

Σειρά Εργασιών 1**1.1 FIFO pipes**

Υλοποιήστε μια «βιβλιοθήκη» που παρέχει αγωγούς FIFO μιας κατεύθυνσης (που μπορεί να χρησιμοποιηθούν από την εφαρμογή για την επικοινωνία ανάμεσα σε δύο νήματα), 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 γράφει στον αγωγό και μόνο ένα νήμα διαβάζει από τον αγωγό ανά πάσα στιγμή. Βασίστε την υλοποίηση στην τεχνική της κυκλικής αποθήκης (ring buffer) προσπαθώντας να αποφύγετε συνθήκες ανταγωνισμού κατά την ταυτόχρονη εκτέλεση των παραπάνω λειτουργιών. Το μπλοκάρισμα, όπου/όταν χρειάζεται, πρέπει να γίνεται με ενεργή αναμονή. Ελέγξτε την υλοποίηση σας μέσω μιας εφαρμογής όπου δύο νήματα επικοινωνούν μεταξύ τους μέσω δύο αγωγών μεγέθους 64 bytes, ο πρώτος εκ των οποίων χρησιμοποιείται για τη μεταφορά ενός (μεγάλου) αρχείου από το ένα νήμα στο άλλο (τα περιεχόμενα θα σώζονται σε ξεχωριστό αντίγραφο), και ο δεύτερος για την μεταφορά του αντιγράφου προς την ανάποδη κατεύθυνση (η δεύτερη μεταφορά πρέπει να αρχίζει αφού ολοκληρωθεί η πρώτη, και τα περιεχόμενα να αποθηκεύονται σε ένα δεύτερο ξεχωριστό αντίγραφο).

1.2 Αναγνώριση πρώτων αριθμών

Υλοποιήστε ένα πρόγραμμα που διαβάζει 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 πρέπει να χρησιμοποιεί νήματα στο πνεύμα του παρακάτω ψευδοκώδικα (N όρισμα του προγράμματος):

<pre>main thread: create N workers while (job exists) { wait for a worker to become available read next value & assign it to an available worker notify the worker to process the value } wait for all workers to become available notify workers to terminate wait for all workers to terminate</pre>	<pre>worker thread: 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>
---	--

Αναλύστε την απόδοση ως συνάρτηση του αριθμού των νημάτων και των τιμών εισόδου που δέχεται το πρόγραμμα.

1.3 External mergesort

Υλοποιήστε μια παράλληλη έκδοση του external mergesort με νήματα. Σε κάθε επίπεδο αναδρομής, ένα νήμα (αρχικά το κυρίως νήμα) σπάει το τμήμα του αρχείου που του ανατίθεται σε δύο τμήματα, ανατεθεί 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 ένα προσωρινό ενδιάμεσο αρχείο), και επιστρέφει. Αν το τμήμα του αρχείου είναι αρκετά μικρό (64 ακέραιες τιμές), το νήμα ταξινομεί τα περιεχόμενα στην μνήμη μέσω του απλού mergesort. Τα επιπλέον 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 binary 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 κοινές μεταβλητές χωρίς** την χρήση κάποιου μηχανισμού συγχρονισμού των pthreads ή του λειτουργικού.

Παράδοση: Σάββατο 6 Νοεμβρίου 2021, 23:59