Ονοματεπώνυμο: Γεώργιος Τσίρης
Αριθμός Μητρώου: 1115201700173

- ***<u>Περιγραφή:</u>***

  Στο πρόγραμμα αρχικά ορίζεται ο τύπος Entry που ως μέλη του έχει έναν ακέραιο data (απλά και μόνο για να δώσει νόημα στις διαδικασίες Read και Write που πρόκειται να γίνουν), έναν ακέραιο read_count και έναν ακέραιο write_count που ανά πάσα στιγμή έχουν καταμετηρημένο το πλήθος των Read και Write αντίστοιχα, που έχουν εφαρμοστεί σε αυτό το entry έως τότε. Ακολουθεί ο ορισμός των συναρτήσεων Read και Write για την ανάγνωση και την εγγραφή από και προς το εκάστοτε entry συνοδευόμενη από αύξηση του κατάλληλου μετρητή από αυτούς που περιγράφηκαν παραπάνω.

  Στην συνέχεια ορίζεται ο τύπος Peer_Info που χρησιμοποιείται για την διατήρηση των στατιστικών στοιχείων ενός peer που ζητούνται στο τέλος των επαναλήψεων. Τα μέλη του είναι ένας αριθμός κινητής υποδιαστολής average_waiting που χρησιμοποιείται για την καταχώρηση του μέσου χρόνου αναμονής του peer όπως αυτός διαμορφώνεται, έναν ακέραιο read_count και έναν ακέραιο write_count που ανά πάσα στιγμή έχουν καταμετηρημένο το πλήθος των Read και Write αντίστοιχα, που έχουν εκτελεστεί από αυτό το peer έως τότε.

  Έπειτα ορίζονται συναρτήσεις που αφορούν τις λειτουργίες των σημαφόρων και είναι βασισμένες στα system calls που παρέχονται από το σύστημα. Πιο συγκεκριμένα, ορίζονται οι εξής: η Sem_Create() για την παραγωγή από το σύστημα ενός αριθμού από σημαφόρους, η Sem_Init() για την αρχικοποίηση κάθε σημαφόρου ξεχωριστά στην επιθυμητή τιμή, η Sem_P() για την μείωση της τιμής ενός σημαφόρου κατά 1 (που στην περίπτωση του δυαδικού σημαφόρου σηματοδοτεί την δέσμευση του ή θέτει σε αναμονή την διεργασία αν η τιμή του είναι ήδη 0 μέχρι να γίνει ξανά διαθέσιμος), η Sem_V() για την αύξηση της τιμής του σημαφόρου κατά 1 (που στην περίπτωση του δυαδικού σημαφόρου σηματοδοτεί ότι πλέον είναι αποδεσμευμένος, ενώ στην περίπτωση του counting σημαφόρου σηματοδοτεί απλά την αύξηση του πλήθους).

  Στο σώμα της main() αρχικά γίνεται ένας έλεγχος για το πλήθος των ορισμάτων που δόθηκαν από την γραμμή εντολών και έπειτα προχωράμε σε αποθήκευση τους σε μεταβλητές και εκτύπωση των απαιτήσεων σύμφωνα με αυτές. Υπολογίζεται το μέγεθος της διαμοιραζόμενης μνήμης που θα χρειαστεί (θα πρέπει να επαρκεί τόσο για τα entries όσο και για τις πληροφορίες που θέλουμε για τα peers) και δεσμεύται segment μέσω της κλήσης συστήματος shmget(), το οποίο προσαρτάται στην διαμοιραζόμενη μνήμη μέσω shmat(). Αρχικοποιούνται κατάλληλα τα entries και οι πληροφορίες των peers στον χώρο που τους έχει δοθεί στην διαμοιραζόμενη μνήμη. Εν συνεχεία, δεσμεύονται και αρχικοποιούνται 3 σημαφόροι για κάθε entry (ο πρώτος φροντίζει μόνο ένας reader να εκτελεί το critical section του ανά πάσα στιγμή, ο δεύτερος φροντίζει για τον αμοιβαίο αποκλεισμό μεταξύ readers και writers και ο τρίτος είναι counting σημαφόρος που μετρά το πλήθος των ενεργών διεργασιών που είναι readers) και τέλος 2 επιπλέον σημαφόροι που επιτελούν ρόλο μετρητών (ο πρώτος για το πλήθος των readers που έχουν εμφανιστεί και ο δεύτερος για το πλήθος των ενεργοποιήσεων (είτε ως readers είτε ως writers) που έχουν πραγματοποιηθεί από την αρχή του προγράμματος μέχρι εκείνη την χρονική στιγμή). Αφού ορίσουμε δομές για την αποθήκευση των process ids όπως αυτά θα επιστρέφονται απο την fork(), ξεκινούν οι επαναλήψεις. Σε κάθε επανάληψη, καθένας από τους peers ενεργοποιείται είτε ως reader είτε ως writer. Συγκεκριμένα, για κάθε peer γενιέται διεργασία παιδί μέσω της fork(). Στην διεργασία παιδί, επιλέγεται τυχαία ένα entry στο οποίο θα τελέσει ο peer. Τυχαία αποφασίζεται αν θα λειτουργήσει ως reader ή ως writer, υπό την προυπόθεση ότι το νούμερο των readers (και κατά συνέπεια και των writers αφού είναι συμπληρωματικά μεταξύ

τους) είναι μέσα στα προκαθορισμένα από το input πλαίσια. Όπως και να 'χει, ο counting σημαφόρος για το πλήθος των ενεργοποιήσεων θα πρέπει να αυξηθεί. Στην περίπτωση που είναι reader, αρχικά πρέπει να δεσμευτεί ο σημαφόρος που διασφαλίζει ότι είναι ο μόνος reader στο critical section του. Τότε ο counting σημαφόρος για τους readers τόσο συνολικά όσο και αυτός για τους ενεργούς readers στο entry αυξάνονται με ασφάλεια. Προχωράμε σε έλεγχο τιμής του σημαφόρου των ενεργών readers στο entry μέσω semctl() και αν είναι 1 σημαίνει ότι ειναι ο πρώτος ενεργός reader σε αυτό το entry, οπότε οφείλουμε να δεσμεύσουμε τον σημαφόρο για τον προσωρινό αποκλεισμό των writers που θα προσπαθήσουν να γράψουν σε σε εκείνο entry (θα τίθενται σε αναμονή). Ο σημαφόρος για το critical section των readers αποδεσμεύεται και προσομοιώνεται ο τυχαίος εκθετικά κατανεμημένος χρόνος αναμονής για την προσπέλαση του entry. Έπειτα, προχωράμε στην ανάγνωση αυτή καθεαυτή που όμως επειδή αυξάνει την τιμή των μετρητών του entry (με άλλα λόγια τροποποιεί τμήμα της διαμοιραζόμενης μνήμης) είναι μέρος του critical section. Ως εκ'τούτου θα πρέπει να προηγείται επαναδέσμευση του αντίστοιχου σημαφόρου και έπειτα γίνεται ασφαλώς τόσο το Read() όσο και οι απαραίτητες ανανεώσεις στις πληροφορίες του peer (αύξηση του read_count κατά 1, υπολογισμός του νέου μέσου χρόνου αναμονής). Προτού τερματίσει η διεργασία παιδί που υλοποιεί τον reader, μειώνεται ο counting σημαφόρος των ενεργών reader σε αυτό το entry, ελέγχεται αν ήταν ο τελευταίος ενεργός reader (αν ναι αποδεσμεύεται ο σημαφόρος που αποκλείει τους writers) και αποδεσεύεται ο σημαφόρος που διασφάλιζε ότι ήταν ο μοναδικός reader την δεδομένη χρονική στιγμή που εκτελούσε το critical section του. Τώρα στην περίπτωση που το peer ενεργοποιηθεί ως writer, τα πράγματα είναι απλούστερα. Αρχικά, δεσμεύεται ο σημαφόρος που αποκλείει τους readers και τους υπόλοιπους writers. Προσομοιώνουμε τον τυχαίο εκθετικά κατανεμημένο χρόνο αναμονής για προσπέλαση του entry και στην συνέχεια προχωράμε σε Write() ενός τυχαίου αριθμού (αυθαίρετα επιλέγεται το δίαστημα από το 0 μέχρι το 999 να περιορίζει τον τυχαίο αριθμό) και κάνουμε τις απαραίτητες ανανεώσεις στις πληροφορίες του peer (αύξηση του write_count κατά 1, υπολογισμός του νέου μέσου χρόνου αναμονής). Προτού τερματίσει η διεργασία παιδί που υλοποιεί τον writer, αποδεσμεύεται ο σημαφόρος που αποκλείει τους readers και τους υπόλοιπους writers. Στην περίπτωση της διεργασίας γονέα, αποθηκεύει τα process ids σε μία δομή ώστε να μπορεί να προσδιορίσει μία-μία τις διεργασίες παιδιά που χρειάζεται να περιμένει σε κάθε επάναληψη. Αφού εκτυπωθούν τα ζητούμενα στατιστικά στοιχεία που συλλέχθηκαν κατά την εκτέλεση, οι σημαφόροι διαγράφονται και η διαμοιραζόμενη μνήμη ελευθερώνεται (μετά την αποκόλληση του segment).

- ***<u>Τεχνικές λεπτομέρειες:</u>***
  Υπάρχουν αναλυτικά στα σχόλια του κώδικα, ώστε να φαίνεται που αναφέρεται η κάθε μια.

- ***<u>Παραδείγματα εκτέλεσης:</u>***

Παράδειγμα 1

./ergasia1 5 3 60 2

Requirements:
Peers in each iteration: 5
Number of iterations: 3
Readers are 60% of 15 total activations: 9 readers
Number of entries: 2

Simulation:
Iteration 0:
Peer 1 is WRITING to entry 1 the following data: 261
Peer 0 is READING from entry 0 the following data: 0
Peer 2 is WRITING to entry 0 the following data: 998
Peer 3 is WRITING to entry 0 the following data: 573
Peer 4 is WRITING to entry 1 the following data: 733
Iteration 1:
Peer 0 is WRITING to entry 1 the following data: 472
Peer 1 is READING from entry 0 the following data: 573
Peer 2 is WRITING to entry 1 the following data: 853
Peer 3 is READING from entry 0 the following data: 573
Peer 4 is READING from entry 1 the following data: 853
Iteration 2:
Peer 0 is READING from entry 0 the following data: 573
Peer 1 is READING from entry 1 the following data: 853
Peer 2 is READING from entry 0 the following data: 573
Peer 3 is READING from entry 1 the following data: 853
Peer 4 is READING from entry 1 the following data: 853

Results based on peers:
Peer 0: 2 reads, 1 writes and average waiting time is 0.732493
Peer 1: 2 reads, 1 writes and average waiting time is 0.415332
Peer 2: 1 reads, 2 writes and average waiting time is 1.080007
Peer 3: 2 reads, 1 writes and average waiting time is 1.415651
Peer 4: 2 reads, 1 writes and average waiting time is 0.765565
Total reads done: 9
Total writes done: 6
Average waiting time: 0.881810

Results based on entries:
Entry 0: 5 reads, 2 writes and current data is 573
Entry 1: 4 reads, 4 writes and current data is 853
Total reads done: 9
Total writes done: 6

Παράδειγμα 2

./ergasia1 10 5 35 4

Requirements:
Peers in each iteration: 10
Number of iterations: 5
Readers are 35% of 50 total activations: 18 readers
Number of entries: 4

Simulation:
Iteration 0:
Peer 0 is WRITING to entry 3 the following data: 59
Peer 1 is READING from entry 2 the following data: 0
Peer 2 is WRITING to entry 1 the following data: 609
Peer 3 is WRITING to entry 1 the following data: 135
Peer 4 is READING from entry 3 the following data: 59
Peer 5 is READING from entry 3 the following data: 59
Peer 6 is WRITING to entry 0 the following data: 702
Peer 7 is WRITING to entry 0 the following data: 918
Peer 8 is WRITING to entry 3 the following data: 996
Peer 9 is READING from entry 1 the following data: 135
Iteration 1:
Peer 0 is READING from entry 1 the following data: 135
Peer 1 is WRITING to entry 3 the following data: 414
Peer 2 is WRITING to entry 1 the following data: 148
Peer 3 is WRITING to entry 3 the following data: 178
Peer 4 is READING from entry 3 the following data: 178
Peer 5 is WRITING to entry 0 the following data: 554
Peer 6 is WRITING to entry 1 the following data: 927
Peer 7 is WRITING to entry 1 the following data: 440
Peer 8 is WRITING to entry 0 the following data: 561
Peer 9 is WRITING to entry 0 the following data: 740
Iteration 2:
Peer 0 is READING from entry 0 the following data: 740
Peer 1 is READING from entry 3 the following data: 178
Peer 2 is WRITING to entry 1 the following data: 925
Peer 3 is WRITING to entry 0 the following data: 787
Peer 4 is WRITING to entry 3 the following data: 719
Peer 5 is WRITING to entry 0 the following data: 424
Peer 6 is WRITING to entry 2 the following data: 463
Peer 7 is WRITING to entry 0 the following data: 872
Peer 8 is WRITING to entry 0 the following data: 271
Peer 9 is READING from entry 2 the following data: 463
Iteration 3:
Peer 0 is WRITING to entry 3 the following data: 85
Peer 1 is WRITING to entry 3 the following data: 464
Peer 2 is READING from entry 0 the following data: 271
Peer 3 is READING from entry 0 the following data: 271
Peer 4 is WRITING to entry 1 the following data: 110
Peer 5 is READING from entry 3 the following data: 464
Peer 6 is READING from entry 3 the following data: 464
Peer 7 is WRITING to entry 0 the following data: 179

Peer 8 is WRITING to entry 3 the following data: 932
Peer 9 is WRITING to entry 2 the following data: 394
Iteration 4:
Peer 0 is READING from entry 0 the following data: 179
Peer 1 is WRITING to entry 1 the following data: 319
Peer 2 is READING from entry 0 the following data: 179
Peer 3 is READING from entry 3 the following data: 932
Peer 4 is READING from entry 1 the following data: 319
Peer 5 is WRITING to entry 1 the following data: 86
Peer 6 is READING from entry 0 the following data: 179
Peer 7 is WRITING to entry 2 the following data: 161
Peer 8 is WRITING to entry 1 the following data: 656
Peer 9 is WRITING to entry 1 the following data: 122

Results based on peers:
Peer 0: 3 reads, 2 writes and average waiting time is 0.244052
Peer 1: 2 reads, 3 writes and average waiting time is 0.398151
Peer 2: 2 reads, 3 writes and average waiting time is 0.668021
Peer 3: 2 reads, 3 writes and average waiting time is 0.146966
Peer 4: 3 reads, 2 writes and average waiting time is 0.268362
Peer 5: 2 reads, 3 writes and average waiting time is 0.415517
Peer 6: 2 reads, 3 writes and average waiting time is 0.282100
Peer 7: 0 reads, 5 writes and average waiting time is 0.446095
Peer 8: 0 reads, 5 writes and average waiting time is 0.221267
Peer 9: 2 reads, 3 writes and average waiting time is 0.318756
Total reads done: 18
Total writes done: 32
Average waiting time: 0.340929

Results based on entries:
Entry 0: 6 reads, 10 writes and current data is 179
Entry 1: 3 reads, 11 writes and current data is 122
Entry 2: 2 reads, 3 writes and current data is 161
Entry 3: 7 reads, 8 writes and current data is 932
Total reads done: 18
Total writes done: 32

Σχολιασμός παραδειγμάτων:

Παρατηρούμε ότι τα αποτελεσμάτα που προκύπτουν μέσω της πληροφορίας που συλλέγουμε για τους peers ταυτίζονται με τα αποτελέσμα που προκύπτουν από την πληροφορία των entries. Συγκεκριμένα, το Total reads done και το Total writes done που εκτυπώνονται παραπάνω αποτιμούνται και με τους δύο τρόπους στις ίδιες τιμές αντίστοιχα.