

## Readme File

Εργασία 3 Προγραμματισμός Συστήματος 2013 – 2014

( Γλώσσα : C )

---

Περιεχόμενα :

1. Makefile - Εκτέλεση
2. Δομές
3. Σύντομη περιγραφή κώδικα
4. Σχεδιαστικές επιλογές

---

### 1. Makefile - Εκτέλεση

Το Makefile δημιουργεί τα εκτελέσιμα `remoteClient` και `dataServer` . Κάνει `separate compilation` και έχει `clean mode` .

Τα εκτελέσιμα τρέχουν ως εξής :

```
./dataServer -p <port> -s <thread_pool_size> -q <queue_size>  
./remoteClient -i <server_ip> -p <server_port> -d <directory>
```

Τα ζευγάρια ορισμάτων μπορούν να περαστούν και με διαφορετική σειρά.

---

### 2. Δομές

- Ο `dataServer` έχει την εξής λίστα (List):

```
struct Job{  
    char * filepath ;  
    int sock ;  
};
```

1. *filepath* : filenames (μαζί με τα paths) .
2. *sock* : socket μεταξύ `dataServer` και `remoteClient` .

```
struct Node {  
    Job * job ;  
    Node * next ;
```

```
};
```

```
struct List{  
    Node * first ;  
    Node * current ;  
    int size ;  
};
```

1. *first* : δείκτης στο πρώτο στοιχείο της λίστας ώστε να προσθέτει στοιχείο στην αρχή της λίστας και άρα σε  $O(1)$ .
2. *current* : δείκτης στο τρέχον στοιχείο της λίστας ώστε να το παίρνω σε  $O(1)$ .

Λίστα στην οποία βάζει ο `dataServer` τα `filenames` (μαζί με τα `paths`) και τα `socket` που θα περάσει το εκάστοτε αρχείο . Χρησιμοποιήθηκε ώστε να μπορέσω να πάρω τον αριθμό των αρχείων πριν αυτά αρχίσουν να δίνονται για κατανάλωση στο `fifo array`. Αυτό χρειάζεται ώστε να ξέρει τόσο ο `remoteClient` όσο και ο `dataServer` πότε να κλείσουνε το μεταξύ τους `socket` .

- Ο `dataServer` έχει και το `fifo array` :

```
struct Cell{  
    char * file_name ;  
    int sock ;  
};
```

```
struct Fifo{  
    Cell * cell_array ;  
    int start ;  
    int end ;  
    int count ;  
};
```

Το οποίο `array` είναι παρμένο απο τις σημειώσεις του μαθήματος ουσιαστικά και στο οποίο χρησιμοποιήτε ένα `mutex(mtx_fifo)` και 2 `condition variables(cond_full , cond_empty)` για το μοντέλο `producer consumer`.

---

### 3.Σύντομη περιγραφή κώδικα

Ο `remoteClient` δίνει τον φάκελο προς αντιγραφή μέσω `socket` στον `dataServer` να και εν συνεχεία περιμένει στο `socket` για να λάβει τα δεδομένα (όνομα αρχείου και περιεχόμενα).

Ο `dataServer` λαμβάνει το όνομα του φακέλου προς αντιγραφή και δημιουργεί ένα `thread` πρώτης γραμμής το οποίο βάζει σε λίστα όλα τα ονόματα αρχείων .Στη συνέχεια στέλνει στον `remoteClient` τον αριθμό των αρχείων που θα λάβει και τα βάζει στο `fifo array`.

Εν τω μεταξύ το main thread περιμένει για νέα σύνδεση.

Τα worker threads αναλαμβάνουν να πάρουν απο το fifo array το όνομα του αρχείου και αφού το ανοίξουνε το περνάνε 4096 - 4096 bytes στον remoteClient.

Μόλις ο remoteClient λάβει όλα τα αρχεία που πρέπει κλείνει το socket .Το ίδιο κάνει και ο dataServer.

---

#### 4.Σχεδιαστικές επιλογές

- Κλείσιμο dataServer : Ο remoteClient καλείται ως εξής :

```
./remoteClient -i <server_ip> -p <server_port> -d “ “
```

δηλαδή με μία κενή συμβολοσειρά στη θέση του directory.Ο dataServer αφού μαζέψει το τελευταίο νήμα πρώτης γραμμής ( το οποίο ουσιαστικά κάνει το αίτημα για να κλείσει ο dataServer) και το thread pool ελευθερώνει πόρους και τερματίζει.