

# ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

## REPORT ΕΡΓΑΣΙΑΣ 1

Ονοματεπώνυμο: Θωμάϊς Σάντα

Αριθμός Μητρώου: 1115201800171

Εντολές ώστε να εκτελεστεί ο κώδικας της εργασίας

**make**

**./project1 pink\_floyd\_lyrics.txt K N**

K: ο αριθμός των παιδιών

N: το πλήθος των αιτημάτων που θα ζητήσει το κάθε παιδί

Η παρούσα εργασία περιλαμβάνει τα αρχεία child.c , child.h, parent.c, parent.h, main.c, shared\_memory.h, pink\_floyd\_lyrics.txt και Makefile, για τα οποία παρακάτω παρέχεται επεξήγηση για τη λειτουργία του κώδικα κάθε αρχείου.

Αρχικά, το πρόγραμμα δέχεται ορίσματα από το τερματικό ένα αρχείο κειμένου, τον αριθμό των παιδιών (child process) που θα πραγματοποιήσουν αιτήματα (K) και το πλήθος των αιτημάτων που θα στείλει το κάθε παιδί (N) προς τη διαδικασία γονέα (parent process). Το αίτημα που στέλνεται από τη διαδικασία παιδί είναι ο αριθμός της γραμμής από το αρχείο txt που δόθηκε από το τερματικό σαν όρισμα. Η διαδικασία γονέας θα ψάξει το αρχείο για να επιστρέψει την αιτηθείσα γραμμή στην child process. Για την υλοποίηση του παραπάνω σεναρίου έχει διαμορφωθεί κατάλληλα ο χώρος της διαμοιραζόμενης μνήμης, στην οποία έχουν πρόσβαση τόσο η child process όσο και η parent process. Ο χώρος της διαμοιραζόμενης μνήμης υπάρχει στο header file shared\_memory.h. Μετά το πέρας των αιτημάτων από κάθε παιδί, εκτυπώνεται στην οθόνη και ο χρόνος που χρειάστηκε κάθε παιδί ώστε να ολοκληρώσει τα αιτήματά του.

Για το ζητούμενο της εργασίας έχει υλοποιηθεί το πρότυπο producer-consumer που αναφέρεται και στο βιβλίο του μαθήματος στην ενότητα 7.1 του Abraham Silberschatz, όπου στην περίπτωση της εργασίας έχουμε έναν consumer (parent process) και K producers (child process).

Αναλυτική περιγραφή του κάθε αρχείου:

### **shared\_memory.h**

Στο συγκεκριμένο αρχείο επικεφαλίδας γίνονται define το LINE SIZE σε 101, όπου αναπαριστά το μέγιστο μήκος της γραμμής εντός του αρχείου κειμένου (100 χαρακτήρες + 1 για το \n) και το MAX\_REQUESTS όπου αναπαριστά το μέγιστο συνολικό πλήθος αιτημάτων από όλα τα παιδιά που είναι δυνατόν να γίνουν ( $K * N$ ). Έχει δηλωθεί  $MAX\_REQUESTS = 10000$ . Επίσης στο αρχείο επικεφαλίδας shared\_memory.h περιέχονται τα τμήματα που αποτελούν την

διαμοιραζόμενη μνήμη. Η διαμοιραζόμενη μνήμη θα αποτελείται από 2 τμήματα.

Το πρώτο κομμάτι της διαμοιραζόμενης μνήμης είναι τύπου struct request. Το struct αυτό περιλαμβάνει έναν buffer με αριθμό θέσεων MAX\_REQUESTS που περιέχει struct request\_item. Το struct request\_item περιέχει τον αριθμό του παιδιού (child\_i) και τον αριθμό της γραμμής που ζητάει (number\_of\_line). Εν συνεχεία το struct request, περιέχει 5 σημαφόρους. Ο empty που αρχικοποιείται στον αριθμό των παιδιών, ο full στο 0, ο mutex στο 1 (για αμοιβαίο αποκλεισμό), ο lock στο 1 για αμοιβαίο αποκλεισμό στις εντολές printf και ο my\_turn στο 1 ώστε να επιτυγχάνεται κατάλληλα ο συγχρονισμός μεταξύ της διαργασίας παιδί με τη διεργασία γονέα. Επίσης, περιέχεται και ο αριθμός των αιτημάτων που γίνονται (num\_of\_request).

Το δεύτερο κομμάτι της διαμοιραζόμενης μνήμης που αφορά στην απάντηση του αιτήματος και είναι τύπου struct response. Το struct αυτό περιέχει έναν πίνακα χαρακτήρων μεγέθους LINE\_SIZE όπου θα γραφτεί η γραμμή που θα επιστρέψει ο γονέας στο αίτημα ενός παιδιού. Επίσης, έναν σημαφόρο τον buff\_sem (αρχικοποιημένο στο 0) που το παιδί τον κάνει down καθώς περιμένει να ικανοποιηθεί το αίτημά του και ο γονέας τον κάνει up μόλις βρει την γραμμή που ζητήθηκε ψάχνοντας το αρχείο κειμένου. Τέλος, περιλαμβάνονται 3 ακέραιοι, για τον αριθμό της απάντησης (num\_of\_response), για τον αριθμό του παιδιού στο οποίο θα απαντήσει ο γονέας (child\_res) και για τον αριθμό της γραμμής που ζήτησε το παιδί (line\_you\_asked).

### main.c

Αρχικά πραγματοποιείται έλεγχος για τον αν έχει δοθεί σωστό πλήθος ορισμάτων από το τερματικό (4 ορίσματα καθώς συμπεριλαμβάνεται και το όνομα του εκτελέσιμου αρχείου) και έλεγχος ώστε το γινόμενο των παιδιών με τα αιτήματα να μην υπερβαίνουν το MAX\_REQUESTS (στην συγκεκριμένη περίπτωση έχει γίνει defined σε 10.000).

Στη συνέχεια βρίσκεται ο αριθμός των γραμμών του txt file και ο αριθμός αυτός εκχωρείται στη μεταβλητή lines (αν δοθεί όρισμα το pink\_floyd\_lyrics.txt τότε lines=85).

Δημιουργείται το τμήμα διαμοιραζόμενης μνήμης και γίνεται attach με τις συναρτήσεις shmget και shmat, μαζί με τον κατάλληλο έλεγχο σε περίπτωση που αποτύχει κάποια συνάρτηση. Η διαμοιραζόμενη μνήμη αποτελείται από 2 κομμάτια, ένα τμήμα για το αίτημα και ένα για την απόκριση του αιτήματος. Για το λόγο αυτό οι συναρτήσεις shmget και shmat εκτελούνται 2 φορές η κάθε μία.

Έπειτα, αρχικοποιούνται οι σημαφόροι με τη συνάρτηση sem\_init. Οι τιμές με τις οποίες αρχικοποιούνται έχουν αναφερθεί παραπάνω στην επεξήγηση του αρχείου shared\_memory.h

Στη συνέχεια, εκτελείται μια επανάληψη ώστε να δημιουργηθούν τα K παιδιά, όπου η συνάρτηση fork() επιστρέφει το process id (pid) το οποίο κρατείται στον πίνακα pid[ ] που έχει K θέσεις.

Σε περίπτωση που το pid είναι 0, τότε αναφερόμαστε στην διαδικασία παδί. Μέσα στη διαδικασία παιδί εφαρμόζεται η srand με feed το άθροισμα getpid() (συνάρτηση που επιστρέφει το process id της καλούσας διαδικασίας) και τη συνάρτηση time(NULL), ώστε να είναι βέβαιο ότι κάθε παιδί σε κάθε του αίτημα θα ζητάει και διαφορετική γραμμή από το αρχείο κειμένου. Ύστερα, υλοποιείται μια επανάληψη N φορές, όσες δηλαδή είναι οι αιτήσεις που θα πραγματοποιήσει κάθε παιδί. Μέσα στην επανάληψη αυτή καλείται η συνάρτηση που υλοποιεί την child process (ο πηγαίος κώδικας της βρίσκεται στο αρχείο child.c το οποίο θα εξηγηθεί παρακάτω για τη λειτουργία της). Η συνάρτηση child επιστρέφει τον χρόνο που χρειάστηκε για να εκτελεστεί η παρούσα συνάρτηση, εκχωρείται στη μεταβλητή total\_time τύπου double. Μόλις το συγκεκριμένο παιδί ολοκληρώσει τα N αιτήματα, ο χρόνος διαιρείται με τον αριθμό των αιτημάτων, καθώς στην εκφώνηση της εργασίας ζητείται ο μέσος χρόνος ο οποίος παρέρχεται από την υποβολή ενός αιτήματος μέχρι τη λήψη της αντίστοιχης απάντησης. Τέλος, γίνεται αμοιβαίος αποκλεισμός ώστε να εκτυπωθεί ο μέσος χρόνος εκτέλεσης του κάθε παιδιού (ο σημαφόρος lock γίνεται down (sem\_wait) πριν την εντολή printf και up (sem\_post) μετά την εντολή printf).

Σε περίπτωση που το pid δεν είναι 0, τότε αναφερόμαστε στην γονεϊκή διαδικασία, η οποία είναι έξω από την επανάληψη που δημιουργεί τα παιδιά και εκτελείται K \* N φορές, δηλαδή για όλα τα αιτήματα όλων των παιδιών. Καλείται, λοιπόν, η συνάρτηση parent που δεν επιστρέφει κάποια τιμή (είναι τύπου void). Ο κώδικας της συνάρτησης parent υπάρχει στο αρχείο parent.c και η λειτουργία της εξηγείται παρακάτω.

Εφόσον έχουν ικανοποιηθεί όλα τα αιτήματα ( ο αριθμός των αιτημάτων και ο αριθμός των απαντήσεων είναι K\*N) τότε εκτελείται η συνάρτηση wait με όρισμα το status ώστε να ολοκληρωθούν όλα τα παιδιά.

Στο τέλος, καταστρέφονται όλοι οι σημαφόροι που χρειάστηκαν με τη συνάρτηση sem\_destroy, καθώς και τα 2 κομμάτια της διαμοιραζόμενης μνήμης, μέσω των συναρτήσεων shmdt και shmctl, οι οποίες καλούνται 2 φορές η κάθε μία.

### **child.c**

Στο αρχείο αυτό υπάρχει ο κώδικας της συνάρτησης child. Η συνάρτηση child δέχεται ως ορίσματα έναν δείκτη σε request, έναν δείκτη σε response ώστε να έχει πρόσβαση στο κομμάτι request και στο κομμάτι response της διαμοιραζόμενης μνήμης αντίστοιχα. Επίσης έναν ακέραιο (το όνομα της μεταβλητής λέγεται child) που δείχνει τον αριθμό του παιδιού, έναν ακόμη ακέραιο που γνωστοποιεί στη συνάρτηση τον αριθμό των γραμμών του αρχείου κειμένου και άλλες 2 μεταβλητές τύπου clock\_t που χρειάζονται στην εύρεση του χρόνου εκτέλεσης του παιδιού.

Αρχικά, εκτελείται η συνάρτηση clock που σηματοδοτεί την αρχή του αιτήματος. Ύστερα ο σημαφόρος my\_turn γίνεται down ώστε να μην ξεκινήσει άλλο αίτημα. Στη συνέχεια, γίνονται down οι σημαφόροι empty (πρόκειται να προστεθεί ένα ακόμα αίτημα στον buffer) και mutex (αμοιβαίος αποκλεισμός).

Εισάγονται στη διαμοιραζόμενη μνήμη τα στοιχεία του αιτήματος, στο κομμάτι του request, δηλαδή ποιο παιδί έχει κάνει αίτημα και ποια γραμμή ζητάει. Εκτυπώνεται ανάλογο μήνυμα έπειτα από αμοιβαίο αποκλεισμό με τη βοήθεια του σημαφόρου lock. Ο αριθμός των αιτημάτων αυξάνεται κατά 1 και οι σημαφόροι empty και mutex γίνονται up, καθώς έχει ολοκληρωθεί η διαδικασία του αιτήματος. Στο σημείο αυτό, το παιδί με το να κάνει down το σημαφόρο buff\_sem που αφορά στην απάντηση για το συγκεκριμένο παιδί (res[child].buff\_sem), περιμένει να λάβει απάντηση από το γονέα. Μόλις λάβει απάντηση εκτελεί την συνάρτηση clock για να βρει τον χρόνο που χρειάστηκε για να τελειώσει το αίτημα και ύστερα τυπώνει την απάντηση που έγραψε μέσα στην διαμοιραζόμενη μνήμη ο γονεάς (μέσα στον πίνακα line[101] που βρίσκεται στο κομμάτι response).

Τέλος, η συνάρτηση process επιστρέφει τον χρόνο που χρειάστηκε, τη μεταβλητή total\_time.

### **parent.c**

Η συνάρτηση parent δέχεται ορίσματα: έναν δείκτη σε request, έναν δείκτη σε response ώστε να έχει πρόσβαση στο κομμάτι request και στο κομμάτι response της διαμοιραζόμενης μνήμης αντίστοιχα. Επίσης, το αρχείο κειμένου. Αρχικά, κάνει down τους σημαφόρους full (αφαιρείται ένα αίτημα) και mutex. Εν συνεχεία ο αριθμός του παιδιού που έκανε το αίτημα γράφεται στην διαμοιραζόμενη μνήμη του κομματιού response, καθώς και η γραμμή που ζήτησε το παιδί ώστε ο γονεάς να αναζητήσει τη συγκεκριμένη γραμμή. Ο γονεάς αφού ξέρει ποια γραμμή έχει ζητήσει το παιδί αναζητά στο αρχείο κειμένου την συγκεκριμένη γραμμή και έπειτα κάνει up το σημαφόρο buff\_sem που αναφέρεται στο συγκεκριμένο παιδί (sem\_post(&res[res->child\_res].buff\_sem)), ώστε η child process να σταματήσει να περιμένει καθώς τώρα γνωρίζει ότι έχει ικανοποιηθεί το αίτημά της. Οι απαντήσεις που έχει εκπληρώσει ο γονεάς αυξάνονται κατά 1 και ο σημαφόρος my\_turn γίνεται up ώστε να ξεκινήσει κάποιο άλλο αίτημα να ζητηθεί από το παιδί. Τέλος, οι σημαφόροι mutex και empty γίνονται up, όπως στο πρότυπο producer-consumer για τη διαδικασία του consumer.

### **pink floyd lyrics.txt**

Το συγκεκριμένο αρχείο κειμένου είναι αυτό που δίνεται ως όρισμα στο τερματικό μέσα από το οποίο τα παιδιά αιτούνται τυχαία γραμμές και ο γονεάς τις επιστρέφει. Σε αυτό περιέχονται οι στίχοι από 3 κορυφαία τραγούδια, τα: High Hopes, Time και Sorrow των Pink Floyd, ενός από τα καλύτερα συγκροτήματα όλων των εποχών!!!

Τέλος, παρατίθεται ένα παράδειγμα από την εκτέλεση του κώδικα της εργασίας στα linux της σχολής. Το παράδειγμα εκτελείται για 3 παιδιά όπου το κάθε παιδί πραγματοποιεί 2 αιτήματα.

```
linux15:/home/users/sdi1800171/1115201800171_Thomais_Santa>make
cc -g -Wall -c -o child.o child.c
cc -g -Wall -c -o parent.o parent.c
cc -g -Wall -c -o main.o main.c
gcc child.o parent.o main.o -o project1 -lpthread
linux15:/home/users/sdi1800171/1115201800171_Thomais_Santa>./project1 pink_floyd_lyrics.txt 3 2
Child 1 to parent: 45
Parent to child 1 line 45: And you run, and you run to catch up with the sun but it's sinking__line45

Child 1 to parent: 57
Child 3 to parent: 9
Parent to child 1 line 57: Far away across the field__line57

Time to execute for child 1 : 0.000021
Child 2 to parent: 40
Parent to child 3 line 9: Running before times took our dreams away__line9

Parent to child 2 line 40: Waiting for someone or something to show you the way__line40

Child 3 to parent: 62
Parent to child 3 line 62: 3.PINK FLOYD - SORROW (LYRICS)__line62

Time to execute for child 3 : 0.000008
Child 2 to parent: 64
Parent to child 2 line 64: Plumes of smoke rise and merge into the leaden sky__line64

Time to execute for child 2 : 0.000059
linux15:/home/users/sdi1800171/1115201800171_Thomais_Santa>
```