

Προγραμματισμός Συστήματος

Εργασία 3η

ΠΑΛΥΜΜΕΝΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ – 1115201500133

1. Συνοψη

Εχουν υλοποιηθει ολα τα ζητουμενα της εργασιας.

Web Creator – Bash script

HTTP Server – C

Web Crawler – C

Ενσωματωση JobExecutor στο project

Παραδιδεται Makefile στο οποιο γινεται separate compilation και για την μεταγλωττιση του jobExecutor καλειται το Makefile του αναδρομικα.

Η εκτελεση γινεται ως εξης :

```
./webcreator.sh root_dir text_file w p
```

Το οποιο δημιουργει w*p html pages με τυχαια κομματα κειμενου απο το text_file και τα αποθηκευει στο root_dir

Εχει γινει δοκιμη μεχρι και 1000 pages

```
./myhttpd -p serving_port -c command port -t num_threads  
-d serving_dir
```

```
./mycrawler -h host -p host_server_port -c command port -t num_threads -d save_dir starting_url
```

Το starting_url θα πρεπει να ειναι της μορφης /root_dir/siteX/pageX_YYYY.html

Server και Crawler εχουν δοκιμαστει στο ιδιο μηχανημα (localhost) αλλα και απομακρυσμενα με τον server σε μηχανημα της σχολης.

Εχει δοκιμαστει επισης η εντολη SEARCH απομακρυσμενα μεσω telnet.

Η ομοιοτητα των καταλογων εγινε με την εντολη diff -brief -Nr root_dir save_dir/root_dir

Τα commands δοκιμαστηκαν μεσω telnet και τα read / write πανω απο socket εγιναν οπως υποδειχθηκε.

Χειρισμος σηματος γινεται μονο για το SIGINT το οποιο παρατρεπει τον χρηστη να τερματισει κανονικα με εντολη SHUTDOWN

2. Web Creator

Ακολουθούνται ένα προς ένα τα βήματα που υποδεικνύονται, αν υπάρχει ήδη ο φακέλος γίνεται `purge` , εκτυπώνονται τα καταλληλά μηνύματα και βρίσκει αν όλες οι σελίδες έχουν τουλάχιστον ένα εισερχόμενο `link`. Αυτό γίνεται μέσω ενός αρχείου που κρατείται ένας αριθμός 0 ή 1 που υποδεικνύει αν η σελίδα έχει επιλεγεί σαν `link` από μια άλλη. Αν όλες οι σελίδες έχουν 1 τότε όλες έχουν εισερχόμενα `links` .

3. HTTP Server

Η πολυπλεξή των `serving` and `command ports` γίνεται μέσω της συνάρτησης `poll()` δεν γίνεται `busy waiting`. Το `main thread` δέχεται τις συνδέσεις και αποθηκεύει τους `file descriptors` που επιστρέφονται από την `accept()` σε έναν πίνακα με ένα όριο `MAX_REQUESTS` ο οποίος δεσμεύεται στο `heap`. Ο συγχρονισμός γίνεται με `monitor-style programming` ακολουθώντας την τεχνική `readers-writers`. Για την ομαλή έξοδο όταν δοθεί εντολή `SHUTDOWN` η `global` μεταβλητή `exit_flag` γίνεται `True`, ειδοποιούνται τα `θρεντς` με `broadcast` βλέπουν ότι το `exit_flag == 1` και τερματίζουν. Το `main thread` κάνει `pthread_join` αποδεσμεύει τις `δομές` και τερματίζει.

4. Web Crawler

Το `main thread` κατεβαίνει την πρώτη σελίδα που έχει περαστεί ως `ορίσμα` την αναλύει και αποθηκεύει τα `links` σε μια `ουρά`. Ύστερα ξεκινάνε τα `threads` παίρνουν ένα ένα τα `links` από την `ουρά` και κάνουν την ίδια δουλειά. Το `main thread` συνεχίζει και περιμένει `εντολές` στο `command port`. Για να μην αποθηκευτούν ξανά τα ίδια `links` ελέγχεται αν το αρχείο υπάρχει στον `δίσκο` ή αν είναι ήδη στην `ουρά` ή αν επεξεργάζεται εκείνη την στιγμή από άλλο `thread`. Η συνθήκη τερματισμού του `crawling` είναι η `ουρά` να είναι `αδεια` ΚΑΙ να μην επεξεργάζεται εκείνη την στιγμή κάποιο `link` από κάποιο `thread`. Η `ουρά` αυτή είναι με μορφή `λίστας` αποθηκεύεται το `relative path` και δεν έχει `όριο`. Αν έχει ολοκληρωθεί το `crawling` τότε μπορεί ο `jobExecutor` να αρχίσει κανοντας `fork()` και `exec()` παίρνοντας ως `ορίσμα` το `ονόμα` του `FIFO`.

Η επικοινωνία `web crawler` και `jobExecutor` γίνεται μέσω `blocking pipe` και ο `web crawler` ύστερα στέλνει τα `αποτελέσματα` μέσω `socket` στο `command port`.

Με την εντολή `SHUTDOWN` στέλνεται επίσης εντολή `/exit` στον `jobexecutor` ώστε να τερματίσει κανονικά.