



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

Κουβαράς Μαρίνος

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

Ταύρος, 21 Ιανουαρίου 2024

Περιεχόμενα	
Άσκηση 1	3
Κώδικας	3
Ενδεικτικές εκτελέσεις (screenshots):	5
Γενικά Σχόλια/Παρατηρήσεις	5
Με δυσκόλεψε / δεν υλοποίησα	5
Links που χρησιμοποίησα	6
Άσκηση 2	7
Κώδικας	7
Ενδεικτικές εκτελέσεις (screenshots):	10
Γενικά Σχόλια/Παρατηρήσεις	11
Με δυσκόλεψε / δεν υλοποίησα	11
Links που χρησιμοποίησα	11
Άσκηση 3	12
Κώδικας	12
Ενδεικτικές εκτελέσεις (screenshots):	19
Γενικά Σχόλια/Παρατηρήσεις	20
Με δυσκόλεψε / δεν υλοποίησα	21
Links που χρησιμοποίησα	21
Άσκηση 4	22
Κώδικας	22
Ενδεικτικές εκτελέσεις (screenshots):	26
Γενικά Σχόλια/Παρατηρήσεις	27
Με δυσκόλεψε / δεν υλοποίησα	27
Links που χρησιμοποίησα	27
Συνοπτικός Πίνακας	28

Άσκηση 1

Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

#=====
#
#
#      FILE: ap23011_1.sh
#
#      USAGE: ./ap23011_1.sh
#
#  DESCRIPTION: This script returns the files greater than 10MB in your HOME
directory.
#
#      OPTIONS: -d -h
#  REQUIREMENTS: ---
#      BUGS: ---
#      NOTES: ---
#      AUTHOR: Marinos Kouvaras (ap23011), ap23011@hua.gr
#  ORGANIZATION: ---
#      CREATED: 16/1/2023
#      REVISION: v1
#=====
#
```

```

#Create a here document.
help()
{
    cat <<-EOF
        $SCRIPTNAME "This script prints the files greater than 10MB in the home
directory"
        Syntax: ./ap23011_1.sh [-d | -h]
        -d  returns the files
        -h  prints this help message
        ...
    EOF
}

# Check the arguments passed
while getopts ":h" OPTION; do
    case $OPTION in
        h) # -h option
            help
            exit;;
        \?) # If a wrong argument is passed
            help
            exit;;
    esac
done

#find in depth 1 all the files which have size greater than 10MB and are not
secret
RESULT=$(find $HOME -maxdepth 1 -type f -size +10M -not -name ".*" -exec ls
-lhS {} \;)
echo "$RESULT"

```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
(~/WinDocuments)
(19:12:09)→ ./ap23011_1.sh
-rw-r--r-- 1 entech entech 11M Jan 21 16:01 /home/entech/ostechnix.txt
-rw-r--r-- 1 entech entech 156M Dec  4 23:18 /home/entech/z
```

Παρατηρήσεις: Η εντολή επιστρέφει τα αρχεία του Home directory με μέγεθος μεγαλύτερο των 10 MB με αύξουσα κατα μέγεθος σειρά.

- Έλεγχος παραμέτρων χρήστη

```
(~/WinDocuments)
(16:16:32)→ ./ap23011_1.sh -h
: "This script prints the files greater than 10MB in the home directory"
Syntax: ./ap23011_1.sh [-d | -h]
-d returns the files
-h prints this help message
...
```

```
(~/WinDocuments)
(16:19:12)→ ./ap23011_1.sh -p
: "This script prints the files greater than 10MB in the home directory"
Syntax: ./ap23011_1.sh [-d | -h]
-d returns the files
-h prints this help message
...
```

Παρατηρήσεις: Στην επιλογή -h παρατηρούμε πως επιστρέφεται μήνυμα βοήθειας. Ελέγχουμε επίσης την περίπτωση όπου το flag είναι διαφορετικό από τις διατιθέμενες επιλογές. Σε όλες τις περιπτώσεις μας εμφανίζεται μήνυμα βοήθειας.

Γενικά Σχόλια/Παρατηρήσεις

Αρχικά δημιουργούμε ένα here document, με μία function help η οποία μας εξυπηρετεί στην εμφάνιση μηνύματος βοήθειας στο χρήστη. Στη συνέχεια με μία δομή επανάληψης ελέγχουμε τις παραμέτρους που χρησιμοποιούνται. Εάν περάσουμε την επιλογή -h ή οποιαδήποτε άλλη μας επιστρέφεται μήνυμα βοήθειας. Εάν η εντολή εκτελεστεί χωρίς flags τότε μας επιστρέφεται το σωστό αποτέλεσμα.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

<https://www.redhat.com/sysadmin/arguments-options-bash-scripts>

<https://swcarpentry.github.io/shell-novice/07-find.html>

<https://www.tecmint.com/35-practical-examples-of-linux-find-command/>

Άσκηση 2

Κώδικας

To shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

#=====
#
#
#       FILE: ap23011_2.sh
#
#
#       USAGE: ./ap23011_2.sh
#
#
#   DESCRIPTION: This script calculates the time of a capital, relative to TZ
Europe/Athens, based on the given time and
#
#               and the capital.
#
#
#       OPTIONS: <time> <timezone> | -h
#
#   REQUIREMENTS: ---
#
#       BUGS: Ambiguous timezones i.e. Europe/Istanbul
#
#       NOTES: TimeZones -->
https://en.wikipedia.org/wiki/List\_of\_tz\_database\_time\_zones
#
#       AUTHOR: Marinos Kouvaras (ap23011), ap23011@hua.gr
#
#   ORGANIZATION: ---
#
#       CREATED: 16/1/2023
#
#       REVISION: v1
#=====
#
#Create a here document.
help()
{
```

```

cat <<-EOF
    $SCRIPTNAME "This script calculates the time of a capital, relative to TZ
Europe/Athens, based on the given time and
    and the capital."
    "Syntax: ./ap23011_2.sh [ <time> <timezone> | -h]"
    example: ./ap23011_2.sh 18:30 "Europe/Minsk"
    -h for help
    ...
EOF
}

# Variable time from the first argument past
TIME="$1"
# Variable timezone from the second argument past
TIME_ZONE="$2"

# Call here document if no arguments passed or if less than two arguments
passed
if [[ $# -eq 0 ]] || [[ $# -lt 2 ]] ; then
    help
    exit 0
fi

# Check the arguments passed in order to execute the correct action
while getopts ":h" OPTION; do
    case $OPTION in
        h) # -h option
            help
            exit;;
        \?)
            help
            exit;;
    esac
done

# Check the current timezone in Athens

```



```
if [ "$(date +%Z)" == EET ];  
then  
    # winter time, return the winter timezone of the specified Capital  
    echo "Time in "$TIME_ZONE" is $(TZ=$TIME_ZONE date -d $TIME+"EET")"  
else  
    #summer time, return the summer timezone of the specified Capital  
    echo "Time in "$TIME_ZONE" is $(TZ=$TIME_ZONE date -d $TIME+"EEST")"  
fi
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
(~/WinDocuments)
(19:21:05)→ ./ap23011_2.sh 19:00 "Europe/Paris"
Time in Europe/Paris is Sun Jan 21 18:00:00 CET 2024
```

Παρατηρήσεις: Εισοδος ώρας και TZ

- Έλεγχος παραμέτρων χρήστη

```
(19:21:12)→ ./ap23011_2.sh 19:00
"This script calculates the time of a capital, relative to TZ Europe/Athens, based on the given time and
and the capital."
"Syntax: ./ap23011_2.sh [ <time> <timezone> | -h]"
example: ./ap23011_2.sh 18:30 "Europe/Minsk"
-h for help
...
```

```
(19:22:24)→ ./ap23011_2.sh "Europe/Paris"
"This script calculates the time of a capital, relative to TZ Europe/Athens, based on the given time and
and the capital."
"Syntax: ./ap23011_2.sh [ <time> <timezone> | -h]"
example: ./ap23011_2.sh 18:30 "Europe/Minsk"
-h for help
...
```

```
(19:23:18)→ ./ap23011_2.sh
"This script calculates the time of a capital, relative to TZ Europe/Athens, based on the given time and
and the capital."
"Syntax: ./ap23011_2.sh [ <time> <timezone> | -h]"
example: ./ap23011_2.sh 18:30 "Europe/Minsk"
-h for help
...
```

```
(19:23:40)→ ./ap23011_2.sh -h
"This script calculates the time of a capital, relative to TZ Europe/Athens, based on the given time and
and the capital."
"Syntax: ./ap23011_2.sh [ <time> <timezone> | -h]"
example: ./ap23011_2.sh 18:30 "Europe/Minsk"
-h for help
...
```

```
L(19:24:07)→ ./ap23011_2.sh -p
"This script calculates the time of a capital, relative to TZ Europe/Athens, based on the given time and
and the capital."
"Syntax: ./ap23011_2.sh [ <time> <timezone> | -h]"
example: ./ap23011_2.sh 18:30 "Europe/Minsk"
-h for help
...
```

Παρατηρήσεις: Έλεγχος παραμέτρων και περιπτώσεων.

Γενικά Σχόλια/Παρατηρήσεις

Αρχικά δημιουργούμε ένα here document, με μία function help η οποία μας εξυπηρετεί στην εμφάνιση μηνύματος βοήθειας στο χρήστη. Ελέγχουμε αν οι παράμετροι είναι λιγότεροι από 2 ή 0 και αν ναι επιστρέφουμε μήνυμα βοήθειας. Εάν η επιλογή μας είναι -h ή κάποιο άλλο flag επιστρέφουμε επίσης μήνυμα βοήθειας. Τέλος αν η εντολή δοθεί σωστά εξετάζουμε εάν η ζώνη ώρας μας είναι κανονική ή είναι ορισμένη στην κατηγορία daylight saving και αναλόγως περίπτωσης επιστρέφουμε την υπολογιζόμενη ώρα της πόλης που επιλέξαμε.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

https://www.gnu.org/software/bash/manual/html_node/Quoting.html

https://www.gnu.org/software/coreutils/manual/html_node/Examples-of-date.html

Άσκηση 3

Κώδικας

To shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash
#=====
#
#
#      FILE: ap23011_3.sh
#
#      USAGE: ./ap23011_3.sh
#
#  DESCRIPTION: This shell script is a contact mini app. Displays an action
menu to select.
#
#      Data is stored in a file named "contacts.list" in the working
directory.
#
#      OPTIONS: ---
#  REQUIREMENTS: ---
#      BUGS: ---
#      NOTES: ---
#      AUTHOR: Marinos Kouvaras (ap23011), ap23011@hua.gr
#  ORGANIZATION: ---
#      CREATED: 14/1/2023
#      REVISION: v1
#=====
#
# Specify the file we want our data to be stored
FILE=./contacts.list

#The intro message function with the available options
```

```

message () {
    echo "##### CONTACT APP #####"
    echo "# 1. Insert Contact      #"
    echo "# 2. Delete Contact      #"
    echo "# 3. Modify Contact       #"
    echo "# 4. Search Contact        #"
    echo "# 5. Sort Contacts by LastName #"
    echo "# 6. Sort Contacts by FirstNam #"
    echo "# 7. Quit                  #"
    echo "##### CONTACT APP #####"
}

# Function for data insert
insert_contact () {
    while true;
    do
        #Ask for data entry
        echo "Insert <firstname>,<lastname>,<phone1>,<phone2>"
        #Set space delimiter
        IFS=","
        # Read the inputs
        read -a inputs firstname lastname phone1 phone2
        #Check that 4 data fields are entered and that name,surname are alphabetic
        characters and phones are numbers
        if [[ "${#inputs[@]}" -eq 4 && "${inputs[0]}" =~ ^[a-zA-Z]+$ &&
"${inputs[1]}" =~ ^[a-zA-Z]+$ && "${inputs[2]}" =~ ^[0-9]+$ && "${inputs[3]}"
=~ ^[0-9]+$ ]];
        then
            #Write data with the specified format
            NEW_CONTACT=${inputs[0]}', '${inputs[1]}', '${inputs[2]}',
'${inputs[3]}
            # Append contact to the file
            echo "$NEW_CONTACT">>$FILE
            #Inform user that data entry is completed
            echo "Contact inserted"
            break
        fi
    done
}

```

```

    fi
    # Notify for bad input and continue prompting until success entry
    echo "Bad input"
    done
}

# Function for data deletion
delete_contact () {
    # Insert lastname
    read -p "Insert lastname of the contact to be deleted: " LASTNAME
    # Search contact by its last name
    RESULT=$(awk -F ' ' -v last_name="$LASTNAME" '$2 == last_name {print NR, $0}' $FILE)
    #If there is a contact return result
    if [ "${RESULT}" ]
    then
        #Print the result
        echo "$RESULT"
        #Prompt user to select the contact he wants
        read -p "Which contact to delete, insert the left row line? " LINE
        #Confirm action
        read -p "Are you sure Y/N? " OPTION
        if [[ $OPTION == "Y" || $OPTION == "y" ]]
        then
            #Delete contact line
            sed -i "${LINE}d" $FILE
            #Update file with deleting empty line
            sed '/^$/d' $FILE
            #Inform user
            echo "Contact deleted"

            #Exit function
        else [[ $OPTION == "N" || $OPTION == "n" ]]
        then
            return
        fi
    else
        echo "No contacts found"
    fi
}

```

```

    fi
}

# Function for data update
modify_contact () {
    # Search contact by its lastname
    read -p "Insert lastname of the contact to be modified: " LASTNAME
    #Search contact by its lastname
    RESULT=$(awk -F ' ' -v last_name="$LASTNAME" '$2 == last_name {print NR,
$0}' $FILE)
    if [ "${RESULT}" ]
    then
        echo "$RESULT"
        read -p "Which contact to be modified? " LINE
        #save the contact to be replaced in a variable
        OLD_CONTACT=$(awk NR=="$LINE" "$FILE")
        read -p "Are you sure Y/N? " OPTION
        if [[ $OPTION == "Y" || $OPTION == "y" ]]
        then
            while true;
            do
                #Ask for data entry
                echo "Insert <firstname>,<lastname>,<phone1>,<phone2>"
                #Set space delimiter
                IFS=","
                # Read the inputs
                read -a inputs firstname lastname phone1 phone2
                #Check that 4 data fields are entered and that name,surname are
                alphabetic characters and phones are numbers
                if [[ "${#inputs[@]}" -eq 4 && "${inputs[0]}" =~ ^[a-zA-Z]+$ &&
"${inputs[1]}" =~ ^[a-zA-Z]+$ && "${inputs[2]}" =~ ^[0-9]+$ && "${inputs[3]}"
=~ ^[0-9]+$ ]];
                then
                    #Write data with the specified format
                    NEW_CONTACT=${inputs[0]}', '${inputs[1]}', '${inputs[2]}',
'${inputs[3]}

```

```

        #Replace the old contact with the new
        sed -i "${LINE}s|$OLD_CONTACT|$NEW_CONTACT|" "$FILE"
        echo "Contact modified"
        break
    fi
    # Notify for bad input and continue prompting until success entry
    echo "Bad input"
done
else [[ $OPTION == "N" || $OPTION == "n" ]]
    return
fi
else
    echo "No contacts found"
fi
}

# Function for data search
search () {
    read -p "Insert firstname or lastname or phone of the contact: " FIELD
    # Search for matching word
    RESULT=$(grep -inw "$FIELD" "$FILE")
    if [[ $RESULT ]]
    then
        echo "Contacts found"
        echo "$RESULT"
    else
        echo "No contacts found"
    fi
}

# Function for sort by name
sort_name () {
    sort -t ',' -k1 -o $FILE $FILE
    echo "Contacts sorted by name"
    echo "$(cat $FILE)"
}

```



```

# Function for sort by lastname
sort_lastname() {
    sort -t ',' -k2 -o $FILE $FILE
    echo "Contacts sorted by lastname"
    echo "${cat $FILE}"
}

while true; do
    message
    read -p "What do you want to do? " SELECTION
    case "$SELECTION" in
        1)
            insert_contact
            ;;
        2)
            delete_contact
            ;;
        3)
            modify_contact
            ;;
        4)
            search
            ;;
        5)
            sort_name
            ;;
        6)
            sort_lastname
            ;;
        7)
            exit
            ;;
        *)
            echo "Invalid option"
            ;;
    esac
done

```

```
esac  
done
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
(~/WinDocuments)
(19:41:54)→ ./ap23011_3.sh
##### CONTACT APP #####
# 1. Insert Contact          #
# 2. Delete Contact         #
# 3. Modify Contact         #
# 4. Search Contact         #
# 5. Sort Contacts by LastName #
# 6. Sort Contacts by FirstNam #
# 7. Quit                   #
##### CONTACT APP #####
What you want to do? █
```

Παρατηρήσεις: Αρχικό μενού επιλογών

- Έλεγχος παραμέτρων χρήστη

```
What you want to do? 1
Insert <firstname>,<lastname>,<phone1>,<phone2>
Marinos,Kouvaras,6999999990,6999999991
Contact inserted
```

```
What you want to do? 2
Insert lastname of the contact to be deleted: kouvaras
3 marinos, kouvaras, 1111111111, 1111111111
Which contact to delete, insert the left row line? 3
Are you sure Y/N? Y
MARINOS, KOUVARAS, 222222222, 2222222222
william, adams, 333333333, 333333333
silvester, stalone, 44444444, 44444444
Marinos, Kouvaras, 6999999990, 6999999991
Contact deleted
```

```
What you want to do? 3
Insert lastname of the contact to be modified: Kouvaras
4 Marinos, Kouvaras, 6999999990, 6999999991
Which contact to be modified? 4
Are you sure Y/N? Y
Insert <firstname>,<lastname>,<phone1>,<phone2>
marinos,kouvaras,1234567890,0987654321
Contact modified
```

```
What you want to do? 4
Insert firstname or lastname or phone of the contact: marinos
Contacts found
1:MARINOS, KOUVARAS, 222222222, 2222222222
4:marinos, kouvaras, 1234567890, 0987654321
```

```
What you want to do? 5
Contacts sorted by name
MARINOS, KOUVARAS, 222222222, 2222222222
marinos, kouvaras, 1234567890, 0987654321
silvester, stalone, 44444444, 44444444
william, adams, 33333333, 33333333
```

```
What you want to do? 6
Contacts sorted by lastname
MARINOS, KOUVARAS, 222222222, 2222222222
william, adams, 33333333, 33333333
marinos, kouvaras, 1234567890, 0987654321
silvester, stalone, 44444444, 44444444
```

Παρατηρήσεις: Έλεγχος κάθε επιλογής

Γενικά Σχόλια/Παρατηρήσεις

Η εκτέλεση του κώδικα εμφανίζει ένα μενού επιλογών και εξυπηρετεί στην διαχείριση επαφών. Η επιλογή 1 μας επιτρέπει να εισάγουμε επαφές με την μορφοποίηση που μας υποδεικνύεται. Η επιλογή 2 μας δίνει τη δυνατότητα διαγραφής επαφών εκτελώντας αναζήτηση με βάση το επώνυμο, μας επιστρέφει τα αποτελέσματα και στη συνέχεια επιλέγουμε την επαφή που επιθυμούμε να διαγράψουμε, αφού ερωτηθούμε αν είμαστε βέβαιοι η επαφή διαγράφεται εφόσον η απάντησή μας είναι θετική. Εάν η επαφή δεν βρεθεί ή απαντήσουμε αρνητικά επιστρέφουμε στο αρχικό μενού. Η επιλογή 3 μας δίνει την δυνατότητα τροποποίησης μίας επαφής, η αναζήτηση εκτελείται με βάση το επώνυμο και αν υπάρχουν επαφές που αντιστοιχούν τότε επιστρέφονται σαν αποτελέσματα. Η αναζήτηση εκτελείται για ολόκληρη την λέξη κλειδί ενώ είναι case insensitive, σε περίπτωση αδυναμίας εύρεσης επιστρέφουμε στις αρχικές επιλογές. Η επιλογή 4 μας επιτρέπει να αναζητήσουμε επαφές με βάση οποιοδήποτε στοιχείο εγγραφής

τους και επιστρέφει τα αποτελέσματα, σε περίπτωση μη εύρεσης μας επιστρέφει στο αρχικό μενού. Η επιλογή 5 και 6 μας επιστρέφει τις επαφες ταξινομημένες κατά επώνυμο και όνομα αντίστοιχα. Τέλος η επιλογή 7 τερματίζει την εφαρμογή.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

<https://www.regular-expressions.info/replacematch.html>

<https://regex101.com/>

<https://linuxhint.com/bash-script-user-input/>

<https://phoenixnap.com/kb/bash-read>

Άσκηση 4

Κώδικας

To shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

=====
#
#
#       FILE: ap23011_4.sh
#
#       USAGE: ./ap23011_4.sh
#
#   DESCRIPTION: This shell script reads data from a source file named
"access.log" or
#               prompts for download. The -i flag returns the 10 ips with the
most requests while
#               the -c flag return each ip with the country code.
#
#       OPTIONS: -i -c -h
#   REQUIREMENTS: access.log file
#       BUGS: ---
#       NOTES: ---
#       AUTHOR: Marinos Kouvaras (ap23011), ap23011@hua.gr
#   ORGANIZATION: ---
#       CREATED: 14/1/2023
#       REVISION: v1
=====
#
# Variable for our data.
SOURCE_FILE=./access.log
```

```

# This strips off the pathname info from the scriptname,
# ("${0}"), leaving just the last (filename) part:
SCRIPTNAME=${0##*/}

# Create a here document.
help()
{
    cat <<-EOF
    $SCRIPTNAME: Show data from ./access.log file
    -h for help
    -i for the 10 ips with the most calls
    -c for displaying also the country code of each ip
    ...
    EOF
}

# This function checks if there is the source file otherwise it downloads it.
test_file () {
    if [ ! -f $SOURCE_FILE ]; then
        read -p "Source file does not exist, do you want to download it? Y/N "
OPTION
        if [[ $OPTION == "Y" || $OPTION == "y" ]]
        then
            echo "Downloading..."
            curl -s
https://gitlab.com/atsadimas/operating-systems/-/raw/master/exercises/basic-com
mands/access.log -o access.log
            echo "File downloaded"
        else [[ $OPTION == "N" || $OPTION == "n" ]]
        then
            help
            exit
        fi
    fi
}

```

```

# Create a function for the 10 ips with the most calls.
top_ips () {
    #Get the ip field from the data, sort it, find and count the unique values,
    sort them in reverse and print the 10 first of them.
    echo "$( awk '{print $1}' $SOURCE_FILE | sort -k 1n | uniq -c | sort -nr |
head -n 10 )"
}

# Create a function to return the country code for each ip
country_ip () {
    #Save in the variable the result from the top_ips function
    RESULT=$(top_ips)
    #Get only the ips
    IP_ADRESSES=$(echo "$RESULT" | awk '{print $2}')
    #Start a loop
    for ip_addr in $IP_ADRESSES
    do
        #use the api to get the json data and then return the country code
        echo "$ip_addr" -->$(echo $(curl -s
"https://ipapi.co/"${ip_addr}"/json/" | jq '.country_code'))
    done
}

# Check if no argument pass and print help
if [[ $# -eq 0 ]] ; then
    help
    exit
fi

while getopts ":ich" OPTION; do
    case $OPTION in
        i) # -i (ip) option
            test_file
            top_ips
            exit;;
        c)

```



```
test_file
country_ip
exit;;
h) # -h option
help
exit;;
\?) # if a wrong argument is passed
help
exit;;
esac
done
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
(~/WinDocuments)
(20:18:38)→ ./ap23011_4.sh -i
Source file does not exist, do you want to download it? Y/N Y
Downloading...
File downloaded
 3584 154.16.105.25
 3424 191.96.227.224
 2315 83.212.240.12
 1750 194.177.195.232
 1191 143.233.252.2
  654 176.58.227.164
  599 195.251.31.193
  575 74.208.68.9
  413 ::1
  171 79.131.218.250
```

```
(~/WinDocuments)
(20:22:13)→ ./ap23011_4.sh -c
154.16.105.25 -->"US"
191.96.227.224 -->"US"
83.212.240.12 -->"GR"
194.177.195.232 -->"GR"
143.233.252.2 -->"GR"
176.58.227.164 -->"GR"
195.251.31.193 -->"GR"
74.208.68.9 -->"US"
::1 -->null
79.131.218.250 -->"GR"
```

Παρατηρήσεις: Εκτέλεση κώδικα με τις πιθανές επιλογές

- Έλεγχος παραμέτρων χρήστη

```
(~/WinDocuments)
(20:22:21)→ ./ap23011_4.sh
ap23011_4.sh: Show data from ./access.log file
-h for help
-i for the 10 ips with the most calls
-c for displaying also the country code of each ip
...
```

```
(~/WinDocuments)
(20:23:41)→ ./ap23011_4.sh -p
ap23011_4.sh: Show data from ./access.log file
-h for help
-i for the 10 ips with the most calls
-c for displaying also the country code of each ip
...
```

Παρατηρήσεις: Έλεγχος για παραμέτρους

Γενικά Σχόλια/Παρατηρήσεις

Η εκτέλεση του κώδικα μας επιστρέφει ορισμένα αποτελέσματα από ένα αρχείο log file. Εάν το αρχείο δεν υπάρχει δίνεται η επιλογή να το κατεβάσουμε, τα διάφορα flags -i και -c μας επιστρέφουν αντίστοιχα τις 10 ips με τα περισσότερα requests και την χώρα από που αντιστοιχούν.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

<https://www.baeldung.com/linux/bash-pass-function-arg>

<https://wpollock.com/AUnix2/GetoptHeredoc.htm>

<https://sentry.io/answers/determine-whether-a-file-exists-or-not-in-bash/>

<https://stackoverflow.com/questions/8865241/bash-curl-and-variable-in-the-middle-of-the-url>

<https://www.geeksforgeeks.org/awk-command-unixlinux-examples/?ref=lbp>

Συνοπτικός Πίνακας

1η Εργασία		
	Υλοποιήθηκε (ΝΑΙ/ΟΧΙ/ΜΕΡΙΚΩΣ)	Παρατηρήσεις
Άσκηση 1	ΝΑΙ	
Άσκηση 2	ΝΑΙ	
Άσκηση 3	ΝΑΙ	
Άσκηση 4	ΝΑΙ	