



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

ΠΑΝΑΓΙΩΤΗΣ ΠΕΤΕΙΝΑΡΗΣ

it21978

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

Ταύρος, ΧΧ Δεκεμβρίου 2022

Περιεχόμενα

Άσκηση 1	3
Κώδικας	3
Ενδεικτικές εκτελέσεις (screenshots):	4
Γενικά Σχόλια/Παρατηρήσεις	4
Με δυσκόλεψε / δεν υλοποίησα	4
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

To shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

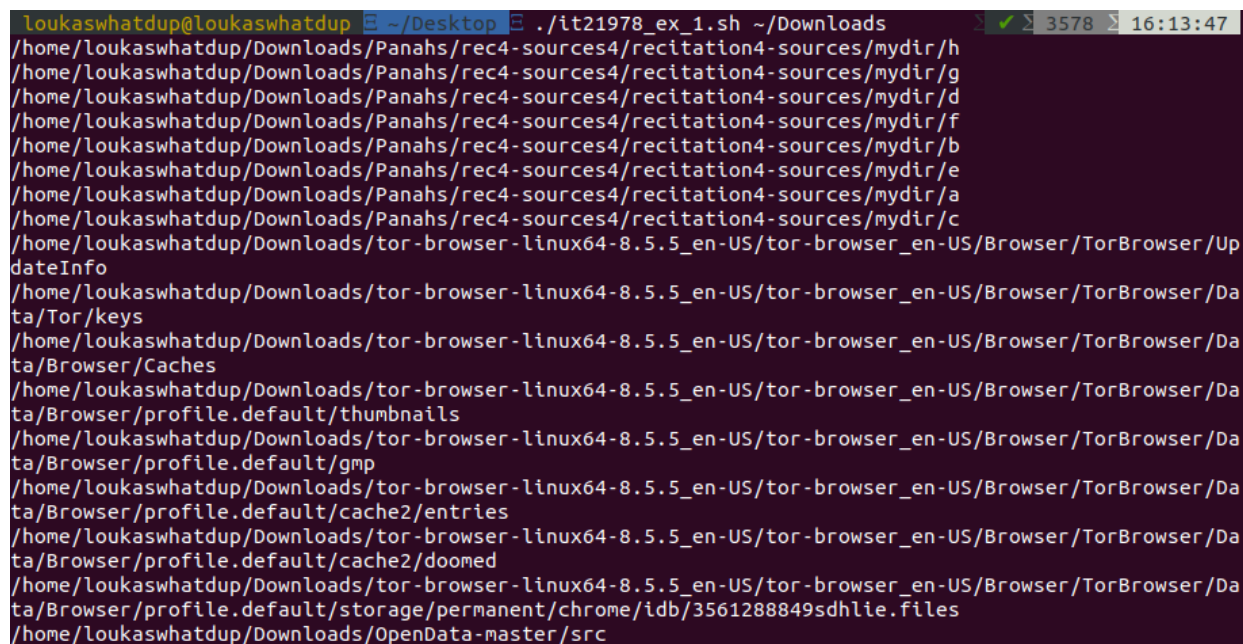
```
#!/bin/bash

if [ ! -d $1 ];
then
    echo " Error Message : You did not provide a directory as an
argument. Please try again!"
    exit 1
fi

find $1 -type d -not -name ".*" -empty -print 2>/dev/null
```

Ενδεικτικές εκτελέσεις (screenshots):

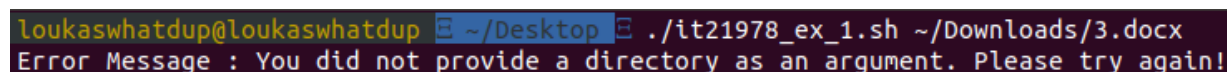
- Επιτυχής εκτέλεση script



```
loukaswhatdup@loukaswhatdup ~/Desktop: ./it21978_ex_1.sh ~/Downloads
/home/loukaswhatdup/Downloads/Panahs/rec4-sources4/recitation4-sources/mydir/h
/home/loukaswhatdup/Downloads/Panahs/rec4-sources4/recitation4-sources/mydir/g
/home/loukaswhatdup/Downloads/Panahs/rec4-sources4/recitation4-sources/mydir/d
/home/loukaswhatdup/Downloads/Panahs/rec4-sources4/recitation4-sources/mydir/f
/home/loukaswhatdup/Downloads/Panahs/rec4-sources4/recitation4-sources/mydir/b
/home/loukaswhatdup/Downloads/Panahs/rec4-sources4/recitation4-sources/mydir/e
/home/loukaswhatdup/Downloads/Panahs/rec4-sources4/recitation4-sources/mydir/a
/home/loukaswhatdup/Downloads/Panahs/rec4-sources4/recitation4-sources/mydir/c
/home/loukaswhatdup/Downloads/tor-browser-linux64-8.5.5_en-US/tor-browser_en-US/Browser/TorBrowser/Up
dateInfo
/home/loukaswhatdup/Downloads/tor-browser-linux64-8.5.5_en-US/tor-browser_en-US/Browser/TorBrowser/Da
ta/Tor/keys
/home/loukaswhatdup/Downloads/tor-browser-linux64-8.5.5_en-US/tor-browser_en-US/Browser/TorBrowser/Da
ta/Browser/Caches
/home/loukaswhatdup/Downloads/tor-browser-linux64-8.5.5_en-US/tor-browser_en-US/Browser/TorBrowser/Da
ta/Browser/profile.default/thumbnails
/home/loukaswhatdup/Downloads/tor-browser-linux64-8.5.5_en-US/tor-browser_en-US/Browser/TorBrowser/Da
ta/Browser/profile.default/gmp
/home/loukaswhatdup/Downloads/tor-browser-linux64-8.5.5_en-US/tor-browser_en-US/Browser/TorBrowser/Da
ta/Browser/profile.default/cache2/entries
/home/loukaswhatdup/Downloads/tor-browser-linux64-8.5.5_en-US/tor-browser_en-US/Browser/TorBrowser/Da
ta/Browser/profile.default/cache2/doomed
/home/loukaswhatdup/Downloads/tor-browser-linux64-8.5.5_en-US/tor-browser_en-US/Browser/TorBrowser/Da
ta/Browser/profile.default/storage/permanent/chrome/idb/3561288849sdhlie.files
/home/loukaswhatdup/Downloads/OpenData-master/src
```

Παρατηρήσεις: Το συγκεκριμένο shell script μας επιστρέφει όλους τους κενούς και μη κρυφούς καταλόγους που βρίσκονται στον κατάλογο Downloads.

- Έλεγχος παραμέτρων χρήστη



```
loukaswhatdup@loukaswhatdup ~/Desktop: ./it21978_ex_1.sh ~/Downloads/3.docx
Error Message : You did not provide a directory as an argument. Please try again!
```

Παρατηρήσεις: Σε αυτήν την περίπτωση το shell script δέχεται ως είσοδο ένα αρχείο doc, και εμφανίζει το error message που έχουμε γράψει για τις περιπτώσεις που στην είσοδο δεν του δώσουμε κάποιον κατάλογο.

Γενικά Σχόλια/Παρατηρήσεις

Αρχικά με το “ if [! -d \$1] “ ελέγχουμε αν στην είσοδο δεν έχουμε κατάλογο στο \$1 που είναι το πρώτο argument και εμφανίζουμε ένα μήνυμα ενημερώνοντας ότι δεν έχουμε λάβει κατάλογο ως είσοδο.

Με την εντολή “ **find \$1 ! -path '*/.*' -type d -empty -print 2>/dev/null** “ ψάχνουμε όλους τους καταλόγους από το \$1 οι οποίοι δεν αρχίζουν με “.”, δηλαδή δεν είναι κρυφοί κατάλογοι, είναι κενοί, με το empty, και τους τυπώνουμε. Με το 2>/dev/null εμφανίζουμε ότι error message μας βγάλει το σύστημα και το ανακατευθύνουμε στους /dev/null όπου εκεί το σύστημα τα διαγράφει.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

<https://stackoverflow.com/questions/9417967/how-to-list-empty-folders-in-linux>

<https://www.cyberciti.biz/faq/bash-find-exclude-all-permission-denied-messages/>

<https://unix.stackexchange.com/questions/611685/finding-all-directories-except-hidden-ones>

Άσκηση 2

Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

datetime=$(date "+%Y-%m-%d,%T")

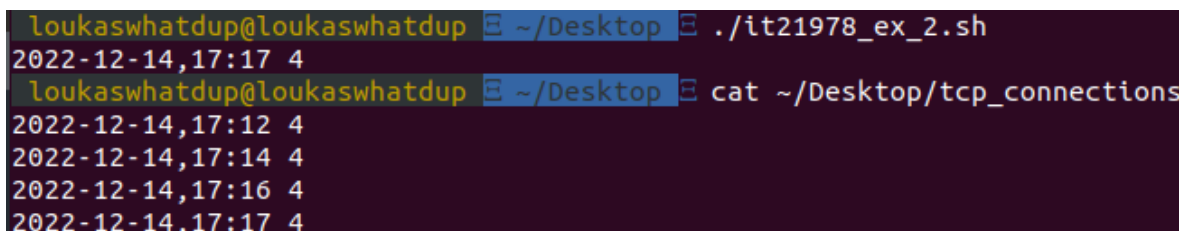
sum_of_tcp_connections=$(ss -t | wc -l)

echo "$datetime $sum_of_tcp_connections"

echo "$datetime $sum_of_tcp_connections" >> ~/Desktop/tcp_connections
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script



```
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_2.sh
2022-12-14,17:17 4
loukaswhatdup@loukaswhatdup ~/Desktop $ cat ~/Desktop/tcp_connections
2022-12-14,17:12 4
2022-12-14,17:14 4
2022-12-14,17:16 4
2022-12-14,17:17 4
```

Παρατηρήσεις: Το shell script εμφανίζει την ημερομηνία και ώρα μαζί με το σύνολο των tcp connection και τις γράφει στο αρχείο tcp_connections.

- Έλεγχος παραμέτρων χρήστη

Παρατηρήσεις: Δεν χρησιμοποιήσαμε κάποια παράμετρο από την είσοδο, συνεπώς δεν έκανα κάποιον έλεγχο.

Γενικά Σχόλια/Παρατηρήσεις

Αρχικά με την εντολή `datetime=$(date "+%Y-%m-%d,%H:%M")` παίρνουμε την ημερομηνία και ώρα στην διάταξη που μας ζητείται.

Με `sum_of_tcp_connections=$(ss -t | wc -l)` παίρνουμε όλες τις established tcp συνδέσεις, μετράμε πόσες είναι μέσω των σειρών και το περνάμε σε μία μεταβλητή.

Στην συνέχεια το ανακατευθύνουμε στο αρχείο tcp_connections και το εμφανίζουμε.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

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

<https://phoenixnap.com/kb/ss-command>

<https://www.baeldung.com/linux/bash-count-lines-in-file>

Άσκηση 3

Κώδικας

To shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

if [ ! -d $1 ];
then
    echo "Error Message : You did not provide a directory as an argument.
Please try again!"
    exit 1
fi

file -b $1/* | awk '{print $1,$2}' | sort | uniq -c | sort -nr
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
loukaswhatdup@loukaswhatdup ~/Desktop$ ./it21978_ex_3.sh ~/Downloads
24 Zip archive
15 PDF document,
14 directory
12 ASCII text,
10 C source,
9 Composite Document
5 Microsoft Word
5 gzip compressed
4 HTML document,
4 Debian binary
4 ASCII text
3 XZ compressed
3 RAR archive
3 PNG image
3 OpenDocument Text
3 Microsoft PowerPoint
2 JPEG image
2 compiled Java
1 XML 1.0
1 UTF-8 Unicode
1 POSIX tar
1 POSIX shell
1 Perl5 module
1 Microsoft OOXML
1 Java source,
1 Java serialization
1 ISO-8859 text
1 ELF 64-bit
1 DOS/MBR boot
```

Παρατηρήσεις: Το shell script παίρνει έναν κατάλογο Downloads και ταξινομεί με βάση ποιός τύπος αρχείου εμφανίζεται πιο συχνά.

- Έλεγχος παραμέτρων χρήστη

```
loukaswhatdup@loukaswhatdup ~/Desktop$ ./it21978_ex_3.sh ~/Downloads/3.docx
Error Message : You did not provide a directory as an argument. Please try again!
```

Παρατηρήσεις: Στην περίπτωση που δεν δίνουμε κατάλογο ως είσοδο μας εμφανίζει ένα error message και μας ενημερώνει ότι δεν έχουμε δώσει κατάλογο ως είσοδο και πρέπει να προσπαθήσουμε ξανά.

Γενικά Σχόλια/Παρατηρήσεις

Αρχικά με το “ **if [! -d \$1]** “ ελέγχουμε αν στην είσοδο δεν έχουμε κατάλογο στο \$1 που είναι το πρώτο argument και εμφανίζουμε ένα μήνυμα ενημερώνοντας ότι δεν έχουμε λάβει κατάλογο ως είσοδο.

Με την εντολή “ **file -b \$1/* | awk '{print \$1,\$2}' | sort | uniq -c | sort -nr** “ παίρνουμε ,από τον κατάλογο \$1 που έχουμε δώσει ως argument, όλους τους τύπους των αρχείων, επιλέγουμε με το **awk** τις σειρές που μας ενδιαφέρουν και μετά με τα **sort** και το **uniq** τα εμφανίζουμε με βάση το πιο εμφανίζεται πιο συχνά.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

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

<https://www.geeksforgeeks.org/sort-command-linuxunix-examples/>

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

<https://phoenixnap.com/kb/linux-file-command>

Άσκηση 4

Κώδικας

To shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

if [ $# -lt 3 ];
then
    echo "Error Message : You must give me at least 2 numbers, so I can
sort them in the right way. Please try again!"
    exit 1
fi

arrangement=$1

shift

numbers=($@)

for i in ${numbers[@]}; do
    if ! [[ $i =~ ^[0-9]+$ ]]; then
        echo "Error: All arguments must be numbers. Please try again!"
        exit 1
    fi
done

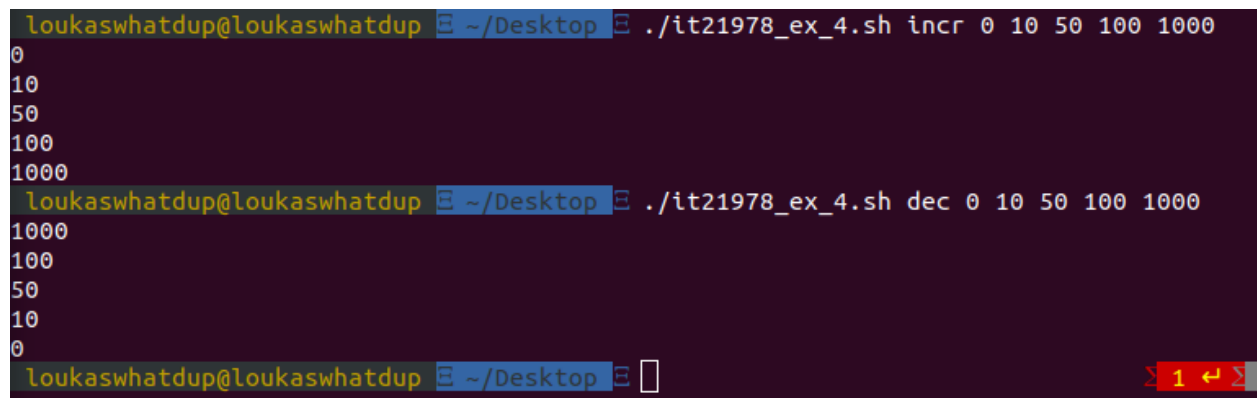
if [ "$arrangement" == "dec" ];
then
    numbers=$(echo "${numbers[@]}" | tr " " "\n" | sort -nr)
    echo "${numbers[@]}" | tr " " "\n"

elif [ "$arrangement" == "incr" ];
then
    numbers=$(echo "${numbers[@]}" | tr " " "\n" | sort -n)
    echo "${numbers[@]}" | tr " " "\n"
```

```
else
    echo "Please try again. State the arrangement and give at least 2
numbers"
fi
```

Ενδεικτικές εκτελέσεις (screenshots):

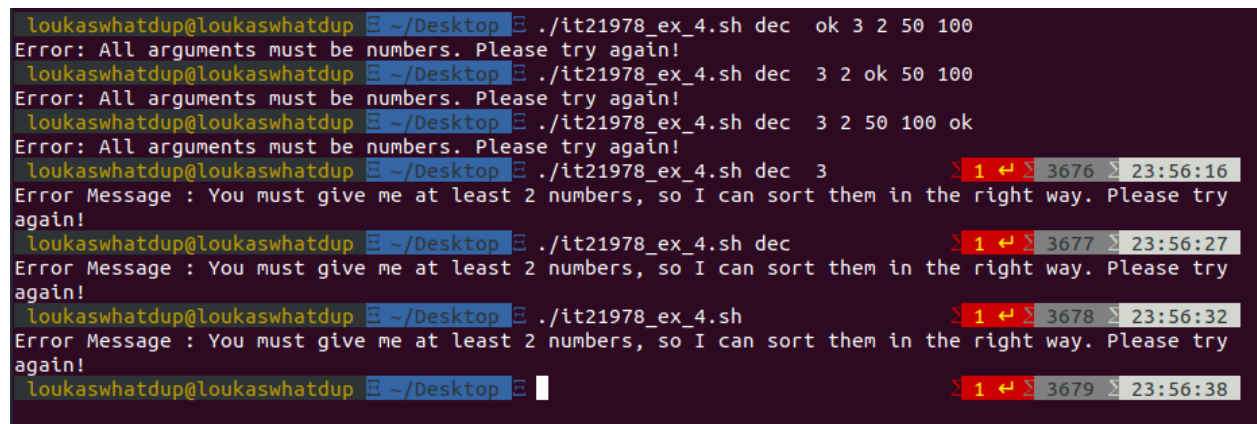
- Επιτυχής εκτέλεση script



```
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_4.sh incr 0 10 50 100 1000
0
10
50
100
1000
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_4.sh dec 0 10 50 100 1000
1000
100
50
10
0
loukaswhatdup@loukaswhatdup ~/Desktop $
```

Παρατηρήσεις: Το shell script λαμβάνει στην είσοδο αριθμούς και την διάταξη και επιστρέφει τους αριθμούς διατεταγμένους.

- Έλεγχος παραμέτρων χρήση



```
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_4.sh dec ok 3 2 50 100
Error: All arguments must be numbers. Please try again!
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_4.sh dec 3 2 ok 50 100
Error: All arguments must be numbers. Please try again!
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_4.sh dec 3 2 50 100 ok
Error: All arguments must be numbers. Please try again!
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_4.sh dec 3
Error Message : You must give me at least 2 numbers, so I can sort them in the right way. Please try again!
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_4.sh dec
Error Message : You must give me at least 2 numbers, so I can sort them in the right way. Please try again!
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_4.sh
Error Message : You must give me at least 2 numbers, so I can sort them in the right way. Please try again!
loukaswhatdup@loukaswhatdup ~/Desktop $
```

Παρατηρήσεις: Στις παραπάνω περιπτώσεις έχουμε τα κατάλληλα error messages σε διάφορα

λάθη που έχει η είσοδος.

Γενικά Σχόλια/Παρατηρήσεις

Αρχικά κάνουμε ένα έλεγχο με το `"if [$# -lt 3]"`, ώστε στην είσοδο να έχουμε τουλάχιστον 3 arguments, την διάταξη και 2 αριθμούς, και εμφανίζουμε ένα error message αν δεν περάσει από τον έλεγχο.

Στην συνέχεια χρησιμοποιούμε το **shift** ώστε να "διώξουμε" την διάταξη από το σύνολο με τους αριθμούς, για να μπορέσουμε να χρησιμοποιήσουμε ξεχωριστά τον πίνακα με τους αριθμούς.

Μετα χρησιμοποιούμε το **for loop** διαπερνώντας τον πίνακα με τους αριθμούς, έναν-έναν και κάνουμε έναν έλεγχο `"if ! [[$i =~ ^[0-9]+$]]"`, δηλαδή όποιο *i* δεν είναι αριθμός θα μπαίνει στον έλεγχο και θα τυπώνει error message σταματώντας το πρόγραμμα.

Τέλος με μία **if-elif-else** κάνουμε έλεγχο για την διάταξη, και βάζουμε την κατάλληλη αύξουσα ή φθίνουσα. Τυπώνουμε τα αποτελέσματα με το `$(echo "${numbers[@]}" | tr " " "\n" | sort -nr)` όπου το `"${numbers[@]}"` μας δίνει τα στοιχεία ένα-ένα, με το `tr " " "\n"` αλλάζουμε τα κενά ανάμεσα στους αριθμούς και βάζουμε το σύμβολο newline για να γράφει το ένα κάτω από το άλλο και με το `sort -nr` ή `-r` τα ταξινομούμε ανάλογα με την διάταξη.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

<https://www.computerhope.com/unix/bash/shift.htm>

<https://fedingo.com/how-to-test-if-variable-is-number-in-shell-script/>

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

<https://stackoverflow.com/questions/16034749/if-elif-else-statement-issues-in-bash>

<https://stackoverflow.com/questions/39690037/is-it-normal-for-printf-s-n-array-to-print-one-line-per-array-element>

Άσκηση 5

Άσκηση 5.1

Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

log_file='apache.log.txt'

awk '{print $1}' $log_file | sort | uniq -c | sort -nr | head
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
loukaswhatdup@loukaswhatdup ~/Desktop $ ./it21978_ex_5_1.sh
1715 194.177.195.20
1702 94.65.67.241
1598 83.212.240.12
784 91.220.176.250
753 141.98.81.38
711 94.71.4.182
570 74.208.68.9
554 ::1
368 193.92.188.170
276 94.71.4.44
```

Παρατηρήσεις: Το shell script εμφανίζει τις 10 IPs με τις περισσότερες επισκέψεις σύμφωνα με το αρχείο `apache.log.txt`.

- Έλεγχος παραμέτρων χρήστη

Παρατηρήσεις: Δεν χρησιμοποιήσαμε κάποια παράμετρο από την είσοδο, συνεπώς δεν έκανα κάποιον έλεγχο.

Γενικά Σχόλια/Παρατηρήσεις

Αρχικά παίρνουμε το αρχείο “`apache.log.txt`” μέσω της μεταβλητής `log_file` και με την εντολή `awk '{print $1}' $log_file | sort | uniq -c | sort -nr | head` παίρνουμε την πρώτη γραμμή του log αρχείου κάνουμε χρήση του `uniq -c`, με την βοήθεια του `sort`, μετρώντας έτσι τις IPs και με το `head` μας εμφανίζει τις 10 πρώτες σειρές.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

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

<https://dev.to/vumdao/using-bash-shell-to-parsing-apache-logs-4pi0>

<https://luther.io/linux/how-to-parse-apache-log-files-with-awk/>

<https://www.sumologic.com/blog/apache-access-log/>

Άσκηση 5.2

Κώδικας

Το shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

log_file='apache.log.txt'

cat $log_file | cut -d'"' -f2 | awk '{print $1 " " $2}' | cut -d? -f1 |
sort | uniq -c | sort -nr | head -n 20
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script

```
loukaswhatdup@loukaswhatdup ~/Desktop ./it21978_ex_5_2.sh
1585 POST /wp-cron.php
1532 POST /xmlrpc.php
752 POST /administrator/index.php
554 OPTIONS *
542 GET /index.php
206 POST /wp-login.php
200 GET /
147 GET /stations/salamina/downld02.txt
147 GET /stations/Harokopio-Athens/downld02.txt
147 GET /stations/geraki/downld02.txt
147 GET /stations/derveni/downld02.txt
147 GET /stations/Delacroix-Attiki/downld02.txt
106 POST /index.php
101 GET /labelgoogle.html
98 GET /stations/geraki/NOAAMO.TXT
98 GET /stations/derveni/NOAAMO.TXT
96 GET /templates/ja_purity_ii/images/favicon.ico
91 -
83 GET /diatrofi/forum/showthread.php
82 GET /stations/geraki/
```

Παρατηρήσεις:

- Έλεγχος παραμέτρων χρήστη

Παρατηρήσεις: Δεν χρησιμοποιήσαμε κάποια παράμετρο από την είσοδο, συνεπώς δεν έκανα κάποιον έλεγχο.

Γενικά Σχόλια/Παρατηρήσεις

Χρησιμοποιούμε την εντολή **cat \$log_file | cut -d'"'"' -f2 | awk '{print \$1 " " \$2}' | cut -d? -f1 | sort | uniq -c | sort -nr | head -n 20** για να εμφανίσουμε τα 20 πιο διάσημα http requests μαζί

με τα path τους. Αρχικά διαβάζουμε το **log_file** με το **cat**. Στην συνέχεια με τα **cut** χωρίζουμε τις γραμμές, την πρώτη φορά με το “ ” και παίρνουμε την δεύτερη στήλη και μετά με το “ ? ” για να πάρουμε μόνο το **path** του url. Με το **awk** παίρνουμε τις δύο στήλες που μας ενδιαφέρουν, δηλαδή το http request μαζί με το requested path. Τέλος κάνουμε χρήση του **uniq -c** μαζί με τα **sort** παίρνωντας αλλά και μετρώντας τα http requests και εμφανίζουμε το 20 πρώτα που τα έχουμε διατάξει με φθίνουσα σειρά με το **head -n 20**.

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

<https://www.inmotionhosting.com/support/server/apache/view-request-type-url-and-response-codes-apache-access-log/>

Άσκηση 5.3

Κώδικας

To shell script που δημιουργήθηκε μαζί με τα σχόλια είναι:

```
#!/bin/bash

log_file='apache.log.txt'

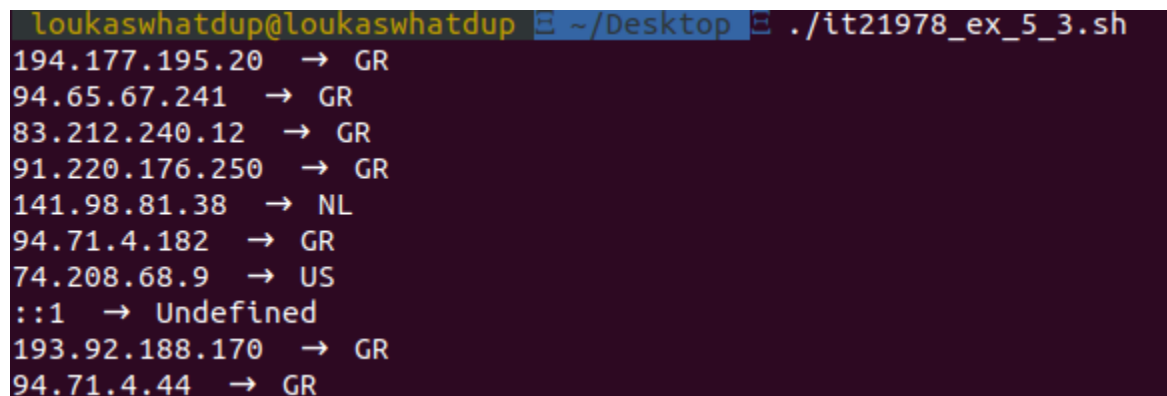
ips=$(awk '{print $1}' $log_file | sort | uniq -c | sort -nr | head )

list_of_ips=$(echo "$ips" | awk '{print $2}')

for i in ${list_of_ips[@]}; do
    countries=$(curl -s "https://ipapi.co/$i/country/")
    echo "$i → ${countries[@]}"
done
```

Ενδεικτικές εκτελέσεις (screenshots):

- Επιτυχής εκτέλεση script



```
loukaswhatdup@loukaswhatdup ~/Desktop$ ./it21978_ex_5_3.sh
194.177.195.20 → GR
94.65.67.241 → GR
83.212.240.12 → GR
91.220.176.250 → GR
141.98.81.38 → NL
94.71.4.182 → GR
74.208.68.9 → US
::1 → Undefined
193.92.188.170 → GR
94.71.4.44 → GR
```

Παρατηρήσεις: Στο συγκεκριμένο shell script παίρνουμε τις IPs του ερωτήματος 5.1 και βρίσκουμε την χώρα προέλευσης μέσω του geolocation API Ipapi.

- Έλεγχος παραμέτρων χρήστη

Παρατηρήσεις: Δεν χρησιμοποιήσαμε κάποια παράμετρο από την είσοδο, συνεπώς δεν έκανα κάποιον έλεγχο.

Γενικά Σχόλια/Παρατηρήσεις

Αρχικά παίρνουμε το αρχείο “apache.log.txt” μέσω της μεταβλητής **log_file** και με την εντολή **awk '{print \$1}' \$log_file | sort | uniq -c | sort -nr | head** παίρνουμε την πρώτη γραμμή του log αρχείου κάνουμε χρήση του **uniq -c**, με την βοήθεια του **sort**, μετρώντας έτσι τις IPs και με το **head** μας εμφανίζει τις 10 πρώτες σειρές.

Στην συνέχεια με την χρήση του **awk '{print \$2}'** παίρνουμε μόνο την στήλη με τις IPs που μας ενδιαφέρουν.

Τέλος κάνουμε ένα **for loop** και παίρνουμε μία μία τις IPs και τις βάζουμε στο **Ipapi API** και αυτό μας επιστρέφει την χώρα από όπου προέρχονται και τις εμφανίζουμε .

Με δυσκόλεψε / δεν υλοποίησα

Links που χρησιμοποίησα

<https://ipapi.co/api/?shell#specific-location-field>

<https://stackoverflow.com/questions/39690037/is-it-normal-for-printf-s-n-array-to-print-one-line-per-array-element>

Συνοπτικός Πίνακας

1η Εργασία		
	Υλοποιήθηκε (ΝΑΙ/ΟΧΙ/ΜΕΡΙΚΩΣ)	Παρατηρήσεις
Άσκηση 1	ΝΑΙ	
Άσκηση 2	ΝΑΙ	
Άσκηση 3	ΝΑΙ	
Άσκηση 4	ΝΑΙ	
Άσκηση 5.1	ΝΑΙ	
Άσκηση 5.2	ΝΑΙ	
Άσκηση 5.3	ΝΑΙ	

