

3η ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ **ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ**

sdi1900050

Στυλιανός Δημητριάδης

*Παρακάτω θα γίνει επεξήγηση της υλοποίησης μου:
Περιλαμβάνονται τα παρακάτω αρχεία:*

account.h
Makefile
delete_print.c
init_shm.c
reader.c
reader.h
writer.h
writer.c
test.c
test.h
shm.h

Οδηγίες εκτέλεσης του προγράμματος:

make

Για αρχή είναι απαραίτητη η κλήση:

`./init_shm`

για την αρχικοποίηση του shared memory

Για το τεστ ενός μόνο reader:

`./reader -f filename -l recid[,recid] -d time -s shmid`

recid=50(example)

η

recid=30,50(example)

Για το τεστ ενός μόνο writer:

`./writer -f filename -l recid -v value -d time -s shmid`

shmid=my_smh(Αυτό μένει παντα ίδιο καθώς είναι το id του shared memory)

Για το γενικό τεστ του προγράμματος με πολλαπλά ταυτόχρονα random read or write processes

./test -f filename

(η for πάει μέχρι 2 και αυτό μπορεί να αλλάξει)

Τέλος υπάρχει το πρόγραμμα delete_print

./delete_print

το οποίο καλείται αυτόματα όταν κληθεί το test πρόγραμμα στο τέλος του.

- **init_shm.c**

Το init_shm πρόγραμμα είναι υπεύθυνο για την δημιουργία ενός shared memory το οποίο θα χρησιμοποιηθεί από κοινού από readers και writers.

Το shared memory αποτελείται από πίνακες για τα active pid των readers και writers (readers, writers arrays) πέντε ints που αποθηκεύουν το σύνολο των readers, writers accounts processed και average reader writer time, έναν δυσδιάστατο πίνακα που αποθηκεύει το σύνολο των account που γίνονται read την δεδομένη χρονική στιγμή (αποθηκεύει ακόμα και το πόσες φορές γίνεται read το κάθε account για αυτό και η δεύτερη στήλη), και παρόμοιος μονοδιάστατος πίνακας για writers (με μία στήλη). Οι πίνακες αυτοί έχουν στατικό μέγεθος το οποίο έχει γίνει define και μπορεί να αλλάξει από τον χρήστη.

Τέλος το πρόγραμμα δημιουργεί τους named semaphores που θα χρειαστούν για τον συγχρονισμό, file, shared_memory, readers, writers.

- **delete_print.c**

Το πρόγραμμα delete_print κάνει unmap το shared memory και unlink, ενώ εκτυπώνει όλα τα στατιστικά που βρίσκονται στο shared memory.

1. Αριθμός των αναγνωστών που δούλεψαν με το αρχείο εγγραφών.
2. Μέσο χρονικό διάστημα στο οποίο οι αναγνώστες δραστηριοποιήθηκαν.
3. Αριθμός των συγγραφέων που ενημέρωσαν εγγραφές στο αρχείο.
4. Μέσος χρονικός όρος δραστηριότητας συγγραφέων.
5. Συνολικός αριθμός από εγγραφές που είτε διαβάστηκαν είτε ενημερώθηκαν στην διάρκεια εκτέλεσης των πολλαπλών αναγνωστών/συγγραφέων.

- **test.c**

Το πρόγραμμα αυτό χρησιμοποιείται για το testing των readers και writers. Δημιουργεί έναν αριθμό από processes που τρέχουν ταυτόχρονα και τυχαία επιλέγει το ποιο πρόγραμμα θα τρέξει reader ή writer. Αν το πρόγραμμα που θα τρέξει είναι reader τότε επιλέγει τυχαία αν θα διαβάσει πολλά accounts ή ένα, ενώ δίνει ως όρισμα τυχαία accounts και sleep time. Τέλος καλεί την delete_print για να τερματίσει και να εκτυπώσει όλα τα στατιστικά του shared memory.

- **reader.c**

Ο reader ξεκινάει με το parsing των ορισμάτων του user ενώ ανοίγει το shared memory τους named semaphores και το αρχείο με τα accounts. Η λογική του reader και του συγχρονισμού είναι η εξής. Υπάρχει ο semaphore shared memory ο οποίος γίνεται wait κάθε φορά πριν χρησιμοποιηθεί το shared memory ενώ γίνεται post όταν τελειώσει η χρήση του shared memory. Ακόμα χρησιμοποιείται ο semaphore file για να κλειδώνει το file πριν χρησιμοποιηθεί από τον εκάστοτε reader και γίνεται post όταν τελειώσει ο εκάστοτε reader. Με την βοήθεια του acc_being_written γίνονται track τα accounts που γράφονται την δεδομένη χρονική στιγμή. Αν το συγκεκριμένο recid γίνεται write ή ακόμα βρίσκεται μέσα στο range των recid, τότε ο readers semaphore περιμένει μέχρι ο writer να τελειώσει και από την μεριά του να

απελευθερώσει τον reader semaphore ώστε να συνεχίσει την διαδικασία του .Όταν ελευθερωθεί ή αν είναι ήδη ελεύθερος με την σειρά του γεμίζει τον acc_being_read array με το recid ή με τα recid που διαβάζοντα την δεδομένη χρονική στιγμή.Αν ένα account υπάρχει ήδη στο array τότε αυξάνεται το πλήθος του. Με την σειρά του ο reader όταν τελειώσει με το διάβασμα το οποίο κρατάει για random sleep time απελευθερώνει για τα εκάστοτε accounts τον writer semaphore. Τέλος ανανεώνονται τα στατιστικά του shared memory και εκτυπώνονται τα accounts που έγιναν read καθώς και ο μέσος όρος του balance των accounts που διαβάστηκαν.

- **writer.c**

Ο writer λειτουργεί με παρόμοιο σκεπτικό με τον reader με την διαφορά ότι αντι για τον reader semaphore έχει τον writer semaphore ο οποίος περιμένει όταν ένα account γίνεται read απο τον reader και έχει ζητηθεί να γραφτεί το συγκεκριμένο account. Αν ο σεμαφόρος ελευθερωθεί ή είναι ήδη ελεύθερος τότε ο acc_being_written πίνακας γεμίζει με το recid για όσο χρόνο θα γίνει sleep ο συγκεκριμένος writer. Αντίστοιχα όταν τελειώσει την διαδικασία του ο writer θα απελευθερώσει τον reader semaphore.

Ενα παράδειγμα εκτέλεσης του προγράμματος:

make clean

make

./init_shm

./test -f SampleData4Proj3/accounts5000.bin

H

./init_shm

./reader -f SampleDataProj3/accounts5000.bin -l 30 -d 20 -s my_smh

H

./reader -f SampleDataProj3/accounts5000.bin -l 30,50 -d 20 -s my_smh

H

./writer -f SampleDataProj3/accounts5000.bin -l 30 -v 1000 -d 20 -s my_smh

Και

./delete_print(optional)