

## Σειρά Εργασιών 1

**1.1 FIFO pipes**

Υλοποιήστε μια «βιβλιοθήκη» αγωγών FIFO μιας κατεύθυνσης, με την παρακάτω διεπαφή προγραμματισμού:

<code>int pipe_open(int size);</code>	Δημιουργία και άνοιγμα αγωγού για γράψιμο και διάβασμα. Επιστρέφει το αναγνωριστικό του αγωγού που δημιουργήθηκε (θετικός ακέραιος).
<code>int pipe_write(int p, char c);</code>	Γράψιμο ενός byte στον αγωγό p. Επιστρέφει 1 για επιτυχία, ή -1 αν δεν υπάρχει ανοιχτός αγωγός για γράψιμο με αυτό το αναγνωριστικό.
<code>int pipe_writeDone(int p);</code>	Κλείνει τον αγωγό p για γράψιμο. Επιστρέφει 1 για επιτυχία ή -1 αν δεν υπάρχει ανοιχτός αγωγός για γράψιμο με αυτό το αναγνωριστικό.
<code>int pipe_read(int p, char *c);</code>	Απομάκρυνση ενός byte από τον αγωγό p. Επιστρέφει 1 αν η ανάγνωση είναι επιτυχής, 0 αν ο αγωγός είναι άδειος και κλειστός για γράψιμο, ή -1 αν δεν υπάρχει ανοιχτός αγωγός για διάβασμα με αυτό το αναγνωριστικό. Αν επιστραφεί 0 ο αγωγός κλείνει και καταστρέφεται αυτομάτως.

Υποθέστε ότι μόνο ένα νήμα καλεί την read και μόνο ένα νήμα καλεί την write ανά πάσα στιγμή, όμως η read και write μπορεί να καλούνται ταυτόχρονα. Βασίστε την υλοποίηση στην τεχνική της κυκλικής αποθήκης (ring buffer) προσπαθώντας να αποφύγετε συνθήκες ανταγωνισμού. Το μπλοκάρισμα, όπου/όταν χρειάζεται, πρέπει να γίνεται με ενεργή αναμονή. Ελέγξτε την υλοποίηση σας μέσω ενός προγράμματος που παίρνει ως όρισμα το όνομα ενός αρχείου, δημιουργεί δύο νήματα και δύο αγωγούς μεγέθους 64 bytes. Ο πρώτος αγωγός χρησιμοποιείται για τη μεταφορά των περιεχομένων του αρχείου από το ένα νήμα στο άλλο, που θα σώζει τα δεδομένα σε ένα νέο αρχείο (αντίγραφο) με το ίδιο όνομα και κατάληξη “.copy”. Ο δεύτερος αγωγός χρησιμοποιείται για την μεταφορά των περιεχομένων του αντιγράφου προς την αντίθετη κατεύθυνση. Η δεύτερη μεταφορά αρχίζει αφού ολοκληρωθεί η πρώτη, και τα δεδομένα αποθηκεύονται σε ένα δεύτερο αντίγραφο με την κατάληξη “.copy2”.

**1.2 Αναγνώριση πρώτων αριθμών**

Υλοποιήστε ένα πρόγραμμα που διαβάσει επαναληπτικά από την είσοδό του ακέραιες τιμές, για κάθε μια από τις οποίες υπολογίζει αν αντιστοιχεί σε πρώτο αριθμό και εκτυπώνει 1 ή 0 αντίστοιχα. Για να επιταχυνθεί ο υπολογισμός, το πρόγραμμα χρησιμοποιεί N νήματα, στο πνεύμα του παρακάτω ψευτοκώδικα (N όρισμα του προγράμματος):

main thread:	worker thread:
<pre>create N workers while (input exists) {     read next value     wait for a worker to become available     notify the worker to process the value } notify workers to terminate wait for all workers to terminate</pre>	<pre>while (1) {     notify main that I am available     wait for notification by main     if notified to terminate, break     else process assigned value } notify main that I will terminate</pre>

Χρησιμοποιήστε όποιον αλγόριθμο εντοπισμού πρώτων αριθμών επιθυμείτε. Αναλύστε την απόδοση ως συνάρτηση του N και των τιμών που δέχεται το πρόγραμμα. Τρέξτε μετρήσεις για μικρούς και μεγάλους πρώτους αριθμούς.

**1.3 External mergesort**

Υλοποιήστε μια παράλληλη έκδοση του external mergesort με νήματα. Σε κάθε επίπεδο αναδρομής, το τρέχον νήμα χωρίζει το τμήμα του αρχείου που του ανατίθεται στα δύο, αναθέτει την ταξινόμηση κάθε τμήματος σε ένα ξεχωριστό νήμα που δημιουργεί για αυτό τον σκοπό, περιμένει να τερματίσουν και τα δύο νήματα, συγχωνεύει τα δύο τμήματα σε ένα ενιαίο ταξινομημένο τμήμα (χωρίς απαραίτητα να χρησιμοποιείται κάποιο προσωρινό ενδιάμεσο αρχείο), και επιστρέφει. Αν το τμήμα που απομένει είναι αρκετά μικρό (64 ακέραιες τιμές), τα περιεχόμενα ταξινομούνται στην μνήμη (χρησιμοποιήστε όποια μέθοδο επιθυμείτε). Τα νήματα και οι μεταβλητές που χρησιμοποιούνται για τον συγχρονισμό τους πρέπει να δημιουργούνται και να καταστρέφονται δυναμικά κατά την αναδρομή. Δοκιμάστε την υλοποίηση σας μέσω ενός προγράμματος που ταξινομεί τα περιεχόμενα ενός δυαδικού αρχείου με ακέραιες τιμές, το όνομα του οποίου λαμβάνει ως όρισμα.

Η υλοποίηση πρέπει να γίνει σε C με χρήση της βιβλιοθήκης pthreads. Ο συγχρονισμός μεταξύ των νημάτων πρέπει να υλοποιηθεί με **απλές μεταβλητές χωρίς** την χρήση μηχανισμών συγχρονισμού των pthreads ή του λειτουργικού.

**Παράδοση:** Σάββατο 18 Νοεμβρίου 2023, 23:59