

ΕΡΓΑΣΙΑ 1



ΟΜΑΔΑ ΣΥΝΤΑΞΗΣ

Όνοματεπώνυμο	A.M.	EMAIL
ΧΡΗΣΤΟΣ ΚΡΑΤΗΜΕΝΟΣ	1115202000234	sdi2000234@di.uoa.gr

Η εργασία υλοποιεί ένα σύστημα επικοινωνίας μεταξύ διεργασιών (IPC) χρησιμοποιώντας κοινόχρηστη μνήμη και σηματοφόρους για τον συντονισμό πολλαπλών διεργασιών.

Μέσα στο τμήμα κοινόχρηστης μνήμης, διατηρούνται μια προσωρινή μνήμη μηνυμάτων και πίνακες που παρακολουθούν τις πληροφορίες του child process, συμπεριλαμβανομένων των PID και των ωρών έναρξης. Το Parent process συντονίζει τη δημιουργία νέων child process με το μήνυμα "SPAWN" και τους δίνει εντολή να τερματίσουν με το μήνυμα "TERMINATE". Επιπλέον, όταν δίνονται οδηγίες και ενώ τα παιδιά είναι ενεργά, ο γονέας επιλέγει τυχαία ένα child process και του στέλνει μια γραμμή κειμένου από το αρχείο mobydick.txt. Κάθε child process όταν λαμβάνει μηνύματα κειμένου, αυξάνει τον αριθμό των επεξεργασμένων μηνυμάτων τους και παραμένει ενεργο μέχρι να λάβουν ένα μήνυμα τερματισμού. Όταν τερματίζει, εκτυπώνει το πλήθος των μηνυμάτων που παρέλαβε καθώς και το συνολικό πλήθος βημάτων (χρονοσήμανση ολοκλήρωσης – χρονοσήμανση εκκίνησης) για τα οποία έμεινε ενεργή, ακριβώς όπως λει και η εκφώνηση

Από σχεδιαστική άποψη, δεν χρησιμοποιούνται κλήσεις συστήματος sleep() ή kill(). Η απουσία αυτών των κλήσεων συστήματος διασφαλίζει ότι ο συγχρονισμός επιτυγχάνεται αποκλειστικά μέσω χρήσης σηματοφόρου, μετάδοσης μηνυμάτων και ελεγχόμενης ανάγνωσης και εγγραφής στην κοινόχρηστη μνήμη. Για τον τερματισμό ενός child process, το parent process δεν στέλνει σήμα, αλλά τοποθετεί ένα ειδικά διαμορφωμένο μήνυμα "TERMINATE" στην κοινόχρηστη μνήμη και δημοσιεύει το σηματοφόρο του child. Με την ανάγνωση αυτού του μηνύματος, το child process ολοκληρώνει τη δραστηριότητά του και τερματίζει κανονικά, αναφέροντας πόσα μηνύματα επεξεργάστηκε και πόσα βήματα ήταν ενεργό.

Ο κώδικας χωρίζεται σε τρία κύρια αρχεία: sharedmem.c, child.c και parent.c. Το πρόγραμμα sharedmem.c προετοιμάζει το τμήμα κοινόχρηστης μνήμης και τον Parent σηματοφόρο. Ρυθμίζει την κοινόχρηστη μνήμη με το όνομα /shared_memory και διασφαλίζει ότι είναι έτοιμη να αντιστοιχιστεί από τις άλλες διεργασίες. Δημιουργεί επίσης τον γονικό σηματοφόρο με το όνομα /sem_parent και εκτυπώνει διαγνωστικές πληροφορίες κατά την επιτυχή

προετοιμασία. Μόλις ολοκληρωθεί αυτό το βήμα, το περιβάλλον είναι έτοιμο για εκτέλεση της γονικής διαδικασίας. Επιπλέον, εκτυπώνονται μηνύματα στο terminal για να βεβαιωθούμε ότι δεν υπάρχει πρόβλημα κατά το allocation.

Το πρόγραμμα `parent.c` ανοίγει την κοινόχρηστη μνήμη και το `parent process` που δημιουργήθηκε προηγουμένως. Στη συνέχεια διαβάζει ένα αρχείο εντολών γραμμή προς γραμμή, προσδιορίζοντας μια χρονική σήμανση και μια ενέργεια. Οι ενέργειες μπορεί να περιλαμβάνουν τη δημιουργία θυγατρικής διεργασίας (σημειώνεται ως "S", τον τερματισμό μιας θυγατρικής διεργασίας (σημειώνεται ως "T") ή την έκδοση εντολής "EXIT" για τον τερματισμό όλων των ενεργών παιδιών και τον τερματισμό του προγράμματος. Κάθε φορά που κάνει `spawn` ένα παιδί, ο `parent` δημιουργεί έναν μοναδικό σηματοφόρο για αυτό το παιδί και εκτελεί το πρόγραμμα `child.c`. Το βήμα ώρας έναρξης κάθε παιδιού καταγράφεται στην κοινόχρηστη μνήμη για το τέλος. Για τον τερματισμό, ο γονέας τοποθετεί ένα μήνυμα "TERMINATE: "χρονος που εγινε το terminate"" στην κοινόχρηστη μνήμη και σηματοδοτεί τον σηματοφόρο του αντίστοιχου παιδιού. Το `child process`, διαβάζοντας αυτό το μήνυμα, βγαίνει και εκτυπώνει τα στατιστικά του.

Μετά την επεξεργασία μιας εντολής, εάν το σύστημα παραμένει ενεργό και υπάρχουν ακόμη παιδιά, ο γονέας διαβάζει μια γραμμή κειμένου από ένα ξεχωριστό αρχείο (`mobydick.txt`), επιλέγει ένα τυχαίο ενεργό παιδί και στέλνει τη γραμμή μηνύματος στο εκείνο το παιδί. Εάν το αρχείο κειμένου τελειώσει, ο γονέας ξεκινάει από την αρχή. Χρησιμοποιούνται σηματοφόροι για να εγγυηθούμε ότι μόνο μία οντότητα γράφει στην κοινόχρηστη μνήμη κάθε φορά και ότι το `child process` αναγνωρίζει σωστά τη λήψη του μηνύματος πριν ξεκινήσει η επόμενη λειτουργία.

Το πρόγραμμα `child.c` δείχνει την διαδικασία που εκτελείται από κάθε `child process`. Κατά την εκκίνηση, το παιδί αποθηκεύει την κοινόχρηστη μνήμη και ανοίγει τους σχετικούς σηματοφόρους: τον `parent` σηματοφόρο και τον δικό του σηματοφόρο. Στη συνέχεια, το παιδί περιμένει στον σηματοφόρο του μέχρι να κάνει `post` ο γονέας, για να πει ότι ένα νέο μήνυμα είναι διαθέσιμο στην κοινόχρηστη μνήμη. Το παιδί διαβάζει το μήνυμα. Εάν το μήνυμα είναι μια κανονική γραμμή κειμένου, το παιδί αυξάνει τον εσωτερικό μετρητή των επεξεργασμένων μηνυμάτων και επιβεβαιώνει τη λήψη στον γονέα. Εάν το μήνυμα είναι μήνυμα τερματισμού, το παιδί υπολογίζει πόσο καιρό ήταν ενεργό χρησιμοποιώντας την ώρα έναρξης που είναι αποθηκευμένη στην κοινόχρηστη μνήμη. Εκτυπώνει πόσα μηνύματα επεξεργάστηκε και τη διάρκεια της δραστηριότητάς του και μετά τερματίζει.

Η δομή `SharedMemory` περιέχει μια προσωρινή μνήμη μηνυμάτων, πίνακες για `child PID`, μια μέτρηση πόσων παιδιών έχουν ξεκινήσει και πίνακες για την αποθήκευση των βημάτων έναρξης για κάθε `child`. Το όνομα σηματοφόρου κάθε παιδιού κατασκευάζεται δυναμικά χρησιμοποιώντας το `index` του, διασφαλίζοντας ότι κάθε παιδί έχει έναν μοναδικό σηματοφόρο. Ο `parent` σηματοφόρος με το όνομα `/sem_parent` συντονίζει τη διέλευση μηνυμάτων και διασφαλίζει ότι δεν έχουμε collisions. Χρησιμοποιώ POSIX συναρτήσεις, όπως `shm_open`, `sem_open`, `mmap`, `sem_wait` και `sem_post`. Η κοινόχρηστη μνήμη και οι σηματοφόροι αποσυνδέονται και κλείνουν στο τέλος της εκτέλεσης.

Όλες οι διεργασίες βασίζονται σε `-lrt` και `-lpthread` για τη σωστή σύνδεση των απαιτούμενων βιβλιοθηκών για την κοινή μνήμη και τις λειτουργίες σηματοφόρου. Πρωτα εκτελούμε το

./sharedmem για την προετοιμασία της κοινόχρηστης μνήμης και του parent σηματοφόρου. Μετά από αυτό, καλείται το εκτελεστικό ./parent M K config_file.txt για να εκτελέσει την κύρια λογική με τις παραμέτρους M και K (πάντα πρέπει να ισχύει ότι $K < \text{max_children}$) και ένα αρχείο εντολών που καθοδηγεί τον parent σχετικά με το πότε να κάνει spawn ή να τερματίσει τα παιδιά. Ο γονέας διαβάζει από το mobydick.txt για να λάβει μηνύματα για αποστολή. Όταν ολοκληρωθεί, τα παιδιά εκτυπώνουν τα στατιστικά τους στοιχεία και τερματίζουν. Ο parent καθαρίζει τους πόρους, έτσι ώστε να μην ξεμεινουν τμήματα μνήμης ή σηματοφόροι.

Για το compile όλων των αρχείων, εφόσον είμαστε στο dir του Makefile, μπορούμε να τρέξουμε make.

Ακολουθούν ενδεικτικές εκτελέσεις:

```
linux02:/home/users/sdi2000234/projects/OS_1/OSExercise>./sharedmem
Semaphore /sem_parent created successfully.
Shared memory created at: 0x7f26b9a61000
Shared memory and semaphores successfully initialized.
linux02:/home/users/sdi2000234/projects/OS_1/OSExercise>./parent 20 10 config_3_
100.txt
Child PID 391188 processed 2 messages and was active for 33 steps before termina
tion.
Child PID 391189 processed 0 messages and was active for 8 steps before termina
tion.
Child PID 391191 processed 1 messages and was active for 33 steps before termina
tion.
Child PID 391190 processed 3 messages and was active for 50 steps before termina
tion.
linux02:/home/users/sdi2000234/projects/OS_1/OSExercise>./sharedmem
Semaphore /sem_parent created successfully.
Shared memory created at: 0x7fd368048000
Shared memory and semaphores successfully initialized.
linux02:/home/users/sdi2000234/projects/OS_1/OSExercise>./parent 10 10 config_10_10000.txt
M must be at least K+1 (M=10, K=10)
linux02:/home/users/sdi2000234/projects/OS_1/OSExercise>
```