



**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**Τίτλος Εργασίας**

1η Προγραμματιστική Εργασία

**Μάθημα**

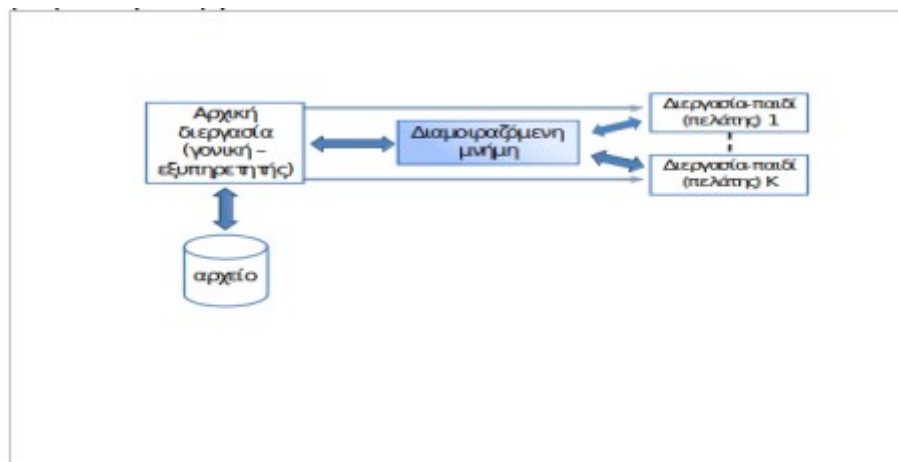
Λειτουργικά Συστήματα (Κ22)  
Χειμερινό εξάμηνο 2021-22

**Ονοματεπώνυμο φοιτητή:**

- Μαυραπίδης Νικόλαος (Α.Μ.: 11152017 00082)

**Αθήνα, 2021**

## ΣΥΝΤΟΜΗ ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ



Όπως βλέπουμε και στο διάγραμμα, τα εκτελέσιμα μας είναι 2 parent(Αρχική διεργασία) και child(Διεργασία παιδί). Μεταξύ τους δημιουργείται κομμάτι **ενδιάμεσης μνήμης για μεταφορά δεδομένων**. Ταυτόχρονα μεταξύ τους **δημιουργούνται 2 σημαφόροι**. Έτσι το πρόβλημα απλοποιείται πολύ περισσότερο και γίνεται πιο ευανάγνωστο. Καλό θα ήταν να επισημάνουμε ότι τα εκτελέσιμα που χρειάζονται να τρέξουν για το πρόγραμμα **είναι το ./parent**. Αυτό γιατί η **parent** δημιουργεί **K child-ren**, ενώ ταυτόχρονα στην διαδικασία αυτή δημιουργούνται και οι απαραίτητοι σημαφόροι και ενδιάμεση μνήμη.

Για την αναπαράσταση των σημαφόρων και της ενδιάμεσης μνήμης χρησιμοποιήθηκαν **POSIX semaphores** και **shared memory**. Το πρόγραμμα είναι υλοποιημένο σε γλώσσα **C/C++** και σε περιβάλλον **linux**.

## ΚΑΤΑΛΟΓΟΣ ΑΡΧΕΪΩΝ ΚΩΔΙΚΑ ΚΑΙ ΕΠΙΚΕΦΑΛΪΔΩΝ

- **parent.cpp**: Το αρχείο περιέχει την γονεϊκή **main()** συνάρτηση από όπου δημιουργείται η θυγατρική διαδικασία **./child K φορές**. Πιο συγκεκριμένα δημιουργούνται **2 σημαφόροι** στο επίπεδο μεταξύ **parent - child** καθώς και η ενδιάμεση τους **shared memory** για την μετάβαση μηνυμάτων από και προς τις δύο διεργασίες. Έπειτα με την βοήθεια της συνάρτησης **fork()** **δημιουργούνται K διεργασίες παιδιά children** όπου το καθένα έχει την ίδια λειτουργία που περιγράφεται στον πηγαίο κώδικα **./child**. **Αφότου** το παιδί γράφει στην ενδιάμεση μνήμη του γονέα τον αριθμό της γραμμής του αρχείου που θέλει, ο γονέας διαβάζει αυτόν τον αριθμό, **ψάχνει την γραμμή στο αρχείο με**

---

βάση τον αριθμό αυτόν και **τέλος γράφει πάλι στην ενδιάμεση μνήμη την γραμμή** για να διαβαστεί πίσω από το παιδί. Στο τέλος περιμένει να τελειώσουν όλες οι θυγατρικές διαδικασίες ενώ εκτυπώνει το **id** καθενός και το **status** με το οποίο ολοκληρώθηκαν.

- **child.cpp**: Αποτελεί την θυγατρική διεργασία που δημιουργείται μέσω **fork() - execvp()** από την **parent**. Η λειτουργία της είναι να **γράψει στην ενδιάμεση μνήμη τον αριθμό γραμμής** που ζητάει από τον πατέρα και τέλος αφότου ο πατέρας της δώσει την γραμμή **να την διαβάσει από την ενδιάμεση μνήμη και να την εκτυπώσει**. Σε αυτό το σημείο να τονίσω ότι σε περίπτωση που δημιουργηθούν πάνω από μια θυγατρικές διεργασίες, δημιουργήθηκε ο **σημαφόρος child\_using\_parent** ο οποίος εσωκλείνει **critical section** κώδικα, για να σηματοδοτηθεί το **availability** του πατέρα και να μπορέσουν έτσι τα παιδιά να γράφουν και να διαβάζουν από την ενδιάμεση μνήμη με ασφάλεια και εγκυρότητα, χωρίς να περιπλέκονται μεταξύ τους. Ταυτόχρονα γίνονται **μετρήσεις του χρόνου** σε **microseconds** και στο τέλος εκτυπώνεται ο μέσος χρόνος με βάση το **N** που δόθηκε αρχικά από τον χρήστη.
- **aux.cpp**: Το αρχείο αυτό δημιουργήθηκε καθαρά για βοηθητικούς λόγους και προσφέρει άνεση και ευαναγνωσιμότητα στο κυρίως πρόγραμμα.

## ΕΚΤΕΛΕΣΗ ΚΑΙ ΧΡΗΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Ο παρεχόμενος φάκελος συμπεριλαμβάνει ένα **makefile** για την μεταγλώττιση όλων των εκτελέσιμων. Αφότου μεταγλωττιστούν τα αρχεία, ο χρήστης οφείλει να τρέξει την εντολή:

**./parent <filename> <K> <N>**

Ξεκινάει λοιπόν η προσομοίωση αφότου ο χρήστης δώσει το όνομα του αρχείου (**filename**), τον αριθμό των παιδιών (**K**) και τον αριθμό των δοσοληψιών (**N**).