

Λειτουργικά εργασία 1

sdi1400294

Παναγιώτης Μαντάς

Μεταγλώττιση

Η άσκηση είναι γραμμένη σε C και συμπεριλαμβάνει makefile. Δεν χρειάζεται κάποιες ειδικές επικεφαλίδες πέρα από αυτά που μάθαμε στο φροντιστήριο.

Εκτέλεση

Παράδειγμα εκτέλεσης:

```
"${OUTPUT_PATH}" lorem.txt 10 1000 200
```

Οι παράμετροι είναι:

1. Το αρχείο κειμένου
2. Το πλήθος παιδιών
3. Οι επαναλήψεις
4. Το πλήθος γραμμών για κάθε τμήμα του αρχείου

Η άσκηση έχει δοκιμαστεί σε Ubuntu 18.04 και στους υπολογιστές της σχολής για αρχεία μεγαλύτερα των 1000 γραμμών.

Το μέγιστο πλήθος γραμμάτων ανά γραμμή είναι 64000 και το ελάχιστο πλήθος γραμμών είναι 1000.

Υλοποίηση

Γενικά έχουν υλοποιηθεί όλες οι λεπτομέρειες της εφαρμογής (εγγραφή σε log, τυχαίες κατανομές ώστε να γίνεται συχνότερα επανεπιλογή του εκάστοτε τμήματος που ζητά το κάθε παιδί κλπ).

Προσέγγιση λύσης επικοινωνίας

Ο τρόπος υλοποίησης της άσκησης γίνεται με ένα σύστημα producer/consumer με τόσες ουρές όσες είναι τα παιδιά συν μία επιπλέον.

Η μία ουρά έχει κατεύθυνση από τα *παιδιά* προς το *γονιό* όπου έχει υλοποιηθεί ως ένα κλασικό πρόβλημα παραγωγού καταναλωτή με πολλούς παραγωγούς (τις διεργασίες παιδιά) και έναν καταναλωτή (τη διεργασία γονιό). Μέσω αυτής της ουράς τα παιδιά στέλνουν αιτήσεις στο γονιό. Κάθε παιδί έχει ένα μοναδικό αναγνωριστικό και στέλνει αιτήσεις που συμπεριλαμβάνουν το αναγνωριστικό του, τον αριθμό τμήματος και τον αριθμό γραμμής. Στην ουρά αυτή έμμεσα γίνεται σειριοποίηση των αιτήσεων καθώς ο γονιός δεν έχει νήματα άρα μπορεί να καταναλώνει μία αίτηση κάθε φορά.

Οι υπόλοιπες ουρές (1 για κάθε παιδί) έχουν αντίστροφη κατεύθυνση. Παραγωγός είναι ο γονιός και καταναλωτής σε κάθε ουρά είναι το εκάστοτε παιδί. Αν ο γονιός κρίνει ότι ένα παιδί είναι έτοιμο να εξυπηρετηθεί τότε γράφει στην αντίστοιχη ουρά την αντίστοιχη γραμμή. Στις ουρές αυτές υπάρχει απόλυτη παραλληλία καθώς τα παιδιά επιτρέπεται ταυτόχρονα να διαβάζουν (το κάθε ένα στη δική του ουρά) διαφορετικές ή ίδιες γραμμές του ίδιου τμήματος του αρχείου. Για τις ουρές έχουν χρησιμοποιηθεί αντίστοιχοι σημαφόροι όπως το πρόβλημα παραγωγού καταναλωτή.

Το ενεργό τμήμα του αρχείου αποθηκεύεται καθαρά στη μνήμη του γονιού (στο heap) και στις ενδιάμεσες ουρές αποθηκεύεται μόνο η κατάλληλη γραμμή που ζητά το παιδί.

Η παραλληλία επιτυγχάνεται ως εξής: Ο γονιός έχει τρεις πίνακες με τόσες θέσεις όσα τα παιδιά:

1. **segments_read**: πίνακας που κρατάμε ποιο παιδί διαβάσει ποιο τμήμα
2. **segments_desired**: πίνακας που κρατάμε ποιο παιδί θέλει να διαβάσει ποιο τμήμα
3. **linenumbers_desired**: πίνακας που κρατάμε ποιο παιδί θέλει να διαβάσει ποια γραμμή

Όταν έρχεται μία αίτηση εκτελείται ο ακόλουθος αλγόριθμος:

1. Αν δεν υπάρχει κανένα τμήμα ήδη στη μνήμη, τότε φορτώνουμε το τμήμα από το δίσκο, αντιγράφουμε τη γραμμή στον αντίστοιχο εξερχόμενο buffer και ανεβάζουμε τον αντίστοιχο σημαφόρο ώστε να πάει το παιδί να διαβάσει την αντίστοιχη γραμμή. Καταγράφουμε στον πίνακα **segments_read** ότι το παιδί θα διαβάσει αυτό το τμήμα.
2. Αν υπάρχει τμήμα και έχει ίδιο ID με αυτό που ζητά το παιδί τότε κάνουμε το ίδιο με το βήμα 1 (χωρίς την ανάγνωση του τμήματος από το δίσκο). Με αυτόν τον τρόπο τα παιδιά μπορούν παράλληλα να διαβάζουν διαφορετικές γραμμές του τμήματος.
3. Αν το παιδί ζητά τμήμα που δεν υπάρχει στη μνήμη τότε κοιτάμε αν στον πίνακα **segments_read** υπάρχει κάποιο παιδί που διαβάσει τώρα αυτό το τμήμα.
 1. Αν όχι τότε κάνουμε το ίδιο με το βήμα 1 (διαβάζουμε το νέο τμήμα στη μνήμη κλπ)
 2. Αν ναι τότε βάζουμε αυτό το παιδί σε αναμονή (ουσιαστικά δεν κάνουμε up τον αντίστοιχο σημαφόρο) και καταγράφουμε στο **segments_desired** και **linenumbers_desired** ότι εκκρεμεί αυτή η αίτηση για εξυπηρέτηση.
4. Όταν ένα παιδί διαβάσει τη γραμμή στέλνει μία αίτηση στο γονιό ότι τελείωσε την ανάγνωση. Οπότε κοιτάμε το **segments_desired** για να δούμε αν υπάρχει κάποιος που θέλει να εξυπηρετηθεί. Αν δεν υπάρχει τότε απλά περιμένουμε για την επόμενη αίτηση. Αν υπάρχει τότε:
 1. Βρίσκουμε στον πίνακα την πρώτη αίτηση που εκκρεμεί
 2. Φορτώνουμε το τμήμα στη μνήμη
 3. Φορτώνουμε τη γραμμή στον αντίστοιχο buffer.
 4. Αναζητούμε στον πίνακα όλες τις διεργασίες που θέλουν το ίδιο τμήμα και επαναλαμβάνουμε τα βήματα 4.1, 4.2, 4.3 για κάθε τέτοιο παιδί (ώστε να διαβάζουν ταυτόχρονα το νέο τμήμα).