

# ΕΡΓΑΣΙΑ 1 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

## ΓΕΩΡΓΙΟΣ ΚΑΡΤΕΡΗΣ

1115202100258

### Εισαγωγή

Το πρόγραμμα δημιουργεί και διαχειρίζεται διεργασίες σε σύστημα UNIX ή Linux με χρήση τεχνικών διαμοιραζόμενης μνήμης (shared memory) και σημαφόρων (semaphores). Ο κύριος στόχος του είναι η διαχείριση θυγατρικών διεργασιών (παιδιών) που επικοινωνούν με τη διεργασία γονέα (η πιο απλά τον γονέα) και εκτελούν εντολές (που αποστέλλονται μέσω κοινής μνήμης) και με χρήση σημαφόρων (που διασφαλίζουν τη σειρά των ενεργειών). Ο γονέας εκκινεί ή τερματίζει τα παιδιά ανάλογα με τις εντολές που διαβάζει από ένα αρχείο εισόδου. Κάθε παιδί λαμβάνει μηνύματα από τον γονέα για να εκτελεί εργασίες και μετρά πόσα μηνύματα έχει λάβει καθώς και τη διάρκεια της δραστηριότητάς του. Το πρόγραμμα παρέχει έναν αποτελεσματικό τρόπο συγχρονισμού διαδικασιών και ανταλλαγής δεδομένων σε περιβάλλον πολλαπλών διεργασιών.

### Απαιτήσεις και Διαμόρφωση Αρχείων

#### Αρχείο Εντολών

Το πρόγραμμα απαιτεί ένα αρχείο εντολών που καθορίζει τη χρονική στιγμή και την ενέργεια που πρέπει να εκτελέσει ο γονέας. Η γραμμογράφηση του αρχείου είναι:

```
timestamp process command:
```

Όπου:

- `timestamp`: Η χρονοσφραγίδα για την εκτέλεση της εντολής.
- `process`: Η αναγνώριση του παιδιού (π.χ., C1 για το παιδί 1).
- `command`: Η εντολή που εκτελείται, είτε `s` για δημιουργία του παιδιού, είτε `T` για τερματισμό του παιδιού.

Κάθε αρχείο εντολών τελειώνει με μια εντολή της μορφής:

```
timestamp EXIT
```

Όπου:

- `timestamp`: Η χρονοσφραγίδα για την εκτέλεση της εντολής
- η λέξη `EXIT` σηματοδοτεί την λήξη εκτέλεσης του προγράμματος.

Υπάρχουν διάφορα αρχεία εντολών με διαφορετικό αριθμό εντολών:

- `config_3_100.txt`: Περιέχει 100 εντολές.
- `config_3_1000.txt`: Περιέχει 1000 εντολές.
- `config_3_10000.txt`: Περιέχει 10000 εντολές.

## Αρχείο Κειμένου (`mobydick.txt`)

Το πρόγραμμα απαιτεί ένα αρχείο κειμένου (π.χ., το βιβλίο "Moby Dick") για την αποστολή μηνυμάτων από τον γονέα στα παιδιά. Το αρχείο αυτό διαβάζεται γραμμή προς γραμμή, και οι γραμμές αποστέλλονται στα παιδιά ως μηνύματα.

## Περιγραφή του Κώδικα

### Δομή Δεδομένων: `ChildProcess`

Η δομή `ChildProcess` αποθηκεύει πληροφορίες για κάθε παιδί. Περιλαμβάνει:

- `pid`: Το PID της διαδικασίας του παιδιού.
- `active`: Ένα flag που δείχνει αν το παιδί είναι ενεργό.
- `messages_received`: Ο αριθμός των μηνυμάτων που έχει λάβει το παιδί.
- `start_time`: Η χρονοσφραγίδα εκκίνησης του παιδιού.
- `end_time`: Η χρονοσφραγίδα τερματισμού του παιδιού.

```
typedef struct {
    pid_t pid;
    int active;
    int messages_received;
    int start_time;
    int end_time;
} ChildProcess;
```

### Διαχείριση Κοινής Μνήμης και Σημάτων

Το πρόγραμμα χρησιμοποιεί κοινή μνήμη για την αποθήκευση των μηνυμάτων που στέλνονται από τον γονέα στα παιδιά. Για να συγχρονιστούν οι διαδικασίες, χρησιμοποιούνται σημαφόροι. Η συνάρτηση `sem_op()` εκτελεί την κατάλληλη λειτουργία στους σημαφόρους, επιτρέποντας στον γονέα να συγχρονίζει τα παιδιά.

### Δημιουργία και Τερματισμός Παιδιών

- **Δημιουργία Παιδιών:** Η συνάρτηση `spawn_child()` δημιουργεί μια νέα θυγατρική διεργασία με τη συνάρτηση `fork()`. Το παιδί αναμένει μηνύματα και εκτελεί ενέργειες με βάση τα μηνύματα που λαμβάνει από την κοινή μνήμη.
- **Τερματισμός Παιδιών:** Η συνάρτηση `terminate_child()` στέλνει το μήνυμα "TERMINATE" στην κοινή μνήμη και χρησιμοποιεί τον σημαφόρο για να τερματίσει το παιδί. Τέλος, καταγράφει τις στατιστικές δραστηριότητας του παιδιού.

### Εκτέλεση Εντολών

Μετά την εκκίνηση διεργασιών, σε κάθε κύκλο λειτουργίας της, ο γονέας επιλέγει τυχαία μία από τις ενεργές διεργασίες-παιδιά και της προωθεί μία (τυχαία επιλεγμένη) γραμμή κειμένου από το αρχείο. Η διεργασία υποδοχής εκτυπώνει την παραληφθείσα γραμμή κειμένου στην τυπική έξοδο. Ο γονέας για κάθε εντολή που διαβάζει από το αρχείο εντολών δημιουργεί ή τερματίζει τα παιδιά και στέλνει μηνύματα για να εκτελούν εργασίες. Αν η εντολή αφορά την εκκίνηση ενός παιδιού, καλείται η συνάρτηση `spawn_child()`, ενώ αν αφορά τον τερματισμό, καλείται η συνάρτηση `terminate_child()`, ενώ αν αφορά την εντολή `EXIT`, τότε τερματίζει όλες τις ενεργές διεργασίες καθώς και το πρόγραμμα.

## Εκτέλεση του Προγράμματος

Για να εκτελέσετε το πρόγραμμα, ακολουθήστε τα παρακάτω βήματα:

### Μεταγλώττιση του Προγράμματος

Χρησιμοποιήστε την εντολή: `gcc -o pr my_OS.c -lrt`

### Εκτέλεση του Προγράμματος

Για να εκτελέσετε το πρόγραμμα, παρέχετε τα αρχεία εντολών και το αρχείο κειμένου ως ορίσματα. Παράδειγμα:

```
./pr config_3_100.txt moby dick.txt 3
```

Αυτό το παράδειγμα θα δημιουργήσει 3 παιδιά και θα εκτελέσει τις εντολές από το αρχείο `config_3_100.txt`, χρησιμοποιώντας το αρχείο `moby dick.txt` για την αποστολή μηνυμάτων.

### Αναμονή και Εκτέλεση

Το πρόγραμμα εκτελεί τις εντολές και παρακολουθεί τη δραστηριότητα των παιδιών. Όταν όλα τα παιδιά ολοκληρώσουν τις εργασίες τους ή όταν συναντηθεί η εντολή `EXIT`, το πρόγραμμα θα τερματίσει.