

Λειτουργικά Συστήματα

3η εργασία - Σεπτέμβριος 2020

Θεοδώρα Παναγέα

1115201400135

Η εργασία υλοποιήθηκε σε περιβάλλον macOS στο VScode.

Το παραδοτέο αποτελείται από τα εξής αρχεία:

- **feeder.c**: Είναι ο γονιός, και μέσα στο συγκεκριμένο αρχείο υπάρχει η main
- **child.c**: Είναι ο κώδικας για τη διεργασία παιδί
- **functions.c/functions.h**: Περιέχουν μέσα βιβλιοθήκες, κλειδιά για τους σημαφόρους, καθώς και μία συνάρτηση για errors.
- **sems.c/sems.h**: Πηγαίος κώδικας και header file για τις συναρτήσεις σχετικές με τους σημαφόρους
- **shm.c/shm.h**: Πηγαίος κώδικας και header file για τις συναρτήσεις σχετικές με τη διαμοιραζόμενη μνήμη

FEEDER

Ο feeder παράγει έναν πίνακα με τυχαίους μη αρνητικούς αριθμούς, διάστασης M (nums[M]). Δημιουργεί επίσης τη shared memory και αρχικοποιεί τους σημαφόρους. Με τη fork παράγει n children (child), ελέγχει με μια switch, αν βρισκόμαστε στο παιδί, στο γονιό ή έχει προκύψει error. Αν βρισκόμαστε στο παιδί, του περνάει ως ορίσματα τα n,M μέσω της execvp(). Αν βρισκόμαστε στο γονιό, περνάει το id του παιδιού σε έναν πίνακα child[n]. Αν υπάρξει λάθος, τερματίζει. Ο feeder έχει και το ρόλο του writer. Ως writer, ξεκινάει κατεβάζοντας το σημαφόρο του (writers_sem), ούτως ώστε όταν πάει να ξαναγράψει να κολλήσει εκεί, και να μπορεί να συνεχίσει μόνο αν του το επιτρέψουν τα παιδιά (θα εξηγηθεί παρακάτω). Αυτό έχει σκοπό να μην γράψει ο writer απευθείας όλους τους ακέραιους στη shared memory, πριν προλάβουν να διαβάσουν οι readers. Γράφει στη shared memory, τον ακέραιο που θέλει να μεταφέρει, ένα timestamp σε milliseconds. Επίσης, μηδενίζει το μετρητή readcount της shared memory που λέει πόσοι την έχουν διαβάσει, και αυξάνει το μετρητή written, που δηλώνει πόσους ακέραιους έχει καταφέρει να περάσει/γράψει. Έπειτα, δίνει σήμα στους readers να ξεκινήσουν, σηκώνοντας το σημαφόρο rmutex. Τέλος, ο γονιός περιμένει να τελειώσουν όλα τα παιδιά, ούτως ώστε να μη μείνει κανένα ορφανό, και μετά αποδεσμεύει τη shared memory.

CHILD

Το κάθε παιδί (child) έχει έναν πίνακα από flags M θέσεων, που υποδηλώνει αν έχει διαβάσει ή όχι τον ακέραιο της θέσης M του αρχικού πίνακα, δηλαδή τον M-οστό ακέραιο κι αρχικοποιείται σε 0, από τη στιγμή που δεν έχει διαβάσει κανέναν αριθμό. Αφότου κάνει attach τη shared memory(shared), κοιτάζει τον ακέραιο στο πεδίο shared->written, και αν δεν τον έχει διαβάσει (δηλαδή αν στον πίνακα flags, στη συγκεκριμένη θέση, έχει 0), τότε προχωράει στην ανάγνωση της shared memory. Κάθε reader που μπαίνει, αυξάνει το readcount, με σκοπό όταν το readcount είναι ίσο με n, να σημαίνει ότι όλοι οι readers έχουν διαβάσει, και ο writer μπορεί να συνεχίσει. Ο reader ολοκληρώνει όταν το shared->written είναι ίσο με M, ή όταν ο feeder στείλει μήνυμα τερματισμού μέσω της shared memory (shared->terminate). Όταν το shared->terminate γίνει 1, πάει να πει ότι ο writer έχει γράψει όσους ακέραιους ήθελε και τα παιδιά μπορούν να τελειώνουν. Έπειτα, κάθε παιδί εμφανίζει τα στατιστικά που ζητήθηκαν στο αρχείο output.txt, δηλαδή ποιους αριθμούς διάβασε και ποιος είναι ο μέσος χρόνος που έκανε για να τους διαβάσει.

ΠΟΡΙΣΜΑΤΑ

Παρατηρώ ότι οι μέσοι χρόνοι των διεργασιών παιδιών δε διαφέρουν πολύ μεταξύ τους, γιατί η ανάγνωση από τη shared memory, επιτρέπεται να γίνει ταυτόχρονα. Η αύξηση του μετρητή όμως, δεν επιτρέπεται να γίνει, για αυτό υπάρχει αυτή η μικρή διαφορά.

Η ΠΙΚΡΗ ΑΛΗΘΕΙΑ

Το θέμα είναι ότι για μικρά M και n όλα δουλεύουν κανονικά και τερματίζει. Όμως για μεγάλο M ή όταν τα M και n έχουν μεγάλη διαφορά μεταξύ τους, το πρόγραμμα κολλάει. Παρακάτω υπάρχουν screenshots με παράδειγμα που δεν κολλάει και με παράδειγμα που κολλάει.

Η γραμμή 51 του `child.c` είναι χρήσιμη, καθώς δείχνει ποιο παιδί διάβασε τι, και γενικά, δείχνει ένα "στιγμιότυπο" εκείνης της στιγμής. Στο παραδοτέο βρίσκεται σε σχόλιο, αλλά στα screenshots θα φαίνεται και το output της για να είναι ξεκάθαρο το τι συμβαίνει. Από εκεί κατάλαβα ότι ανάλογα τον υπολογιστή, κολλάει σε διαφορετικό αριθμό, πχ σε μένα κολλάει αν $M = 26$ και n οποιοσδήποτε ακέριος μικρότερος του 5, αλλά δεν κολλάει όταν $M = 26$ και $n = 5$. Αντίστοιχα αυτό συμβαίνει και για πιο μεγάλα M . Δεν μπορώ να καταλάβω πού και πώς κολλάει. Στο δικό σας υπολογιστή μπορεί να κολλάει σε διαφορετικούς αριθμούς (σίγουρα όμως θα κολλάει για μεγάλο M).

Έπειτα από ΠΑΡΑ ΠΟΛΛΕΣ φορές που το έτρεξα, κατά τύχη μου εμφανίστηκε 1 φορά το error: "Semaphore V(Up): Invalid Argument".

Δυστυχώς δεν έχω καταφέρει έκτοτε να ξαναπετύχω αυτό το output, ακόμα και με ίδιους αριθμούς. Όσο και να έψαξα τι μπορεί να φταίει, έχοντας ως βοήθεια το παραπάνω error, δεν κατάφερα να βρω κάτι.

Παράδειγμα που δε δουλεύει

```
Theodoras-MacBook-Pro:hw3 teo.png$ ./feeder 26 2
Child #4421, num: 65, rc: 1, flag: 0, i is: 0
Child #4420, num: 65, rc: 2, flag: 0, i is: 0
Child #4420, num: 27, rc: 1, flag: 0, i is: 1
Child #4421, num: 27, rc: 2, flag: 0, i is: 1
Child #4420, num: 90, rc: 1, flag: 0, i is: 2
Child #4421, num: 90, rc: 2, flag: 0, i is: 2
Child #4420, num: 13, rc: 1, flag: 0, i is: 3
Child #4421, num: 13, rc: 2, flag: 0, i is: 3
Child #4420, num: 82, rc: 1, flag: 0, i is: 4
Child #4421, num: 82, rc: 2, flag: 0, i is: 4
Child #4421, num: 96, rc: 1, flag: 0, i is: 5
Child #4420, num: 96, rc: 2, flag: 0, i is: 5
Child #4421, num: 44, rc: 1, flag: 0, i is: 6
Child #4420, num: 44, rc: 2, flag: 0, i is: 6
Child #4421, num: 75, rc: 1, flag: 0, i is: 7
Child #4420, num: 75, rc: 2, flag: 0, i is: 7
Child #4420, num: 5, rc: 1, flag: 0, i is: 8
Child #4421, num: 5, rc: 2, flag: 0, i is: 8
Child #4420, num: 64, rc: 1, flag: 0, i is: 9
Child #4421, num: 64, rc: 2, flag: 0, i is: 9
Child #4420, num: 9, rc: 1, flag: 0, i is: 10
Child #4421, num: 9, rc: 2, flag: 0, i is: 10
Child #4420, num: 89, rc: 1, flag: 0, i is: 11
Child #4421, num: 89, rc: 2, flag: 0, i is: 11
Child #4420, num: 85, rc: 1, flag: 0, i is: 12
Child #4421, num: 85, rc: 2, flag: 0, i is: 12
Child #4420, num: 2, rc: 1, flag: 0, i is: 13
Child #4421, num: 2, rc: 2, flag: 0, i is: 13
Child #4420, num: 13, rc: 1, flag: 0, i is: 14
Child #4421, num: 13, rc: 2, flag: 0, i is: 14
Child #4420, num: 25, rc: 1, flag: 0, i is: 15
Child #4421, num: 25, rc: 2, flag: 0, i is: 15
Child #4420, num: 94, rc: 1, flag: 0, i is: 16
Child #4421, num: 94, rc: 2, flag: 0, i is: 16
Child #4420, num: 34, rc: 1, flag: 0, i is: 17
Child #4421, num: 34, rc: 2, flag: 0, i is: 17
Child #4420, num: 81, rc: 1, flag: 0, i is: 18
Child #4421, num: 81, rc: 2, flag: 0, i is: 18
Child #4420, num: 0, rc: 1, flag: 0, i is: 19
Child #4421, num: 0, rc: 2, flag: 0, i is: 19
Child #4420, num: 73, rc: 1, flag: 0, i is: 20
Child #4421, num: 73, rc: 2, flag: 0, i is: 20
Child #4421, num: 81, rc: 1, flag: 0, i is: 21
Child #4420, num: 81, rc: 2, flag: 0, i is: 21
Child #4421, num: 50, rc: 1, flag: 0, i is: 22
Child #4420, num: 50, rc: 2, flag: 0, i is: 22
Child #4421, num: 55, rc: 1, flag: 0, i is: 23
Child #4420, num: 55, rc: 2, flag: 0, i is: 23
Child #4421, num: 36, rc: 1, flag: 0, i is: 24
Child #4420, num: 36, rc: 2, flag: 0, i is: 24
^C
```

Παράδειγμα που δουλεύει

```
Theodoras-MacBook-Pro:hw3 teo.png$ ./feeder 26 5
Child #4498, num: 73, rc: 1, flag: 0, i is: 0
Child #4497, num: 73, rc: 2, flag: 0, i is: 0
Child #4499, num: 73, rc: 3, flag: 0, i is: 0
Child #4496, num: 73, rc: 4, flag: 0, i is: 0
Child #4500, num: 73, rc: 5, flag: 0, i is: 0
Child #4496, num: 90, rc: 1, flag: 0, i is: 1
Child #4497, num: 90, rc: 2, flag: 0, i is: 1
Child #4500, num: 90, rc: 3, flag: 0, i is: 1
Child #4498, num: 90, rc: 4, flag: 0, i is: 1
Child #4499, num: 90, rc: 5, flag: 0, i is: 1
Child #4499, num: 79, rc: 1, flag: 0, i is: 2
Child #4498, num: 79, rc: 2, flag: 0, i is: 2
Child #4497, num: 79, rc: 3, flag: 0, i is: 2
Child #4496, num: 79, rc: 4, flag: 0, i is: 2
Child #4500, num: 79, rc: 5, flag: 0, i is: 2
Child #4499, num: 63, rc: 1, flag: 0, i is: 3
Child #4497, num: 63, rc: 2, flag: 0, i is: 3
Child #4500, num: 63, rc: 3, flag: 0, i is: 3
Child #4496, num: 63, rc: 4, flag: 0, i is: 3
Child #4498, num: 63, rc: 5, flag: 0, i is: 3
Child #4499, num: 30, rc: 1, flag: 0, i is: 4
Child #4497, num: 30, rc: 2, flag: 0, i is: 4
Child #4496, num: 30, rc: 3, flag: 0, i is: 4
Child #4500, num: 30, rc: 4, flag: 0, i is: 4
Child #4498, num: 30, rc: 5, flag: 0, i is: 4
Child #4500, num: 60, rc: 1, flag: 0, i is: 5
Child #4496, num: 60, rc: 2, flag: 0, i is: 5
Child #4499, num: 60, rc: 3, flag: 0, i is: 5
Child #4497, num: 60, rc: 4, flag: 0, i is: 5
Child #4498, num: 60, rc: 5, flag: 0, i is: 5
Child #4499, num: 51, rc: 1, flag: 0, i is: 6
Child #4496, num: 51, rc: 2, flag: 0, i is: 6
Child #4500, num: 51, rc: 3, flag: 0, i is: 6
Child #4498, num: 51, rc: 4, flag: 0, i is: 6
Child #4497, num: 51, rc: 5, flag: 0, i is: 6
Child #4500, num: 48, rc: 1, flag: 0, i is: 7
Child #4496, num: 48, rc: 2, flag: 0, i is: 7
Child #4498, num: 48, rc: 3, flag: 0, i is: 7
Child #4499, num: 48, rc: 4, flag: 0, i is: 7
Child #4497, num: 48, rc: 5, flag: 0, i is: 7
Child #4500, num: 11, rc: 1, flag: 0, i is: 8
Child #4496, num: 11, rc: 2, flag: 0, i is: 8
Child #4498, num: 11, rc: 3, flag: 0, i is: 8
Child #4499, num: 11, rc: 4, flag: 0, i is: 8
Child #4497, num: 11, rc: 5, flag: 0, i is: 8
Child #4499, num: 62, rc: 1, flag: 0, i is: 9
Child #4497, num: 62, rc: 2, flag: 0, i is: 9
Child #4496, num: 62, rc: 3, flag: 0, i is: 9
Child #4500, num: 62, rc: 4, flag: 0, i is: 9
Child #4498, num: 62, rc: 5, flag: 0, i is: 9
Child #4496, num: 84, rc: 1, flag: 0, i is: 10
Child #4499, num: 84, rc: 2, flag: 0, i is: 10
Child #4497, num: 84, rc: 3, flag: 0, i is: 10
Child #4498, num: 84, rc: 4, flag: 0, i is: 10
Child #4500, num: 84, rc: 5, flag: 0, i is: 10
Child #4500, num: 71, rc: 1, flag: 0, i is: 11
Child #4497, num: 71, rc: 2, flag: 0, i is: 11
Child #4499, num: 71, rc: 3, flag: 0, i is: 11
Child #4496, num: 71, rc: 4, flag: 0, i is: 11
Child #4498, num: 71, rc: 5, flag: 0, i is: 11
Child #4500, num: 93, rc: 1, flag: 0, i is: 12
Child #4499, num: 93, rc: 2, flag: 0, i is: 12
Child #4498, num: 93, rc: 3, flag: 0, i is: 12
Child #4497, num: 93, rc: 4, flag: 0, i is: 12
Child #4496, num: 93, rc: 5, flag: 0, i is: 12
Child #4500, num: 15, rc: 1, flag: 0, i is: 13
Child #4497, num: 15, rc: 2, flag: 0, i is: 13
Child #4499, num: 15, rc: 3, flag: 0, i is: 13
Child #4496, num: 15, rc: 4, flag: 0, i is: 13
Child #4498, num: 65, rc: 1, flag: 0, i is: 14
Child #4499, num: 65, rc: 2, flag: 0, i is: 14
Child #4497, num: 65, rc: 3, flag: 0, i is: 14
Child #4500, num: 65, rc: 4, flag: 0, i is: 14
Child #4497, num: 65, rc: 5, flag: 0, i is: 14
Child #4499, num: 85, rc: 1, flag: 0, i is: 15
Child #4498, num: 85, rc: 2, flag: 0, i is: 15
Child #4497, num: 85, rc: 3, flag: 0, i is: 15
Child #4496, num: 85, rc: 4, flag: 0, i is: 15
Child #4499, num: 85, rc: 5, flag: 0, i is: 15
Child #4499, num: 23, rc: 1, flag: 0, i is: 16
Child #4496, num: 23, rc: 2, flag: 0, i is: 16
Child #4497, num: 23, rc: 3, flag: 0, i is: 16
Child #4497, num: 23, rc: 4, flag: 0, i is: 16
Child #4500, num: 23, rc: 5, flag: 0, i is: 16
Child #4497, num: 63, rc: 1, flag: 0, i is: 17
Child #4500, num: 63, rc: 2, flag: 0, i is: 17
Child #4496, num: 63, rc: 3, flag: 0, i is: 17
Child #4498, num: 63, rc: 4, flag: 0, i is: 17
Child #4499, num: 63, rc: 5, flag: 0, i is: 17
Child #4499, num: 56, rc: 1, flag: 0, i is: 18
Child #4497, num: 56, rc: 2, flag: 0, i is: 18
Child #4498, num: 56, rc: 3, flag: 0, i is: 18
Child #4500, num: 56, rc: 4, flag: 0, i is: 18
Child #4496, num: 56, rc: 5, flag: 0, i is: 18
Child #4498, num: 5, rc: 1, flag: 0, i is: 19
Child #4496, num: 5, rc: 2, flag: 0, i is: 19
Child #4499, num: 5, rc: 3, flag: 0, i is: 19
Child #4500, num: 5, rc: 4, flag: 0, i is: 19
Child #4497, num: 5, rc: 5, flag: 0, i is: 19
Child #4497, num: 85, rc: 1, flag: 0, i is: 20
Child #4496, num: 85, rc: 2, flag: 0, i is: 20
Child #4500, num: 85, rc: 3, flag: 0, i is: 20
Child #4499, num: 85, rc: 4, flag: 0, i is: 20
Child #4498, num: 85, rc: 5, flag: 0, i is: 20
Child #4500, num: 64, rc: 1, flag: 0, i is: 21
Child #4498, num: 64, rc: 2, flag: 0, i is: 21
Child #4497, num: 64, rc: 3, flag: 0, i is: 21
Child #4499, num: 64, rc: 4, flag: 0, i is: 21
Child #4496, num: 64, rc: 5, flag: 0, i is: 21
Child #4497, num: 51, rc: 1, flag: 0, i is: 22
Child #4500, num: 51, rc: 2, flag: 0, i is: 22
Child #4496, num: 51, rc: 3, flag: 0, i is: 22
Child #4499, num: 51, rc: 4, flag: 0, i is: 22
Child #4498, num: 51, rc: 5, flag: 0, i is: 22
Child #4499, num: 65, rc: 1, flag: 0, i is: 23
Child #4497, num: 65, rc: 2, flag: 0, i is: 23
Child #4498, num: 65, rc: 3, flag: 0, i is: 23
Child #4496, num: 65, rc: 4, flag: 0, i is: 23
Child #4500, num: 65, rc: 5, flag: 0, i is: 23
Child #4499, num: 42, rc: 1, flag: 0, i is: 24
Child #4500, num: 42, rc: 2, flag: 0, i is: 24
Child #4496, num: 42, rc: 3, flag: 0, i is: 24
Child #4498, num: 42, rc: 4, flag: 0, i is: 24
Child #4497, num: 42, rc: 5, flag: 0, i is: 24
Child #4496, num: 52, rc: 1, flag: 0, i is: 25
Child #4500, num: 52, rc: 2, flag: 0, i is: 25
Child #4498, num: 52, rc: 3, flag: 0, i is: 25
PID: 4496 has Running Average: 4 milliseconds
Child #4497, num: 52, rc: 4, flag: 0, i is: 25
Child #4499, num: 52, rc: 5, flag: 0, i is: 25
PID: 4500 has Running Average: 4 milliseconds
PID: 4497 has Running Average: 4 milliseconds
PID: 4498 has Running Average: 4 milliseconds
PID: 4499 has Running Average: 4 milliseconds
Child #0 with pid 4496 has terminated.
Child #3 with pid 4499 has terminated.
Child #1 with pid 4497 has terminated.
Child #4 with pid 4500 has terminated.
Child #2 with pid 4498 has terminated.
All are dead. Me dies now
```