

Τριτη Ασκηση Syspro

Αγαπιου Μαρινος
AM: 1115201400002

Για την ασκηση υλοποιησα ολα τα ζητουμενα της εκφωνησης.

Στην ασκηση μου εχω φτιαξει τις εξεις κλασσεις :

struct buffer_entries : που ειναι η κλασση κομβος λιστας που χρησιμοποιω στον κοινο buffer μεταζυ των Worker και Manager.

struct directories_buff: που ειναι η κλασση που περιεχει ολα τα απαραιτητα για να γινεται σωστο διαβασμα και γραψημο απο και προς τον shared buffer των Manager-Worker

struct Content_server : που ειναι η κλασση που χρησιμοποιειται για την αναπαρασταση ενος Content_Server και για την συνδεση σε αυτον .

struct Content_thread: που περιεχει χρησιμες πληροφοριες για ενα thread του Content Server οπως το socket για να επικοινωνει με εναν Manager/Worker.

struct struct_for_delay: που περιεχει το ContentServerID και το delay.

Λειτουργιες :

Αρχικα τρεχουμε τους ContentServer και τον MirrorManager και στην συνεχεια ενεργοποιουμε ολη την διαδικασια μεταφορας αρχειων τρεχοντας τον MirrorInitiator.

Ο MirrorInitiator συνδεεται με τον MirrorManager και του στελνει τα απαραιτητα στοιχεια για την συνδεση με τους ContentServer και τις πληροφοριες για τα αρχεια πανω στα οποια πρεπει να η μεταφορα.Στην συνεχεια ο MirrorManager δημιουργει ενα thread Manager για καθε ContentServer , οι Manager συνδεονται με τους ContentServer ,τους στελνουν ενα αιτημα LIST, και αποθηκευουν τα καταλληλα δεδομενα στον shared buffer για να τα παρουν οι Worker.

Για τον συγχρονισμο των Manager ,Worker χρησιμοποιω δυο condition variable και εναν mutex.

Με λιγα λογια οι Manager πρεπει να περιμενουν στο condition variable “write_cond” οταν υπαρχει στο critical section καποιος αλλος Manager/Worker ή αν ο shared buffer γεμισει , ενω οι Worker πρεπει να περιμενουν στο condition variable “read_cond” στην περιπτωση που υπαρχει στο critical_section αλλος Worker/Manager ή αν ο buffer ειναι αδειος.Ετσι καταφερνω να μην εχω busy waiting .

Ο MirrorServer περιπενει suspended στο condition variable allDone και ξυπνα οταν ο τελευταιος ενεργος worker ειναι ετοιμος να τελειωσει (κρατω στην global μεταβλητη numDevicesDone το πληθος των ενεργων Worker και μολις αυτο γινει ισο με 0 τοτε κανουμε signal τον MirrorServer) και στην συνεχεια στελνει τα καταλληλα στατιστικα πσω στον MirrorInitiator ο οποιος περιμενει να τα διαβασει και να ολοκληρωσει την διαδικασια του .

Μεταγλωτιση αρχειων:

Πατηστε “make clean” και αμεσως μετα “make”.

Εκτελεση αρχειων:

Για τον MirrorInitiator(πχ): ./MirrorInitiator -n localhost -p 3000 -s localhost:3001:askisi1/:1,localhost:3001:haskell/:1

(τα στοιχεια για καθε ContentServer χωριζονται με κομμα “,”)

(ΠΡΟΣΟΧΗ !) Αν θέλουμε να ζητήσουμε έναν folder πρέπει στο τέλος του ονοματος του folder να προσθέσουμε τον χαρακτήρα “/” πχ “fwtografies/”.

Για τον MirrorServer (πχ): ./MirrorServer -p 3000 -m new_foler -w 4

Για τον ContentServer (πχ) ./ContentServer -p 3020 -d my_folder