



Εργαστηριακές Ασκήσεις

Ερώτημα 1: Shell Scripting (42 Μονάδες, 6 μονάδες ανά υποερώτημα)

Καλείστε να γράψετε ένα πρόγραμμα το οποίο θα ονομάσετε `logparser.sh` σε BASH shell. Στην αρχή του αρχείου του κώδικα, σε σχόλια, θα τοποθετείτε τα ονοματεπώνυμα και τους ΑΜ των μελών της ομάδας. Το πρόγραμμα προορίζεται για τη διαχείριση του αρχείου καταγραφής `access.log` (σας δίνεται συνημμένα με την εκφώνηση).

Μια εγγραφή αρχείου καταγραφής για το `access.log`, θα μοιάζει κάπως έτσι:

```
127.0.0.1 - userid - [10/Feb/2022:13:55:36 +0300] "GET /server-status HTTP/1.1" 200 2326
```

ή

```
127.0.0.1 - - [29/Mar/2022:17:02:15 +0300] "POST /phpmyadmin/index.php?route=/navigation&ajax_request=1 HTTP/1.1" 200 3941 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0"
```

ή

```
:::1 - - [19/Apr/2022:10:22:23 +0300] "GET /phpmyadmin/themes/pmahomme/img/b_bookmark.png HTTP/1.1" 200 637
```

όπου τα πεδία στο παραπάνω δείγμα εγγραφής αντιπροσωπεύουν τα εξής:

- 127.0.0.1 - Διεύθυνση IP του πελάτη που υπέβαλε το αίτημα (στη συγκεκριμένη περίπτωση localhost) και το :::1 είναι η διεύθυνση loopback στο ipv6, ταυτόσημη με 127.0.0.1 στο ipv4.
- Η πρώτη παύλα ορίζει το δεύτερο πεδίο στο αρχείο καταγραφής που είναι η ταυτότητα του πελάτη. Εάν το αίτημα δεν ελέγχθηκε με έλεγχο ταυτότητας, η καταχώριση στο αρχείο καταγραφής για την τιμή αυτή είναι -.
- Userid- η ταυτότητα του ατόμου που ζητά τον πόρο.
- [10/Feb/2022:13:55:36 +0300]- ημερομηνία, ώρα και χρονική ζώνη του αιτήματος [ημέρα 2 ψηφία, μήνας 3 χαρακτήρες, έτος 4 ψηφία, ώρα 2 ψηφία, λεπτά 2 ψηφία, δευτερόλεπτα 2 ψηφία, ζώνη ('+' | '-') 4 ψηφία]

Ειδικά για τη ζώνη της ώρας, οι σφραγίδες ώρας περιέχουν τα σύμβολα '+' και '-' για να υποδηλώνουν τις ζώνες μετατοπισμένες από το UTC (Coordinated Universal Time) και εάν η ζώνη ώρας είναι μπροστά από το UTC (προς τα ανατολικά) θα έχει πρόσημο + (θετικό), ενώ προς τα δυτικά θα έχει πρόσημο - (αρνητικό). Ο ίδιος ο αριθμός αντιπροσωπεύει τη ζώνη ώρας και τη μετατόπισή της από την ώρα UTC, δηλαδή πόσες ώρες μπροστά ή πίσω από το UTC βρίσκεται το σύστημα.

- "GET /server-status HTTP/1.1"- τύπος αιτήματος και πόρος.
- 200- Κωδικός κατάστασης απόκρισης HTTP.
- 2326- μέγεθος του αντικειμένου που επιστράφηκε στον πελάτη.
- "http://localhost/"- είναι η αναφορά HTTP, η οποία αντιπροσωπεύει τη διεύθυνση από την οποία προήλθε το αίτημα για τον πόρο.
- "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, όπως Gecko) Chrome/78.0.3904.108 Safari/537.36"- παράγοντας χρήστη, ο οποίος προσδιορίζει πληροφορίες σχετικά με το πρόγραμμα περιήγησης που χρησιμοποιεί ο πελάτης πρόσβαση στον πόρο.

Το `logparser.sh` θα πρέπει να εκτελεί τις ακόλουθες λειτουργίες, κατά τις οποίες θα ελέγχεται εάν το <filename> έχει επέκταση ονόματος .log και εφόσον δεν είναι αυτής της μορφής θα εμφανίζει το μήνυμα «Wrong File Argument»:

- i. Όταν το εκτελείτε χωρίς καμία παράμετρο

```
./logparser.sh
```

θα εμφανίζει τον ΑΜ των μελών της ομάδας σας, χωρισμένους μεταξύ τους με «|», χωρίς κανέναν άλλο χαρακτήρα, για παράδειγμα:

```
1234567|1245678|1345654|1343425
```

- ii. Όταν εκτελείτε:



```
./logparser.sh <filename>
```

όπου <filename> είναι το access.log, θα εμφανίζει όλα τα περιεχόμενα του αρχείου που δόθηκε ως όρισμα στο <filename>, στην οθόνη ανά γραμμή.

iii. Όταν εκτελείτε:

```
./logparser.sh <filename> --usrid
```

θα πρέπει να σας εμφανίζει (μετράει) πόσες φορές επαναλαμβάνεται κάθε χρήστης- συμπεριλαμβανομένου και του κενού χρήστη- στα περιεχόμενα του αρχείου και δίπλα το όνομα του χρήστη αλφαβητικά ταξινομημένους, π.χ.

```
7 -  
4 admin  
2 root  
8 user1  
5 user2
```

Για την λειτουργία αυτή θα πρέπει να υλοποιήσετε μια συνάρτηση, την mining_usernames().

Όταν εκτελείτε:

```
./logparser.sh <filename> --usrid <user_id>
```

όπου <user_id> το όνομα του χρήστη, θα πρέπει να ζητήσετε να σας εμφανίζει τα περιεχόμενα του αρχείου που εμφανίζουν κάποιον χρήστη, όπως για παράδειγμα η εκτέλεση:

```
./logparser.sh access.log --usrid root
```

θα εμφανίζει:

```
127.0.0.1 - root - [29/Mar/2022:16:47:20 +0300] "GET /dashboard/images/fastly-  
logo.png HTTP/1.1" 200 1770 "http://localhost/dashboard/" "Mozilla/5.0 (Windows  
NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0"  
  
::1 - root - [19/Apr/2022:10:20:45 +0300] "GET  
/phpmyadmin/themes/pmahomme/img/b_routines.png HTTP/1.1" 200 306
```

iv. Όταν εκτελείτε:

```
./logparser.sh <filename> -method <method_name>
```

όπου <method_name> θα είναι GET ή POST, θα εμφανίζει τα περιεχόμενα του αρχείου που δόθηκε στο <filename> τα οποία αντιστοιχούν στη μέθοδο που δόθηκε ως όρισμα στο -method. Εάν δε συμπληρωθεί το <method_name> ή συμπληρωθεί εσφαλμένα, να εμφανίζεται το μήνυμα «Wrong Method Name».

v. Όταν εκτελείτε:

```
./logparser.sh <filename> --servprot <Network_Protocol>
```

όπου όρισμα στην παράμετρο --servprot θα δίνεται IPv4 ή IPv6 ως <Network_Protocol> και θα εμφανίζει τα περιεχόμενα του αρχείου που δόθηκε ως όρισμα στο <filename> τα οποία αντιστοιχούν στο ένα ή στο άλλο πρωτόκολλο. Εάν το <Network_Protocol> είναι κενό ή οτιδήποτε άλλο, θα εμφανίζει το μήνυμα «Wrong Network Protocol».

vi. Όταν εκτελείτε:

```
./logparser.sh <filename> --browsers
```

θα καλείται η συνάρτηση count_browsers στην οποία θα εμπεριέχεται και η συνάρτηση match(), που εμφανίζει τους browsers Mozilla, Chrome, Safari και Edg και μετά από κάθε browser τον αριθμό των φορών που χρησιμοποιήσαν, όπως για παράδειγμα:

```
Mozilla 234  
Chrome 456  
Safari 789  
Edg 876
```

vii. Όταν εκτελείτε:

```
./logparser.sh <filename> --datum <Date>
```



όπου το <Date> θα πρέπει να προβλεφθεί ότι μπορεί να είναι:

- μήνας, θα αποδέχεται μόνο αλφαριθμητικά της μορφής Jan...Dec και θα εμφανίζει όλες τις εγγραφές του αρχείου access.log που πραγματοποιήθηκαν μέσα σε αυτόν το μήνα ανεξαρτήτως έτους,
- κενό ή οτιδήποτε άλλο, θα πρέπει να εμφανίζει το μήνυμα «Wrong Date».

Απαιτούμενα συνολικά για την άσκηση: Όπου είναι δυνατόν θα πρέπει να γίνει χρήση case, sed και awk.



Ερώτημα 2: Διεργασίες (20 Μονάδες)

Υπολογισμός ολοκληρώματος συνάρτησης

Ο παρακάτω κώδικας υπολογίζει το ολοκλήρωμα της συνάρτησης $f(x) = \ln(x) \cdot \sqrt{x}$ στο διάστημα $[1, 4]$. Ο κώδικας είναι διαθέσιμος στα προγραμματιστικά παραδείγματα του μαθήματος για διαδιεργασιακή επικοινωνία (κατάλογος `os_ipc/4_shmem`). Στον ίδιο κατάλογο είναι διαθέσιμη και η έκδοση του κώδικα που χρησιμοποιεί πολλαπλές διεργασίες και κοινή μνήμη.

```
// integral_seq.c numerical integration - sequential code
#include <stdio.h>
#include <math.h>
#include <sys/time.h>

double get_wtime(void)
{
    struct timeval t;
    gettimeofday(&t, NULL);
    return (double)t.tv_sec + (double)t.tv_usec*1.0e-6;
}

double f(double x)
{
    return log(x)*sqrt(x);
}

// WolframAlpha: integral_1^4 log(x) sqrt(x) dx = 4/9 (4 log(64)-7)
// -> 4.28245881486164
int main(int argc, char *argv[])
{
    double a = 1.0;
    double b = 4.0;
    unsigned long const n = 1e9;
    const double dx = (b-a)/n;

    double S = 0;

    double t0 = get_wtime();
    for (unsigned long i = 0; i < n; i++) {
        double xi = a + (i + 0.5)*dx;
        S += f(xi);
    }
    S *= dx;
    double t1 = get_wtime();

    printf("Time=%lf seconds, Result=%.8f\n", t1-t0, S);

    return 0;
}
```

Ζητούμενο της άσκησης είναι η υλοποίηση του κώδικα χρησιμοποιώντας πολλαπλές διεργασίες και ουρές μηνυμάτων (message queues) για την απαραίτητη επικοινωνία μεταξύ τους. Ο αριθμός των διεργασιών θα πρέπει να ορίζεται δυναμικά, π.χ. να δίνεται ως όρισμα στη γραμμή εκτέλεσης ή να διαβάζεται κατά τον χρόνο εκτέλεσης.