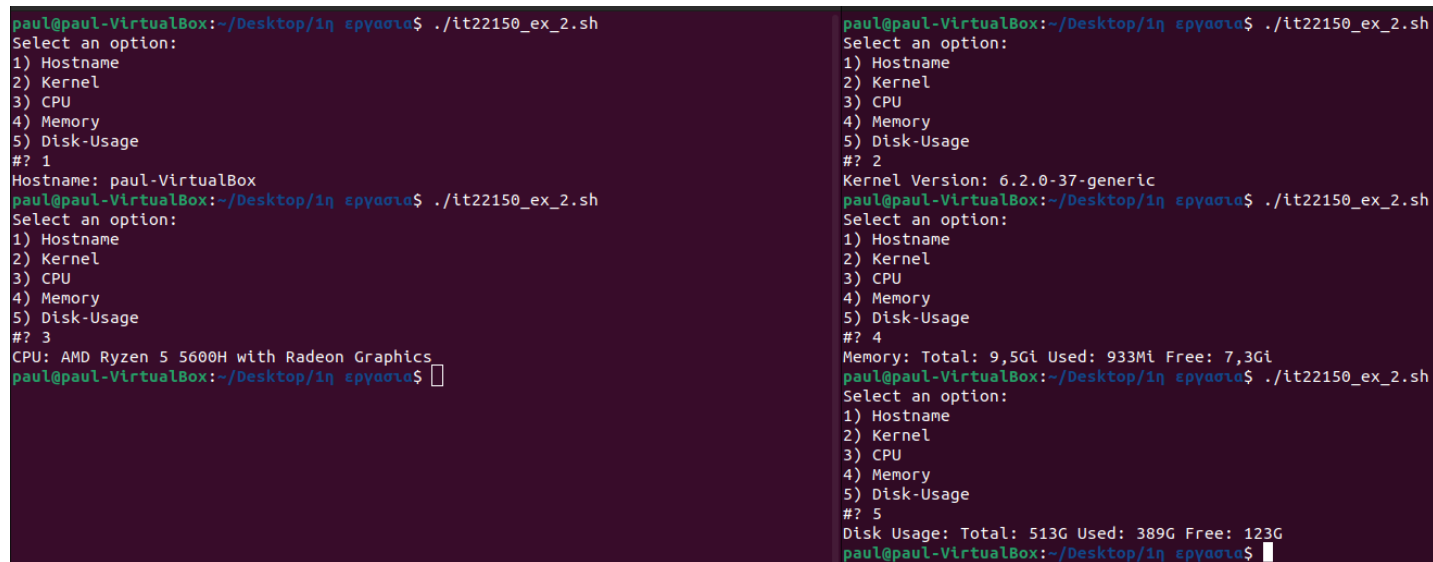
To shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash
show_hostname() {
    echo "Hostname: $(hostname)"
}
show_kernel() {
    echo "Kernel Version: $(uname -r)"
}
show_cpu() {
    echo "CPU: $(lscpu | grep 'Model name' | awk -F: '{print $2}' | sed
's/^[\t]*//')"
}
show_memory() {
    mem_info=$(free -h | awk '/^Mem/ {print "Total: " $2 " Used: " $3 "
Free: " $4}')
    echo "Memory: $mem_info"
}
show_disk_usage() {
    disk_info=$(df -h --total | awk '/total/ {print "Total: " $2 " Used: "
$3 " Free: " $4}')
    echo "Disk Usage: $disk_info"
}
while true; do
    options=("Hostname" "Kernel" "CPU" "Memory" "Disk-Usage")
    echo "Select an option:"
    select opt in "${options[@]}"; do
        case "$opt" in
            "Hostname") show_hostname && exit ;;
            "Kernel") show_kernel && exit ;;
            "CPU") show_cpu && exit ;;
            "Memory") show_memory && exit ;;
            "Disk-Usage") show_disk_usage && exit ;;
            *)
                echo "Error: Μη έγκυρη επιλογή."
            ;;
        esac
    done
done
```

```
exit 1
;;
esac
done
done
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script



```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_2.sh
Select an option:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
#? 1
Hostname: paul-VirtualBox
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_2.sh
Select an option:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
#? 3
CPU: AMD Ryzen 5 5600H with Radeon Graphics
paul@paul-VirtualBox:~/Desktop/1η εργασία$

paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_2.sh
Select an option:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
#? 2
Kernel Version: 6.2.0-37-generic
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_2.sh
Select an option:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
#? 4
Memory: Total: 9,5Gi Used: 933Mi Free: 7,3Gi
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_2.sh
Select an option:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
#? 5
Disk Usage: Total: 513G Used: 389G Free: 123G
paul@paul-VirtualBox:~/Desktop/1η εργασία$
```

Παρατηρήσεις:

Η πρώτη εκτέλεση εκτελεί με την 1) λειτουργία Hostname

Η δεύτερη εκτέλεση εκτελεί με την 2) λειτουργία Kernel

Η τρίτη εκτέλεση εκτελεί με την 3) λειτουργία CPU

Η τέταρτη εκτέλεση εκτελεί με την 4) λειτουργία Memory

Η πέμπτη εκτέλεση εκτελεί με την 5) λειτουργία Disk-Usage

- Έλεγχος παραμέτρων χρήστη

```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_2.sh
Select an option:
1) Hostname
2) Kernel
3) CPU
4) Memory
5) Disk-Usage
#? 0
Error: Μη έγκυρη επιλογή.
paul@paul-VirtualBox:~/Desktop/1η εργασία$
```

Παρατηρήσεις:

Στην εκτέλεση αυτή δεν μας έδωσε το αποτέλεσμα που περιμέναμε, καθώς του δώσαμε λάθος είσοδο και μας ενημέρωσε που έχουμε κάνει λάθος.

Γενικά Σχόλια/Παρατηρήσεις

Δημιουργούμε ένα μενού που επιτρέπει στον χρήστη να επιλέξει από μια σειρά επιλογών για να εμφανίσει πληροφορίες σχετικά με το σύστημα. Κάθε επιλογή καλεί μια συνάρτηση που εμφανίζει συγκεκριμένες πληροφορίες. Ο βρόχος `while true` διατηρεί το μενού ενεργό, και η `select` δημιουργεί έναν ευέλικτο τρόπο για την επιλογή επιλογών από τον χρήστη.

Με δυσκόλεψε / δεν υλοποίησα

-

Links που χρησιμοποίησα

https://linuxhint.com/bash_select_command/

Για τις `lcpu`, `grep`, `awk`, `sed`, `df`, `free`, `uname` χρησιμοποίησα τις παραδειγματικές υλοποιήσεις από το εργαστηριακό μάθημα

Άσκηση 3

Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash
if [ $# -eq 0 ]; then
    echo "Χρησιμοποιήστε το script με έναν κατάλογο ως όρισμα."
    exit 1
fi
logfile="$1"
# Έλεγχος αν το αρχείο καταγραφής υπάρχει
if [ ! -f "$logfile" ]; then
    echo "Το αρχείο καταγραφής ($logfile) δεν υπάρχει."
    exit 1
fi
# Διάβασμα του αρχείου καταγραφής και εμφάνιση πλήθους events ανά ημέρα
echo "Πλήθος events ανά ημέρα:"
grep -oP '^\\w+\\s+\\d+' "$logfile" | sort | uniq -c | awk '{print $2, $3, $1}' | sort -k2M -k3n
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_3.sh /home/paul/Downloads/syslog
Πλήθος events ανά ημέρα:
Nov 27 398
Nov 26 518
Nov 29 594
Dec 1 948
Nov 30 996
Nov 28 1613
paul@paul-VirtualBox:~/Desktop/1η εργασία$
```

Παρατηρήσεις:

Στην εκτέλεση μας βγάζει σαν αποτέλεσμα πόσες φορές συνδεθήκαμε τις συγκεκριμένες ημερομηνίες.

- Έλεγχος παραμέτρων χρήστη

```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_3.sh
Χρησιμοποιήστε το script με έναν κατάλογο ως όρισμα.
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_3.sh /home/paul/Downloads
Το αρχείο καταγραφής (/home/paul/Downloads) δεν υπάρχει.
```

Παρατηρήσεις:

Στις εκτελέσεις αυτές δεν μας έδωσε το αποτέλεσμα που περιμέναμε, καθώς του δώσαμε λάθος είσοδο ή καμία είσοδο της τοποθεσίας ενός αρχείου syslog και μας ενημέρωσε που έχουμε κάνει λάθος.

Γενικά Σχόλια/Παρατηρήσεις

Το script ξεκινά με έναν έλεγχο για την ύπαρξη του καταγραφικού αρχείου (**/var/log/syslog**). Αν το αρχείο δεν υπάρχει, το script εκτυπώνει ένα μήνυμα λάθους και τερματίζει την εκτέλεση. Το script συνεχίζει διαβάζοντας το περιεχόμενο του αρχείου καταγραφής και υπολογίζοντας τον αριθμό των events ανά ημέρα. Χρησιμοποιώντας την **grep** για να εντοπίσει τις ημερομηνίες, την **sort** για να τις ταξινομήσει, την **uniq -c** για να μετρήσει τις εμφανίσεις κάθε ημερομηνίας, και την **awk** για το φORMΑΤ και τον έλεγχο του αποτελέσματος. Τέλος, το script εκτυπώνει τον αριθμό των **events** ανά ημέρα, ταξινομημένα με βάση τον μήνα και την ημερομηνία.

Με δυσκόλεψε / δεν υλοποίησα

-

Links που χρησιμοποιήσα

-

Άσκηση 4

Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash
if [ $# -eq 0 ]; then
    echo "Χρησιμοποιήστε το script με έναν κατάλογο ως όρισμα."
    exit 1
fi
directory=$1
# Τα 5 μεγαλύτερα αρχεία
echo "Τα 5 μεγαλύτερα αρχεία στον κατάλογο:"
find "$directory" -type f -exec du -h {} + | sort -rh | head -n 5
# Αρχεία με πάνω από 1 hard links
echo -e "\nΑρχεία με πάνω από 1 hard links:"
find "$directory" -type f -links +1 -exec stat --format="%n" {} +
# Αρχεία χωρίς read permission
echo -e "\nΑρχεία στα οποία δεν έχουμε read permission:"
find "$directory" -type f ! -readable -exec stat --format="%n" {} +
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_4.sh /home/paul/Downloads/
Ta 5 μεγαλύτερα αρχεία στον κατάλογο:
92M      /home/paul/Downloads/code_1.84.2-1699528352_amd64.deb
89M      /home/paul/Downloads/discord-0.0.36.deb
540K     /home/paul/Downloads/syslog
8,0K     /home/paul/Downloads/CalendarApplication.zip
4,0K     /home/paul/Downloads/1st exercise.sh.zip

Αρχεία με πάνω από 1 hard links:
/home/paul/Downloads/linl_to_file.txt
/home/paul/Downloads/file.txt

Αρχεία στα οποία δεν έχουμε read permission:
/home/paul/Downloads/filenR.txt
paul@paul-VirtualBox:~/Desktop/1η εργασία$
```

Παρατηρήσεις:

Στην εκτέλεση μας βγάδι σαν αποτέλεσμα τα 5 μεγαλύτερα αρχεία στο συγκεκριμένο φάκελο ,μας δείχνει όλα τα hard link που υπάρχουν στον φάκελο και όλα τα αρχεία με μη δικαιώματα διαβάσματος αρχεία

- Έλεγχος παραμέτρων χρήστη

```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_4.sh
Χρησιμοποιήστε το script με έναν κατάλογο ως όρισμα.
paul@paul-VirtualBox:~/Desktop/1η εργασία$
```

Παρατηρήσεις:Σ

Στην εκτέλεση αυτή δεν μας έδωσε το αποτέλεσμα που περιμέναμε, καθώς του δώσαμε λάθος είσοδο ή καμία είσοδο της τοποθεσίας του φακελου που θελουμε να δουμε και μας ενημέρωσε που έχουμε κάνει λάθος.

Γενικά Σχόλια/Παρατηρήσεις

Εύρεση των 5 Μεγαλύτερων Αρχείων: Το script εκτελεί την εντολή **find** για να αναζητήσει όλα τα αρχεία στον καθορισμένο κατάλογο. Τα αποτελέσματα ταξινομούνται με βάση το μέγεθος τους με τη βοήθεια της εντολής **du** και **sort**, και τα πρώτα πέντε μεγαλύτερα αρχεία εμφανίζονται.

Εύρεση Αρχείων με Περισσότερα από 1 Hard Links: Το script εκτελεί και παλι την εντολή **find**, αλλά αυτή τη φορά ελέγχει για αρχεία που έχουν περισσότερα από ένα hard links.

Η εντολή **stat** χρησιμοποιείται για την εμφάνιση των ονομάτων αυτών των αρχείων.

Εύρεση Αρχείων χωρίς Read Permission: Το script αναζητά αρχεία για τα οποία ο χρήστης δεν έχει δικαιώματα ανάγνωσης. Χρησιμοποιώντας τις εντολές **find** και **stat** για την εμφάνιση των ονομάτων αυτών των αρχείων.

Με δυσκόλεψε / δεν υλοποίησα

Δυσκολεύτηκα να δημιουργήσω αρχεία με hard link

Links που χρησιμοποίησα

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

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

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

Άσκηση 5

Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash
if [ $# -eq 0 ]; then
    echo "Χρησιμοποιήστε το script με ένα όνομα domain ως όρισμα."
    exit 1
fi
domain=$1
echo "Ημερομηνία λήξης SSL certificate για το $domain:"
expiration_date=$(echo | openssl s_client -connect "$domain":443
2>/dev/null | openssl x509 -noout -enddate | awk -F= '{print $2}')
expiration_epoch=$(date -d "$expiration_date" +%s)
current_epoch=$(date +%s)
if [ "$current_epoch" -gt "$expiration_epoch" ]; then
    echo "Το πιστοποιητικό έχει λήξει."
else
    echo "Ημερομηνία λήξης: $expiration_date"
fi
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_5.sh www.hua.gr
Ημερομηνία λήξης SSL certificate για το www.hua.gr:
Ημερομηνία λήξης: Oct 31 23:59:59 2024 GMT
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_5.sh www.google.com
Ημερομηνία λήξης SSL certificate για το www.google.com:
Ημερομηνία λήξης: Feb 12 08:09:46 2024 GMT
paul@paul-VirtualBox:~/Desktop/1η εργασία$
```

Παρατηρήσεις:

Στις εκτελέσεις αυτές μας έδωσε το αποτέλεσμα που περιμέναμε ότι οι παραπάνω ιστοσελίδες έχουν άδεια SSL και βλέπουμε πότε λήγει αυτή η άδεια .

- Έλεγχος παραμέτρων χρήστη

```
paul@paul-VirtualBox:~/Desktop/1η εργασία$ ./it22150_ex_5.sh https://eclass.hua.gr/courses/DIT136/
Ημερομηνία λήξης SSL certificate για το https://eclass.hua.gr/courses/DIT136/:
Could not read certificate from <stdin>
Unable to load certificate
Το πιστοποιητικό έχει λήξει.
```

Παρατηρήσεις:

Στην εκτέλεση αυτή δεν μας επιστρέφει το αποτέλεσμα αν το πιστοποιητικό έχει λήξει αλλά μας λέει ότι δεν βρήκε ή δεν μπόρεσε να διαβάσει την άδεια της ιστοσελίδας που του δώσαμε ως είσοδο.

Γενικά Σχόλια/Παρατηρήσεις

Έλεγχος Εισόδου: Το script ξεκινά με έναν έλεγχο για το αν παρέχεται τουλάχιστον ένα όνομα domain ως όρισμα κατά την εκτέλεση. Σε διαφορετική περίπτωση, εμφανίζεται ένα μήνυμα και το script τερματίζει.

Εκτύπωση Ημερομηνίας Λήξης: Το script συνεχίζει να συνδέεται με το καθορισμένο domain στη θύρα 443 (HTTPS) χρησιμοποιώντας το **OpenSSL**. Στη συνέχεια, αποσπά την ημερομηνία λήξης από τις πληροφορίες του πιστοποιητικού.

Έλεγχος Ημερομηνίας: Το script υπολογίζει το timestamp της τρέχουσας ημερομηνίας και συγκρίνει αυτό με το timestamp της ημερομηνίας λήξης του πιστοποιητικού. Εάν η ημερομηνία λήξης έχει παρέλθει, εμφανίζεται ένα μήνυμα ότι το πιστοποιητικό έχει λήξει. Διαφορετικά, εμφανίζεται η ημερομηνία λήξης.

Με δυσκόλεψε / δεν υλοποίησα

Δεν υλοποιήθηκε η εξέταση σε μια ιστοσελίδα που έχει λήξει η ssl άδεια του

Links που χρησιμοποίησα

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

<https://stackoverflow.com/questions/1401482/yyyy-mm-dd-format-date-in-shell-script>

Συνοπτικός Πίνακας

1η Εργασία		
	Υλοποιήθηκε (ΝΑΙ/ΟΧΙ/ΜΕΡΙΚΩΣ)	Παρατηρήσεις
Άσκηση 1	ΝΑΙ	-
Άσκηση 2	ΝΑΙ	-
Άσκηση 3	ΝΑΙ	-
Άσκηση 4	ΝΑΙ	-
Άσκηση 5	ΝΑΙ	Δεν καταφερα να το δοκιμασω σε ιστοσελιδα που εχει ληξει το ssl certificate