

ΕΡΓΑΣΙΑ #2 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ

Group : ΑΡΤΙΟΙ

A.M. : 1115201200106

Όνομα : Βασίλειος

Επίθετο : Μαυρομμάτης

ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

Η εργασία είναι υλοποιημένη σε mix C/C++/Unix Syscalls.

Όλες οι δομές που έχουν υλοποιηθεί είναι δικές μου από εργασίες μαθημάτων κυρίως Αντικειμενοστραφή Προγραμματισμού, Λειτουργικών Συστημάτων και Δομών Δεδομένων με μικροαλλαγές για βελτιστοποίηση στο ζητούμενο.

Η εργασία αποτελείται από τα εξής 6 αρχεία:

1 αρχείο **MirrorClient.cpp** το οποίο περιέχει την main του προγράμματος και αποτελεί τον κορμό της εργασίας.

2 αρχεία **Helpers.cpp** και το αντίστοιχο header **Helpers.hpp** που έχουν βοηθητικές δομές και συναρτήσεις ελέγχου εγκυρότητας δεδομένων τόσο για το input του χρήστη όσο και για τον έλεγχο των clients που εισέρχονται ή φεύγουν απ'το σύστημα.

1 αρχείο bash script **create_infiles.sh** το οποίο δημιουργεί τα input directories και subdirectories μέχρι το ζητούμενο επίπεδο και τα γεμίζει με αρχεία μεγέθους 1-128 KB κατανεμημένα στα directories με χρήση Round-Robin.

1 αρχείο bash script **get_stats.sh** το οποίο στο τέλος ή και κατά την διάρκεια της προσομοίωσης παίρνει ως είσοδο τα logfiles που δημιουργεί ο κάθε client και παράγει στατιστικά όπως πόσα αρχεία και bytes στάλθηκαν και λήφθηκαν, πόσοι clients συνδέθηκαν και ποιοί, πόσοι clients έφυγαν και ποιοί είναι το μέγιστο και ελάχιστο client id που καταγράφηκε.

1 αρχείο **Makefile** για την δημιουργία του τελικού εκτελέσιμου προγράμματος : **mirror_client**

Αρχικά η main του προγράμματος ελέγχει οτι τα αρχικά arguments που απαιτούνται για την αρχικοποίηση των δομών (και μπορεί να έρθουν με οποιαδήποτε σειρά) είναι έγκυρα και αν ναι προχωράει. Στην συνέχεια δημιουργούνται τα common και mirror directories με έλεγχο για τα πιθανά λάθη όπως η δημιουργία τους από άλλους clients. Έπειτα ο τρέχων client μπαίνει σε έναν ατέρμονο βρόγχο κατά τον οποίον γίνονται οι εξής ενέργειες:

1) Έλεγχος του commonDir για καινούριους clients που μπαίνουν στο σύστημα

a) Αν βρεθούν δημιουργούμε δύο διεργασίες σκλάβους που αναλαμβάνουν να συγχρονιστούν με τις αντίστοιχες του κάθε καινούριου client , ώστε ο ένας σκλάβος να στέλνει αρχεία ενώ ο άλλος να λαμβάνει μέχρις ότου να ληφθούν τα δύο bytes

από τον receiver slave πράγμα που σημαίνει επιτυχή μεταφορά όλων των αρχείων. Τότε ο receiver στέλνει στον master ένα σήμα SIGUSR2 για να τον ενημερώσει ότι όλα πήγαν καλά.

b) Για κάθε path που λαμβάνεται δημιουργούνται τα αντίστοιχα directory trees μέσα στο mirrorDir του client και αν πρόκειται για αρχεία δημιουργούνται και τα αρχεία ενώ η κάθε μεταφορά ολοκληρώνεται και με ένα entry ανά γραμμή που γράφεται στο αντίστοιχο logfile του client και αποτελείται από 3 πεδία, το path το μέγεθος και τον αποστολέα.

c) Αν υπάρξει οποιαδήποτε λάθος στην μεταφορά των αρχείων ο receiver slave αναλαμβάνει να στήσει ένα σήμα SIGUSR1 στον master για να τον ενημερώσει και εκείνος με την σειρά του αυξάνει τον μετρητή των προσπαθειών σκοτώνει τον sender slave και ξαναπροσπαθεί μέχρι την τρίτη φορά, όπου και σταματάει, και συνεχίζει να περιμένει για καινούριους clients.

d) Αν επίσης ένα receiver process περιμένει να διαβάσει από το pipe για πάνω από 30 δευτερόλεπτα, τερματίζει στέλνοντας πάλι ένα σήμα SIGUSR1 στον master οπότε επιστρέφουμε πάλι στο βήμα 1c και από εκεί αν το όριο προσπαθειών έχει ξεπεραστεί τότε στο βήμα 1 για καινούριους clients.

2) Έλεγχος του commonDir για clients που έφυγαν από το σύστημα.

a) Αν βρεθούν τότε δημιουργούμε μια διεργασία σκλάβο η οποία αναλαμβάνει να καθαρίσει το mirrorDir subdirectory του συγκεκριμένου client και να ενημερώσει τον master για την ολοκλήρωση της δουλειάς τους

b) Αν όχι ξεκινάει ο κύκλος ξανά στο βήμα 1.

3) Υπάρχει περίπτωση ο client να λάβει ένα σήμα SIGQUIT(3) ή SIGINT(2) τότε τότε πριν τερματίσει σβήνει όλο το mirrorDir του και από το CommonDir διαγράφει το id file του ώστε να ανιχνευτεί από τυχόν άλλους ενεργούς ακόμα clients στο βήμα 2.

Το πρόγραμμα τερματίζει ομαλά με τα signals SIGQUIT(3) ή SIGINT(2).

COMPILING + EXECUTION:

Ανοίγουμε ένα terminal για compile

```
$ make
```

Ανοίγουμε 1 terminal για κάθε client που θέλουμε να προσθέσουμε στο σύστημα και τρέχουμε την παρακάτω εντολή για τον καθένα από τους n clients στο terminal window του.

```
$ ./mirror_client -n clientIdN -c ./CommonDir -i ./InputDirN -m ./MirrorDirN -b bufferSizeN -l log_fileN
```

(με οποιαδήποτε σειρά αρκεί να υπάρχουν όλα)

```
$ make clean (για clenaup στο τέλος)
```

Και για τα scripts

```
1)$ ./create_infiles.sh dir_name num_of_files num_of_dirs levels
```

```
2)$ cat log_file1 log_file2 ... log_fileN | ./get_stats.sh
```

ΥΛΟΠΟΙΗΘΗΚΕ ΣΕ:

Language: C++(version 11), C, Unix(Bourne again)

OS: Ubuntu 18.04.02 LTS

Text Editor: Sublime Text 3

Debugger: Valgrind 3.13.0, gdb