Λειτουργικά Συστήματα - 1η εργασία

30/11/2019

Περιεχόμενα

1 Το πρόβλημα **1**

2 Γενικά **2**

3 Μεταγλώττιση και Εκτέλεση **2**

4 Modules **3**
4.1 Semaphores 3
4.2 Shared Memory 3

5 Coordinator **3**
5.1 Αρχικοποιήσεις 4
5.2 Διεργασίες 4
5.2.1 Γέννηση 4
5.2.2 Αρχικοποίηση 4
5.2.3 Λειτουργία 5
5.3 Εξαγωγή αποτελεσμάτων 6
5.4 Τερματισμός 6

6 Ενδεικτικές εκτελέσεις **6**
6.1 Ποσοστό Readers/Writers 7
6.2 Αριθμός καταχωρήσεων/πόρων 8
6.3 Αριθμός διεργασιών 9
6.4 Αριθμός επαναλήψεων 10

1 Το πρόβλημα

Η εργασία αποτελεί μια προσομοίωση του προβλήματος Readers-Writers. Το πρόβλημα συνίσταται στη δημιουργία διεργασιών που διαβάζουν ή γράφουν δεδομένα σε ένα σύνολο καταχωρήσεων. Οι καταχωρήσεις αυτές βρίσκονται σε μια κοινή διαμοιραζόμενη μνήμη. Σε κάθε καταχώρηση μπορεί να υπάρχει απεριόριστος

αριθμός από διεργασίες-Readers, με την προϋπόθεση ότι δεν υπάρχει διεργασία-Writer που να επενεργεί στην καταχώρηση. Αντίστροφα, μόνο μια διεργασία-Writer μπορεί να χρησιμοποιεί την καταχώρηση, εφόσον δεν δρουν διεργασίες-Readers εκείνη τη στιγμή. Σκοπός είναι η ομαλή ανάγνωση/εγγραφή των καταχωρήσεων από τις διεργασίες, χωρίς απώλεια ή ανεπιθύμητη αλλοίωση δεδομένων.

2 Γενικά

Έχουν υλοποιηθεί όλα τα ζητούμενα της εργασίας. Ο πηγαίος κώδικας είναι γραμμένος στη γλώσσα C. Το παραδοτέο αρχείο .tar αποτελείται από τα ακόλουθα αρχεία:

- Makefile
- coordinator.c
- semaphores.c / semaphores.h
- shared_mem.c / shared_mem.h

Όλα τα αρχεία περιλαμβάνουν αρκετά σχόλια, ώστε να είναι σαφής η λειτουργία του εκάστοτε κομματιού κώδικα.

3 Μεταγλώττιση και Εκτέλεση

Αρχεία:

- Makefile

Μεταγλώττιση

Στο terminal του Unix δώστε την εντολή

```
$ make
```

Εκτέλεση

Παρέχονται 2 επιλογές εκτέλεσης:

1. Δώστε την εντολή:

   ```
   $ ./coordinator <total_entries> <peers/processes> <loops>
       <readers/writers ratio>
   ```

   αντικαθιστώντας τα ορίσματα στα εισαγωγικά με τις τιμές που επιθυμείτε. Για το readers/writers ποσοστό, δώστε το ακέραιο μέρος (πχ. για 70% ratio, δώστε την τιμή 70).

2. Για εκτέλεση με τα προκαθορισμένα ορίσματα, τις τιμές των οποίων μπορείτε να δείτε στο Makefile, δώστε την εντολή

   ```
   $ make run
   ```

4 Modules

4.1 Semaphores

Αρχεία:

- semaphores.c
- semaphores.h

Έχει υλοποιηθεί ένα module με συναρτήσεις χειρισμού σημαφόρων της System V IPC. Ουσιαστικά, παρέχεται μια πιο εύχρηστη διεπαφή προς τον χρήστη για τη δημιουργία, χρήση και καταστροφή των σημαφόρων, συνοδευόμενη από ενσωματωμένο έλεγχο και αντιμετώπιση σφαλμάτων. Έτσι, ο χρήστης απαλλάσσεται από τα πολλά και δύσχρηστα ορίσματα των συναρτήσεων της IPC, καθώς και από τον συνεχή έλεγχο τιμών επιστροφής που υποδεικνύουν σφάλματα.

4.2 Shared Memory

Αρχεία:

- shared_mem.c
- shared_mem.h

Έχει υλοποιηθεί ένα module με συναρτήσεις χειρισμού διαμοιραζόμενης μνήμης, βασισμένο στην System V IPC. Παρέχεται μια πιο εύχρηστη διεπαφή για τη δημιουργία, σύνδεση, αποσύνδεση και διαγραφή τμημάτων διαμοιραζόμενης μνήμης. Περιλαμβάνεται ενσωματωμένος έλεγχος και αντιμετώπιση σφαλμάτων των συναρτήσεων της IPC.

5 Coordinator

Αρχεία:

- coordinator.c

Είναι το πρόγραμμα που χρησιμοποιείται για να προσομοιώσει το πρόβλημα. Η εκτέλεσή του χωρίζεται σε 4 στάδια:

1. Αρχικοποιήσεις
2. Εκκίνηση και λειτουργία διεργασιών
3. Εξαγωγή αποτελεσμάτων
4. Τερματισμός

5.1 Αρχικοποιήσεις

Εδώ γίνεται η επεξεργασία των ορισμάτων της γραμμής εντολών.
Ακόμη, δημιουργούνται δυο σετ σημαφόρων:

- Ένα σετ μετρητών σημαφόρων, για τις διεργασίες-Readers. Έχουν επιλεχθεί counting semaphores, διότι δεν υπάρχει περιορισμός στο πλήθος των διεργασιών που διαβάζουν την καταχώρηση.
- Ένα σετ δυαδικών σημαφόρων, για τις διεργασίες-Writers. Έχουν επιλεχθεί binary semaphores, διότι μπορεί να πραγματοποιείται το πολύ 1 εγγραφή κάθε χρονική στιγμή.

Κάθε σετ αποτελείται από τόσους σημαφόρους όσες και οι καταχωρήσεις στην διαμοιραζόμενη μνήμη, επομένως σε κάθε καταχώρηση αντιστοιχεί ένα ζευγάρι σημαφόρων (1 για readers, 1 για writers). Όλοι οι σημαφόροι αρχικοποιούνται με την τιμή 1.

Επίσης, δημιουργείται και συνδέεται με το πρόγραμμα η διαμοιραζόμενη μνήμη, στην οποία δεσμεύεται χώρος για ακριβώς τόσες καταχωρήσεις όσες δόθηκαν στη γραμμή εντολών.

Τέλος, αρχικοποιούνται οι καταχωρήσεις. Οι καταχωρήσεις ουσιαστικά περιλαμβάνουν 3 μετρητές, όπου φυλάσσονται:

- το πλήθος των αναγνώσεων της καταχώρησης
- το πλήθος των εγγραφών της καταχώρησης
- το πλήθος των διεργασιών που διαβάζουν την καταχώρηση ταυτόχρονα κάθε χρονική στιγμή

5.2 Διεργασίες

5.2.1 Γέννηση

Η γονική διεργασία καλεί τόσες φορές την συνάρτηση $fork()$, όσες υποδεικνύει το όρισμα που δόθηκε στη γραμμή εντολών. Οι διεργασίες-παιδιά που δημιουργούνται, είναι αυτές που τελικά θα διαβάσουν ή θα γράψουν τις καταχωρήσεις της διαμοιραζόμενης μνήμης.

5.2.2 Αρχικοποίηση

Αρχικοποιείται η γεννήτρια τυχαίων αριθμών με τη χρήση της συνάρτησης $srand()$. Αυτό γίνεται σε κάθε διεργασία-παιδί, διότι αν δεν υπήρχε αυτή η κλήση, τα αποτελέσματα της συνάρτησης $rand()$ θα ήταν ίδια για κάθε διεργασία, εφόσον όλα τα παιδία αναλαμβάνουν την εκτέλεση του κώδικα ακριβώς από το σημείο που γεννήθηκαν κι έπειτα.

Ακόμη, αρχικοποιούνται μετρητές που κρατάνε το πλήθος των αιτημάτων ανάγνωσης και εγγραφής της διεργασίας, καθώς και ο μετρητής του συνολικού χρόνου που χρειάστηκε να περιμένει η διεργασία μέχρι να λάβει τον πόρο/καταχώρηση που είχε αιτηθεί.

5.2.3 Λειτουργία

Στη συνέχεια, κάθε διεργασία-παιδί επαναλαμβάνει, τόσες φορές όσες υποδεικνύει το όρισμα που δόθηκε στη γραμμή εντολών, την ακόλουθη διαδικασία.

1. Επιλέγει τυχαία σε ποιά καταχώρηση θα επενεργήσει
2. Επιλέγει τυχαία τη λειτουργία που θα πραγματοποιήσει, δηλαδή ανάγνωση ή εγγραφή, λαμβάνοντας υπ' όψιν το όρισμα *readers/writers ratio* που δόθηκε στη γραμμή εντολών.
3. Εάν έχει επιλεχθεί η λειτουργία ανάγνωσης τότε:
   (i) Αυξάνεται ο μετρητής των αιτημάτων ανάγνωσης της διεργασίας.
   (ii) Αποθηκεύεται η τωρινή χρονική στιγμή. Θεωρούμε ότι τότε στέλνεται το αίτημα της διεργασίας για απόκτηση του πόρου.
   (iii) Ελέγχεται ο σημαφόρος της καταχώρησης και εαν δεν έχει μηδενική τιμή, τότε παραχωρείται ο πόρος στη διεργασία. Διαφορετικά, η διεργασία γίνεται block και περιμένει να τελειώσει αυτή που γράφει την καταχώρηση εκείνη τη χρονική στιγμή.
   (iv) Αποθηκεύεται η τωρινή χρονική στιγμή. Θεωρούμε ότι τότε εγκρίνεται το αίτημα της διεργασίας για την απόκτηση του πόρου, οπότε και δεσμεύεται.
   (v) Αυξάνεται ο μετρητής του χρόνου αναμονής για την απόκτηση του πόρου.
   (vi) Κατεβάζει τον σημαφόρο των εγγραφών, για να δηλώσει την ύπαρξη διεργασίας-αναγνώστη και αυξάνει τον μετρητή ταυτόχρονων αναγνωστών της καταχώρησης.
   (vii) Χρησιμοποιώντας τη συνάρτηση *usleep()* δημιουργεί μια καθυστέρηση κάποιον microseconds, τα οποία προκύπτουν με τη χρήση εκθετικής κατανομής. Η καθυστέρηση αυτή, χρησιμοποιείται για να αναπαραστήσει τον χρόνο για τον οποίον η διεργασία δεσμεύει τον πόρο/καταχώρηση. Στο διάστημα αυτό, η διεργασία αναστέλλεται προσωρινά.
   (viii) Εαν η διεργασία είναι η μοναδική που πραγματοποιεί ανάγνωση εκείνη τη στιγμή, αυξάνει τον σημαφόρο εγγραφής, δίνοντας τη δυνατότητα σε διεργασίες-εγγραφείς να διεκδικήσουν τον πόρο. Για να πραγματοποιήσει αυτόν τον έλεγχο, κατεβάζει τον σημαφόρο ανάγνωσης για να εμποδίσει άλλους αναγνώστες από τη διεκδίκηση, και τον ανεβάζει ξανά, μετά τον έλεγχο.
4. Εάν έχει επιλεχθεί η λειτουργία εγγραφής τότε:
   (i) Αυξάνεται ο μετρητής των αιτημάτων εγγραφής της διεργασίας.
   (ii) Αποθηκεύεται η τωρινή χρονική στιγμή. Θεωρούμε ότι τότε στέλνεται το αίτημα της διεργασίας για απόκτηση του πόρου.

(iii) Ελέγχεται ο σημαφόρος εγγραφής και εαν δεν έχει μηδενική τιμή, τότε μηδενίζεται και παραχωρείται ο πόρος στη διεργασία. Διαφορετικά, η διεργασία γίνεται block και περιμένει να τελειώσουν αυτές που χρησιμοποιούν την καταχώρηση εκείνη τη χρονική στιγμή.

(iv) Αποθηκεύεται η τωρινή χρονική στιγμή. Θεωρούμε ότι τότε εγκρίνεται το αίτημα της διεργασίας για την απόκτηση του πόρου, οπότε και δεσμεύεται.

(v) Αυξάνεται ο μετρητής του χρόνου αναμονής για την απόκτηση του πόρου.

(vi) Χρησιμοποιώντας τη συνάρτηση *usleep*() δημιουργεί μια καθυστέρηση για έναν τυχαίο αριθμό microseconds, ο οποίος προκύπτει με τη χρήση εκθετικής κατανομής. Η καθυστέρηση αυτή, χρησιμοποιείται για να αναπαραστήσει τον χρόνο για τον οποίο η διεργασία δεσμεύει τον πόρο/καταχώρηση.

(vii) Ανεβάζει τον σημαφόρο εγγραφής, δίνοντας τη δυνατότητα σε άλλες διεργασίες-εγγραφείς να διεκδικήσουν τον πόρο.

5. Εκτυπώνονται μερικά στατιστικά για τη διεργασία: το ID της, οι συνολικές αναγνώσεις και εγγραφές που πραγματοποίησε και ο μέσος χρόνος αναμονής για την απόκτηση πόρων.

6. Η διεργασία τερματίζει.

5.3 Εξαγωγή αποτελεσμάτων

Προσπελαύνονται όλες οι καταχωρήσεις και εκτυπώνονται τα συνολικά πλήθη των αναγνώσεων και των εγγραφών που πραγματοποιήθαν σε αυτές.

5.4 Τερματισμός

Η διαμοιραζόμενη μνήμη αποσυνδέεται από το πρόγραμμα και στη συνέχεια διαγράφεται/αποδεσμεύεται. Ελευθερώνονται οι σημαφόροι που χρησιμοποιήθηκαν. Το πρόγραμμα τερματίζει.

6 Ενδεικτικές εκτελέσεις

Παρατίθονται μερικές ενδεικτικές εκτελέσεις του προγράμματος. Σε κάθε εκτέλεση μελετάμε πως η μεταβολή των παραγόντων που δίνονται από τη γραμμή εντολών, επηρεάζει το μέσο χρόνο αναμονής μιας διεργασίας μέχρι να αποκτήσει τον πόρο που έχει αιτηθεί.

6.1 Ποσοστό Readers/Writers

Σε αυτήν την ενδεικτική εκτέλεση, καταδεικνύουμε τη σημασία του ποσοστού των Readers ανά Writer. Πραγματοποιούμε 2 εκτελέσεις, διατηρώντας σταθερό αριθμό διεργασιών, επαναλήψεων και καταχωρήσεων.

Περιμένουμε ότι για μεγάλο ratio, δηλαδή πολλοί readers, λίγοι writers, θα έχουμε μικρότερους χρόνους αναμονής, σε σχέση με το μικρό ratio. Αυτό είναι αναμενόμενο, διότι οι readers έχουν τη δυνανότητα να προσπελαύνουν την καταχώρηση ταυτόχρονα, οπότε δεν εμποδίζουν ο ένας τον άλλο. Αντίθετα οι writers απαιτούν την αποκλειστικότητα του πόρου, οπότε πρέπει να περιμένουν να τελειώσουν όλες οι άλλες διεργασίες, είτε readers είτε writers, πρωτού μπορέσουν να χρησιμοποιήσουν τον πόρο. Πράγματι, τα αποτελέσματα που θα δείτε στην εικόνα 6.1, επιβεβαιώνουν τους ισχυρισμούς μας.

```
                                        /os_proj1>./coordinator 300 10 200 90

Total Entries: 300
Peers: 10
Loops: 200
Reader/Writer ratio: 90%

>Child process stats:
PID: 7452, Reads: 184, Writes: 16 , Average time:    1204 nsec
PID: 7443, Reads: 177, Writes: 23 , Average time:    1268 nsec
PID: 7449, Reads: 186, Writes: 14 , Average time:    1281 nsec
PID: 7450, Reads: 184, Writes: 16 , Average time:    1277 nsec
PID: 7446, Reads: 182, Writes: 18 , Average time:    1318 nsec
PID: 7448, Reads: 181, Writes: 19 , Average time:    2594 nsec
PID: 7445, Reads: 181, Writes: 19 , Average time:    1462 nsec
PID: 7444, Reads: 181, Writes: 19 , Average time:    1365 nsec
PID: 7447, Reads: 181, Writes: 19 , Average time:    2551 nsec
PID: 7451, Reads: 175, Writes: 25 , Average time:    2967 nsec

>Caller process: shared memory stats
Total Reads : 1812
Total Writes: 188

                                        /os_proj1>./coordinator 300 10 200 10

Total Entries: 300
Peers: 10
Loops: 200
Reader/Writer ratio: 10%

>Child process stats:
PID: 7454, Reads: 24 , Writes: 176, Average time:    6502 nsec
PID: 7459, Reads: 26 , Writes: 174, Average time:   11124 nsec
PID: 7457, Reads: 20 , Writes: 180, Average time:    4414 nsec
PID: 7460, Reads: 19 , Writes: 181, Average time:    3969 nsec
PID: 7455, Reads: 25 , Writes: 175, Average time:    4625 nsec
PID: 7462, Reads: 21 , Writes: 179, Average time:    2320 nsec
PID: 7461, Reads: 26 , Writes: 174, Average time:    3751 nsec
PID: 7456, Reads: 21 , Writes: 179, Average time:    4553 nsec
PID: 7458, Reads: 16 , Writes: 184, Average time:    3661 nsec
PID: 7463, Reads: 24 , Writes: 176, Average time:    7166 nsec

>Caller process: shared memory stats
Total Reads : 222
Total Writes: 1778
```

Εικόνα 1: Η επίδραση του ποσοστού readers ανά writer στο μέσο χρόνο αναμονής

6.2 Αριθμός καταχωρήσεων/πόρων

Σε αυτήν την ενδεικτική εκτέλεση, καταδεικνύουμε τη σημασία του αριθμού των καταχωρήσεων, πάνω στις οποίες δρουν οι διεργασίες. Πραγματοποιούμε 2 εκτελέσεις, διατηρώντας σταθερό τον αριθμό διεργασιών, επαναλήψεων και το ποσοστό readers/writers.

Περιμένουμε ότι για μικρό αριθμό καταχωρήσεων, θα έχουμε μεγαλύτερους χρόνους αναμονής, σε σχέση με έναν μεγάλο αριθμό καταχωρήσεων. Αυτό είναι λογικό, αν σκεφτούμε ότι ο αριθμός των writers και στις 2 εκτελέσεις είναι περίπου ο ίδιος. Επομένως, όταν έχουμε λιγότερους πόρους, πολλοί περισσότεροι readers και writers απαιτούν τον ίδιο πόρο, δημιουργώντας μια μεγάλη "ουρά αναμονής" για την δέσμευση του πόρου. Η ίδια "ουρά αναμονής" είναι μικρότερη όταν υπάρχουν πολλοί πόροι, οπότε και υπάρχει πιο ευρεία διασπορά των διεργασιών σε πόρους. Πράγματι, η πειραματική εκτέλεση στην εικόνα 6.2 επιβεβαιώνει τον συλλογισμό μας.

```
                                /os_proj1>./coordinator 100 10 200 75

Total Entries: 100
Peers: 10
Loops: 200
Reader/Writer ratio: 75%

>Child process stats:
PID: 7008, Reads: 151, Writes: 49 , Average time:    4132 nsec
PID: 7010, Reads: 161, Writes: 39 , Average time:    6195 nsec
PID: 7012, Reads: 146, Writes: 54 , Average time:    2683 nsec
PID: 7016, Reads: 147, Writes: 53 , Average time:    8205 nsec
PID: 7011, Reads: 146, Writes: 54 , Average time:    4436 nsec
PID: 7017, Reads: 138, Writes: 62 , Average time:    4051 nsec
PID: 7014, Reads: 144, Writes: 56 , Average time:    6444 nsec
PID: 7009, Reads: 141, Writes: 59 , Average time:    3860 nsec
PID: 7013, Reads: 161, Writes: 39 , Average time:    3373 nsec
PID: 7015, Reads: 150, Writes: 50 , Average time:    6239 nsec

>Caller process: shared memory stats
Total Reads : 1485
Total Writes: 515

                                /os_proj1>./coordinator 1000 10 200 75

Total Entries: 1000
Peers: 10
Loops: 200
Reader/Writer ratio: 75%

>Child process stats:
PID: 7027, Reads: 152, Writes: 48 , Average time:     777 nsec
PID: 7024, Reads: 146, Writes: 54 , Average time:     808 nsec
PID: 7019, Reads: 151, Writes: 49 , Average time:     785 nsec
PID: 7020, Reads: 143, Writes: 57 , Average time:     831 nsec
PID: 7023, Reads: 141, Writes: 59 , Average time:     763 nsec
PID: 7021, Reads: 155, Writes: 45 , Average time:     764 nsec
PID: 7022, Reads: 147, Writes: 53 , Average time:    1137 nsec
PID: 7028, Reads: 146, Writes: 54 , Average time:     768 nsec
PID: 7025, Reads: 155, Writes: 45 , Average time:     769 nsec
PID: 7026, Reads: 150, Writes: 50 , Average time:     774 nsec

>Caller process: shared memory stats
Total Reads : 1486
Total Writes: 514
```

Εικόνα 2: Η επίδραση του αριθμού των καταχωρήσεων στο μέσο χρόνο αναμονής

6.3 Αριθμός διεργασιών

Σε αυτήν την ενδεικτική εκτέλεση, καταδεικνύουμε την επίδραση του αριθμού των διεργασιών στο μέσο χρόνο αναμονής μιας διεργασίας. Πραγματοποιούμε 2 εκτελέσεις, διατηρώντας σταθερό τον αριθμό καταχωρήσεων, επαναλήψεων και το ποσοστό readers/writers.

Περιμένουμε ότι ο αριθμός διεργασιών είναι ανάλογος με τον μέσο χρόνο αναμονής. Αυτό είναι εύλογο, διότι όσο περισσότερες διεργασίες έχουμε, τόσο περισσότερες συνολικές λειτουργίες ανάγνωσης/εγγραφής πραγματοποιούνται, επομένως αυξάνονται τα συνολικά αιτήματα για δέσμευση των πόρων από όλες τις διεργασίες. Συνεπώς, περισσότερες δεσμεύσεις πόρων συνεπάγονται μεγαλύτερους μέσους χρόνους αναμονής, εφόσον κάθε διεργασία πραγματοποιεί περίπου τον ίδιο αριθμό αναγνώσεων/εγγραφών. Τα αποτελέσματα φαίνονται στην εικόνα 6.3.

```
/os_proj1>./coordinator 500 10 100 75

Total Entries: 500
Peers: 10
Loops: 100
Reader/Writer ratio: 75%

>Child process stats:
PID: 9945, Reads: 75 , Writes: 25 , Average time:     826 nsec
PID: 9950, Reads: 73 , Writes: 27 , Average time:    2720 nsec
PID: 9948, Reads: 73 , Writes: 27 , Average time:     825 nsec
PID: 9949, Reads: 85 , Writes: 15 , Average time:     873 nsec
PID: 9944, Reads: 75 , Writes: 25 , Average time:     839 nsec
PID: 9942, Reads: 78 , Writes: 22 , Average time:    3153 nsec
PID: 9951, Reads: 78 , Writes: 22 , Average time:     836 nsec
PID: 9946, Reads: 74 , Writes: 26 , Average time:    1116 nsec
PID: 9943, Reads: 73 , Writes: 27 , Average time:    6755 nsec
PID: 9947, Reads: 74 , Writes: 26 , Average time:     837 nsec

>Caller process: shared memory stats
Total Reads : 758
Total Writes: 242
```

```
Total Entries: 500
Peers: 1000
Loops: 100
Reader/Writer ratio: 75%

>Child process stats:
PID: 9958, Reads: 83 , Writes: 17 , Average time:   14872 nsec
PID: 9960, Reads: 67 , Writes: 33 , Average time:   11194 nsec
PID: 9982, Reads: 74 , Writes: 26 , Average time:   19157 nsec
PID: 9976, Reads: 77 , Writes: 23 , Average time:   23530 nsec
PID: 9967, Reads: 76 , Writes: 24 , Average time:   12289 nsec
[... 990 processes skipped for showcase purposes ...]
PID: 10927, Reads: 69 , Writes: 31 , Average time:   17729 nsec
PID: 10934, Reads: 67 , Writes: 33 , Average time:   12931 nsec
PID: 10929, Reads: 80 , Writes: 20 , Average time:   13632 nsec
PID: 10842, Reads: 75 , Writes: 25 , Average time:   15693 nsec
PID: 10943, Reads: 80 , Writes: 20 , Average time:   28451 nsec

>Caller process: shared memory stats
Total Reads : 75020
Total Writes: 24980
```

Εικόνα 3: Η επίδραση του αριθμού των διεργασιών στο μέσο χρόνο αναμονής κάθε διεργασίας

6.4 Αριθμός επαναλήψεων

Σε αυτήν την ενδεικτική εκτέλεση, καταδεικνύουμε την επίδραση του αριθμού των επαναλήψεων στο μέσο χρόνο αναμονής μιας διεργασίας. Πραγματοποιούμε 2 εκτελέσεις, διατηρώντας σταθερό τον αριθμό καταχωρήσεων, διεργασιών και το ποσοστό readers/writers.

Περιμένουμε ότι ο αριθμός των επαναλήψεων έχει αμελητέα επίδραση στον μέσο χρόνο αναμονής. Αυτό συμβαίνει δίοτι, αφού αυξάνονται οι συνολικές λειτουργίες ανάγνωσης/εγγραφής από όλες τις διεργασίες, αυξάνεται ο συνολικός χρόνος αναμονής (δηλ. ο αριθμητής), καθώς και ο συνολικός αριθμός αναγνώσεων/εγγραφών της κάθε διεργασίας (δηλ. ο παρονομαστής). Επομένως, ο μέσος χρόνος παραμένει (σχεδόν) αμετάβλητος. Οι συλλογισμοί αυτοί επιβεβαιώνονται και από τα αποτελέσματα της εικόνας 6.4.

```
[redacted]/os_proj1>./coordinator 500 10 1000 75

Total Entries: 500
Peers: 10
Loops: 1000
Reader/Writer ratio: 75%

>Child process stats:
PID: 11461, Reads: 748, Writes: 252, Average time:    2542 nsec
PID: 11463, Reads: 739, Writes: 261, Average time:    3197 nsec
PID: 11466, Reads: 739, Writes: 261, Average time:    2548 nsec
PID: 11467, Reads: 757, Writes: 243, Average time:    2260 nsec
PID: 11464, Reads: 756, Writes: 244, Average time:    2554 nsec
PID: 11459, Reads: 725, Writes: 275, Average time:    3813 nsec
PID: 11462, Reads: 720, Writes: 280, Average time:    2512 nsec
PID: 11460, Reads: 743, Writes: 257, Average time:    2467 nsec
PID: 11465, Reads: 773, Writes: 227, Average time:    2345 nsec
PID: 11468, Reads: 743, Writes: 257, Average time:    2359 nsec

>Caller process: shared memory stats
Total Reads : 7443
Total Writes: 2557

[redacted]/os_proj1>./coordinator 500 10 10000 75

Total Entries: 500
Peers: 10
Loops: 10000
Reader/Writer ratio: 75%

>Child process stats:
PID: 11475, Reads: 7568, Writes: 2432, Average time:    2929 nsec
PID: 11473, Reads: 7491, Writes: 2509, Average time:    2809 nsec
PID: 11478, Reads: 7540, Writes: 2460, Average time:    2840 nsec
PID: 11481, Reads: 7523, Writes: 2477, Average time:    2657 nsec
PID: 11476, Reads: 7472, Writes: 2528, Average time:    2863 nsec
PID: 11477, Reads: 7538, Writes: 2462, Average time:    2765 nsec
PID: 11474, Reads: 7531, Writes: 2469, Average time:    2748 nsec
PID: 11479, Reads: 7574, Writes: 2426, Average time:    2780 nsec
PID: 11472, Reads: 7493, Writes: 2507, Average time:    3004 nsec
PID: 11480, Reads: 7526, Writes: 2474, Average time:    2898 nsec

>Caller process: shared memory stats
Total Reads : 75256
Total Writes: 24744
```

Εικόνα 4: Η αμελητέα επίδραση των επαναλήψεων στο μέσο χρόνο αναμονής