

Η εργασία αποτελείται απ' τα αρχεία: parent.c και child.c.

Το αρχείο parent.c δέχεται το αρχείο shakespeare.txt ως όρισμα, το οποίο είναι 1000 γραμμές. Δέχεται, επίσης, ως παράμετρο τον βαθμό κατάτμησης του αρχείου (partition), το πλήθος των παιδιών (children\_num) και το πλήθος των αιτήσεων που θα κάνει κάθε παιδί (request\_num).

### Μεταγλώττιση κι εκτέλεση

Για τη μεταγλώττιση του κώδικα, χρησιμοποιούνται οι εξής εντολές:

```
gcc child.c -lpthread -o child
```

```
gcc parent.c -lpthread -o parent
```

Για την εκτέλεση, χρησιμοποιείται η εντολή:

```
./parent shakespeare.txt 100 1 1
```

### Parent.c

Στο αρχείο αυτό δημιουργείται ένα struct shared\_vars με τις διαμοιραζόμενες μεταβλητές που χρειάζονται για την επικοινωνία γονέα και παιδιών.

Αρχικά το πρόγραμμα επιβεβαιώνει ότι έχει δεχθεί τα απαραίτητα ορίσματα.

Μετράει το πλήθος των γραμμών του text file και τον αριθμό των segments (segments\_num) που θα προκύψουν απ' την κατάτμηση του.

Στη συνέχεια δημιουργούνται οι απαραίτητοι σημαφόροι, οι οποίοι είναι named POSIX. Δημιουργείται δυναμικά ένας πίνακας ("semaphores") από σημαφόρους, με πλήθος ίσο με το segments\_num, και οι sem1, sem2, readwrite\_mutex\_sem, , sem\_completed\_requests, εκ των οποίων οι sem2 και readwrite\_mutex\_sem αρχικοποιούνται στο 0.

Δημιουργούνται shared memory segments και τα κλειδιά τους για το κείμενο που θα ζητηθεί (data), για το struct με τις διαμοιραζόμενες μεταβλητές και για πίνακα ακεραίων στον οποίο θα καταγράφεται το πλήθος των readers για κάθε segment. Γίνεται attach και μετατροπή των κλειδιών σε string προκειμένου να περαστούν στη συνέχεια στο child.c.

Δημιουργούνται τα παιδιά και ελέγχεται το pid - αν είναι 0 τότε το process είναι child, διαφορετικά πρόκειται για τη γονική διεργασία. Αν είναι παιδί, γίνεται execp του εκτελέσιμου child, στο οποίο περνούν τα προαναφερθέντα strings του struct και του data.

Αν πρόκειται για τη διεργασία του γονέα, για όσο διάστημα δεν έχουν πραγματοποιηθεί όλες οι αιτήσεις των παιδιών, με χρήση σημαφόρων,

διαβάζει το `segment_requested` που έχει καταγραφεί απ' το παιδί, γίνεται η σχετική αναζήτηση, λαμβάνοντας υπόψιν ότι η αρίθμηση των `segments` ξεκινά απ' το 0. Ο σημαφόρος `sem2` καθώς είναι αρχικοποιημένος στο 0 συνεπάγεται την αναμονή του γονέα μέχρι να το ενεργοποιήσει κάποιο παιδί. Αφού φέρει στη μνήμη το ζητούμενο `segment`, γίνεται `sem_post` του `readwrite_mutex_sem`, ο οποίος στην ουσία ενημερώνει το παιδί ότι έχει ολοκληρωθεί η εργασία που ζήτησε.

Όταν ολοκληρωθούν όλες οι αιτήσεις, ο γονέας περιμένει την επιστροφή των παιδιών και στο τέλος κλείνει και αποσυνδέει τους σημαφόρους, τη διαμοιραζόμενη μνήμη και αποδεσμεύει τη μνήμη που είχε δυναμικά δεσμευτεί.

## Child.c

Η διεργασία αυτή δέχεται τα κλειδιά του `data`, του `struct shared_vars` και του πίνακα `readers_count`, τα μετατρέπει σε `integers` και με αυτά συνδέεται με τη διαμοιραζόμενη μνήμη. Ανοίγει τους σημαφόρους μέσω των ονομάτων τους.

Επιλέγει έναν τυχαίο αριθμό για το `segment` που θα ζητήσει, δημιουργεί το δικό του `log file`, και για όσο είναι το πλήθος των αιτήσεων που έχει οριστεί να πραγματοποιήσει επαναλαμβάνει την εξής διαδικασία:

Διαλέγει μια τυχαία γραμμή απ' το 1 έως το 10. Κάνει `sem_wait` τον σημαφόρο που αντιστοιχεί στο `segment` που ζήτησε. Το πρώτο παιδί που θα καταφέρει να το κάνει, αφού αυξήσει το `readers_count[i]` και ελέγξει ότι είναι το πρώτο παιδί (`readers_count == 1`), δεσμεύει τον χώρο κάνοντας `sem_wait(sem1)`, γράφει στη μνήμη το `segment_requested`, ενημερώνει τον γονέα μέσω `sem_post(sem2)` και αναμένει την απάντηση μέσω `sem_wait(readwrite_mutex_sem)`. Αφού λάβει απάντηση, κάνει `sem_post(semaphores[i])` επιτρέποντας έτσι την είσοδο και σε άλλο παιδί, το οποίο θα παρακάμψει την επικοινωνία με τον γονέα και θα ζητήσει κατευθείαν τη γραμμή που θέλει απ' τη μνήμη, καταγράφοντάς την στο αρχείο που είχε νωρίτερα δημιουργήσει. Στο τέλος, κάθε παιδί με τη χρήση του `semaphores[i]` ενημερώνει τον πίνακα `readers_count[i]` για την αποχώρησή του και όταν `readers_count[i]==0` (δηλαδή δεν υπάρχει άλλο παιδί που να ενδιαφέρεται για το συγκεκριμένο `segment`), γίνεται `sem_post(sem1)`, το οποίο επιτρέπει σε άλλο παιδί που έχει εκδηλώσει ενδιαφέρον για διαφορετικό `segment`, και βρισκόταν σε αναμονή, να ξεκινήσει την επικοινωνία με τον γονέα.

Με την ολοκλήρωση κάθε αίτησης γίνεται αύξηση της μεταβλητής `completed_requests`, προκειμένου ο γονέας να γνωρίζει τότε θα χρειαστεί να τερματίσει τον βρόχο του.