Λειτουργικά Συστήματα
Εργασία 1

Αθανασία Τουρνάκη
ΑΜ: 1115201600172

Σε αυτήν την εργασία υλοποιήθηκε μια προσομοίωση γραμμής παραγωγής και απαντήθηκαν όλα τα σχετικά ερωτήματα. Ο χρήστης καλείται όταν τρέξει το πρόγραμμα να δώσει σαν όρισμα στη γραμμή εντολών τον αριθμό των εξαρτημάτων που θα ήθελε να δημιουργούνται σε κάθε στάδιο κατασκευής(ίδιος αριθμός σε όλα τα στάδια).

Κάθε στάδιο της παραγωγής, όπως αυτά περιγράφονται στην εκφώνηση της εργασίας, αντιμετωπίστηκε σαν μία διεργασία και πραγματοποιήθηκε ο απαραίτητος συγχρονισμός με τη χρήση σημαφόρων για τη σωστή λειτουργία της προσομοίωσης.

Συνολικά, οι **διεργασίες** που δημιουργήθηκαν σε αυτό το πρόγραμμα είναι **8**(μη συμπεριλαμβανομένης της διεργασίας-πατέρα main), τρεις για την κατασκευή των εξαρτημάτων(μία για κάθε τύπο εξαρτήματος), μία για το βαφείο, τρεις για τον έλεγχο των εξαρτημάτων(όπως η κατασκευή) και άλλη μία για την συναρμολόγηση του τελικού προϊόντος.

Επίσης, χρησιμοποιήθηκαν **5 τμήματα διαμοιραζόμενης μνήμης** για την επικοινωνία μεταξύ των διεργασιών, ένα ανάμεσα στην κατασκευή και το βαφείο, τρία ανάμεσα στο βαφείο και τον έλεγχο(ένα για κάθε τύπο εξαρτήματος) και ένα ανάμεσα στον έλεγχο και την συναρμολόγηση. Κάθε τμήμα δεσμεύει χώρο για ακριβώς ένα εξάρτημα.

Τέλος χρησιμοποιήθηκαν 5 σετ σημαφόρων μεγέθους 3, άρα συνολικά **15 σημαφόροι**.

Το πρόγραμμα έχει διασπαστεί σε πολλά διαφορετικά αρχεία, το καθένα με διαφορετική λειτουργία.

Πιο αναλυτικά:

- **Sem_shm_fun.c:** Σε αυτό το αρχείο περιέχονται συναρτήσεις για την εύκολη δημιουργία, ανάκτηση και διαγραφή σετ σημαφόρων και τμημάτων διαμοιραζόμενης μνήμης με τη χρήση αρχείων για την αποθήκευση των αντίστοιχων κλειδιών. Ένα “λεξικό” με τα ονόματα των αρχείων που χρησιμοποιούνται για την αποθήκευση των κλειδιών περιεχεται στο αρχείο keys.h. Για τη δημιουργία των συναρτήσεων αυτών, έχουν χρησιμοποιηθεί οι σημειώσεις του εργαστηρίου 2 και συγκεκριμένα ο κώδικας 4 για Server και Client.

- **Semaphores.c:** Σε αυτό το αρχείο περιέχονται συναρτήσεις για την αρχικοποίηση και την τροποποίηση σημαφόρων, οι οποίες έχουν παρθεί από τον κώδικα 3 του εργαστηρίου 2 αυτούσιες. Στο αντίστοιχο αρχείο Semaphores.h έχει οριστεί το union semun που είναι απαραίτητο για την λειτουργία των παραπάνω συναρτήσεων.

- **Component.c:** Σε αυτό το αρχείο περιέχονται συναρτήσεις για την δημιουργία ενός εξαρτήματος (που περιλαμβάνει την χρονοσήμανσή του και την απόδοση ενός μοναδικού τετραψήφιου αναγνωριστικού) και τη δημιουργία ενός τελικού προϊόντος(που περιλαμβάνει το συνδυασμό των αναγνωριστικών των εξαρτημάτων που χρησιμοποιήθηκαν για την παραγωγή του για την δημιουργία ενός αναγνωριστικού για το τελικό προϊον και επίσης τον υπολογισμό του χρόνου παραγωγής από τη στιγμή που δημιουργήθηκε το παλαιότερο από τα εξαρτήματά του μέχρι την συναρμολόγησή του). Για την χρονοσήμανση του εξαρτήματος(και σε άλλα σημεία του προγράμματος) χρησιμοποιήθηκε η δομή timespec και η συνάρτηση gettime με πρώτο όρισμα CLOCK_MONOTONIC(που μετρά σταθερά τον χρόνο από ένα συγκεκριμένο σημείο και δεν επηρεάζεται από το εσωτερικό ρολόι του συστήματος) και τελέστηκαν οι απαραίτητες ενέργειες για τις μετατροπές σε δευτερόλεπτα.

- **DynamicQueue.c:** Περιέχει συναρτήσεις δημιουργίας, εκτύπωσης, ένταξης, αφαίρεσης και καταστροφής για μία δυναμική ουρά(η οποία έχει οριστεί στο DynamicQueue.h). Τρεις τέτοιες ουρές θα χρησιμοποιηθούν στην συνάρτηση συναρμολόγησης για να κρατάνε τα εξαρτήματα συγκεκριμένου τύπου πριν αυτά συνδυαστούν.

- **Manufacturing.c:** Αυτό το αρχείο περιέχει τον κώδικα για τις συναρτήσεις-τμήματα παραγωγής. Στην αρχή του αρχείου, ορίζονται οι χρόνοι καθυστέρησης που αντιστοιχούν στο ομώνυμο τμήμα κατασκευής. Συνολικά, υπάρχουν 4 συναρτήσεις. Αναλυτικά για την καθεμία:

  1) void Construction(int number, int type): Πρόκειται για την συνάρτηση κατασκεύης. Γίνεται κλήση της συνολικά τρεις φορές, μία για κάθε τύπο εξαρτήματος. Καλείται με ορίσματα τον αριθμό των εξαρτημάτων που πρέπει να παραχθούν και τον τύπο εξαρτήματος που πρέπει να δημιουργηθεί. Αρχικά, ανακτώνται(με τις συναρτήσεις του Sem_shm_fun.c) το τμήμα μνήμης και το σετ 3 σημαφόρων που θα χρησιμοποιηθούν για την επικοινωνία της με το Βαφείο(ένας mutex, ένας empty και ένας full, όπως αρχικοποιήθηκαν στην Main. Τα παρακάτω εκτελούνται επαναληπτικά τόσες φορές όσος είναι ο αριθμός των εξαρτημάτων που κατασκευάζονται σε κάθε στάδιο:

  α) Δημιουργείται ένα εξάρτημα(ουσιαστικά, “προκύπτει”, χωρίς καθυστέρηση).

  β)Για την διαδικασία μεταφοράς στην διαμοιραζόμενη μνήμη, ακολουθείται η διαδικασία που έχουμε διδακτεί για την επικοινωνία δύο τμημάτων Producer-Consumer, συγκεκριμένα το τμήμα του Producer σε αυτήν την περίπτωση. Η διαδικασία είναι η εξής: i) Ο σημαφόρος empty κατεβαίνει, ii)Ο σημαφόρος mutex κατεβαίνει, iii) Εκτελείται η κρίσιμη περιοχή, που είναι η εγγραφή του εξαρτήματος στην διαμοιραζόμενη μνήμη, iv) Ο σημαφόρος mutex ανεβαίνει, v) Ο σημαφόρος full ανεβαίνει.

  Η επανάληψη τερματίζει. Γίνεται αποδέσμευση της διαμοιραζόμενης μνήμης.

  2) void PaintShop(int number): Πρόκειται για την συνάρτηση Βαφείου. Καλείται με όρισμα τον αριθμό των εξαρτημάτων που θα περάσουν μέσα από αυτό(δηλαδή, 3 * τον αριθμό τον εξαρτημάτων που κατασκευάζονται σε ένα τμήμα κατασκευής). Αρχικά, ανακτώνται τα τμήματα διαμοιραζόμενης μνήμης ανάμεσα στην Κατασκευή και το Βαφείο και ανάμεσα στο Βαφείο και τον Έλεγχο για κάθε τύπο εξαρτήματος και δημιουργούνται δείκτες προς αυτά. Επίσης ανακτώνται τα σετ σημαφόρων που ελέγχουν τον συγχρονισμό στα παραπάνω σημεία. Τα παρακάτω εκτελούνται επαναληπτικά τόσες φορές όσος είναι ο αριθμός που δίνεται σαν όρισμα:

  α) Για να ανακτηθεί το εξάρτημα από τη διαμοιραζόμενη μνήμη, ακολουθείται και πάλι η διαδικάσια επικοινωνίας Producer-Consumer, συγκεκριμένα το τμήμα του Consumer. Η διαδικασία είναι η εξής: i) Ο σημαφόρος full κατεβαίνει, ii) Ο σημαφόρος mutex κατεβαίνει, iii) Εκτελείται η κρίσιμη περιοχή, που είναι η εγγραφή του ανάκτηση του εξαρτήματος από τη διαμοιραζόμενη μνήμη. iv) Ανεβαίνει ο σημαφόρος mutex, v) Ο σημαφόρος empty ανεβαίνει.

  β) Υπολογίζεται για το εξάρτημα που ανακτήθηκε ο χρόνος αναμονής του μέχρι να εξυπηρετηθεί από το βαφείο και αποθηκεύεται σε μία μεταβλητή εσωτερική της δομής Εξάρτημα.

  γ) Γίνεται η απαραίτητη καθυστέρηση και “βάφεται” το εξάρτημα.

  δ) Ανάλογα με τον τύπο του εξαρτήματος γίνεται αντιστοίχιση στον κατάλληλο δείκτη στο τμήμα μνήμης που αντιστοιχεί σε αυτόν τον τύπο και στον κατάλληλο σημαφόρο.

  ε) Εκτελείται η διαδικάσια εγγραφής στη διαμοιραζόμενη μνήμη της μορφής Producer-Consumer από τη μεριά του Producer, όπως περιγράφηκε παραπανω.

  Η επανάληψη τελειώνει. Τέλος, αποδεσμεύονται τα τμήματα μνήμης που χρησιμοποιήθηκαν.

  3) void Inspection(int number, int type) : Πρόκειται για την συνάρτηση Ελέγχου. Γίνεται κλήση της συνολικά τρεις φορές, μία για κάθε τύπο εξαρτήματος. Καλείται με ορίσματα τον αριθμό των εξαρτημάτων που θα ελέγξει(ο αριθμός των εξτημάτων κάθε τύπου) και τον τύπο των εξαρτημάτων που θα εξετάσει. Αρχικά, ανάλογα με τον τύπο που έχει δοθεί σαν όρισμα, ανακτά το κατάλληλο τμήμα μνήμης και το κατάλληλο σετ σημαφόρων και αρχικοποιεί τον δείκτη προς τη μνήμη. Επίσης, ανακτά το τμήμα μνήμης και το σετ σημαφόρων που θα χρησιμοποιηθούν για την επικοινωνία του τμήματος Ελέγχου με το τμήμα Συναρμολόγησης. Τα παρακάτω εκτελούνται επαναληπτικά τόσες φορές όσος είναι ο αριθμός που δόθηκε σαν όρισμα:

α) Γίνεται ανάκτηση του εξαρτήματος από την διαμοιραζόμενη μνήμη μεταξύ Βαφείου και Ελέγχου για τον συγκεκριμένο τύπο εξαρτήματος με την διαδικασία που περιγράφηκε παραπάνω για τον Consumer.

β) Γίνεται η καθυστέρηση που ζητείται και το εξάρτημα “εξετάζεται”.

γ) Γίνεται η εγγραφή του εξαρτήματος στο τμήμα διαμοιραζόμενης μνήμης μεταξύ Ελέγχου και Συναρμολόγησης, με την διαδικασία που περιγράφηκε για τον Producer.

Η επανάληψη τελειώνει. Τέλος, αποδεσμεύονται τα τμήματα μνήμης που χρησιμοποιήθηκαν.

4)void Assembly(int number): Πρόκειται γαι την συνάρτηση Συναρμολόγησης. Καλείται με όρισμα τον αριθμό των εξαρτημάτων που θα παραλάβει(3 * τον αριθμό που δημιουργεί ένα τμήμα κατασκευής). Αρχικα, ανακτώνται το τμήμα διαμοιραζόμενης μνήμης και το σετ σημαφόρων για την επικοινωνία Ελέγχου-Συναρμολόγησης. Επίσης, αρχικοποιούνται 3 δυναμικές ουρές, μία για κάθε τύπο εξαρτήματος. Επιπλέον, αρχικοποιούνται σε 0 οι δύο μεταβλητές που κρατούν τους μέσους χρόνους αναμονής για το Βαφείο και παραγωγής. Τα παρακάτω εκτελούνται επαναληπτικά τόσες φορές όσος είναι ο αριθμός που δίνεται σαν όρισμα:

α) Γίνεται ανάκτηση του εξαρτήματος από την διαμοιραζόμενη μνήμη μεταξύ Ελέγχου και Συναρμολόγησης, με τη διαδικασία που περιγράφηκε για τον Consumer.

β) Προστίθεται ο χρόνος αναμονής του εξαρτήματος που ανακτήθηκε για την κατάληψη του Βαφείου στην αντίστοιχη μεταβλητή που προαναφέρθηκε.

γ) Ανάλογα με τον τύπο του εξαρτήματος, αυτό εντάσσεται στην κατάλληλη ουρά.

δ)Αν όλες οι ουρές περιέχουν έστω και ένα εξάρτημα, αφαιρείται από κάθε ουρά το πρώτο στοιχείο, εκτελείται η ζητούμενη καθυστέρηση και με τον συνδυασμό των τριών εξαρτημάτων δημιουργείται το τελικό προϊον, ενώ ο χρόνος παραγωγής του προστίθεται στην αντίστοιχη μεταβλητή.

Η επανάληψη τελειώνει. Αποδεσμεύεται το τμήμα μνήμης που χρησιμοποιήθηκε. Για να υπολογιστεί ο μέσος όρος των χρόνων, η μεταβλητή αναμονής για το Βαφείο διαιρείται με τον συνολικό αριθμό των εξαρτημάτων, ενώ η μεταβλητή χρόνου παραγωγής διαιρείται με τον συνολικό αριθμό τελικών προϊόντων. Τέλος, εκτυπώνονται τα αποτελέσματα.

- **Main.c:** Είναι το βασικό μέρος του προγράμματος. Αρχικά, ελέγχεται αν ο χρήστης έχει καλέσει σωστά το πρόγραμμα. Σε περίπτωση που ο χρήστης δεν έχει δώσει όρισμα για τον αριθμό των εξαρτημάτων ή έχει δώσει παραπάνω ορίσματα, η λειτουργία διακόπτεται. Δεδομένου ότι ο χρήστης κάλεσε σωστά το πρόγραμμα, το επόμενο βήμα είναι η αρχικοποίηση των τμημάτων διαμοιραζόμενης μνήμης και των σημαφόρων που θα χρησιμοποιηθούν. Για τις αρχικοποιήσεις αυτές χρησιμοποιήθηκαν οι συναρτήσεις του Sem_shm_fun.c και του Semaphores.c. Έπειτα, ξεκινά η διαίρεση του προγράμματος σε διεργασίες. Για κάθε διεργασία πραγματοποιήθηκε μία κλήση της συνάρτησης fork() με διαφορετικό περιεχόμενο κάθε φορά ανάλογα με τον ρόλο της διεργασίας. Στο περιεχόμενο κάθε διεργασία-παιδιού περιέχεται και μια κλήση στη συνάρτηση που αντιστοιχεί στο κομμάτι της παραγωγής με το οποίο σχετίζεται η διεργασία. Η διεργασία- πατέρας περιμένει για την ολοκλήρωση όλων των διεργασιών-παιδιών του. Τέλος, διαγράφει σημαφόρους και τμήματα μνήμης(με τις συναρτήσεις του Sem_shm_fun.c) και τερματίζει.

<u>Οδηγίες για την εκτέλεση:</u>

Μαζί με τα αρχεία του προγράμματος συμπεριλαμβάνεται και ένα αρχείο **makefile.** Για να γίνει compile αρκεί ο χρήστης, ενώ είναι στον φάκελο με τα αρχεία να πληκτρολογήσει στο Terminal την εντολη **make**. Αυτό θα οδηγήσει στη δημιουργία του εκτελέσιμου Fabrication. Για να εκτελεστεί το πρόγραμμα, ο χρήστης πρέπει να πληκτρολογήσει:
./Fabrication <αριθμός εξαρτημάτων σε κάθε στάδιο κατασκευής>

Για την διαγραφή των δημιουργημένων αρχείων, αρκεί να πληκτρολογηθεί **make clean.**

Επιπλεον Πληροφορίες:

Οι καθυστερήσεις που έχουν χρησιμοποιηθεί στο Βαφείο, τον Έλεγχο και την Συναρμολόγηση είναι (τύπος εξαρτηματος * 1000), (τύπος εξαρτήματος * 1000) και (1000) αντίστοιχα. Οι συντελεστές είναι εύκολο να μεταβληθούν, αφού έχουν οριστεί με #define στο αρχείο Manufacturing.c, σε περίπτωση που ο χρήστης το θελήσει.

Τα αποτελέσματα που επιστρέφονται κατά την εκτέλεση του προγράμματος στους υπολογιστές Linux της σχολής είναι:

Για 10 εξαρτήματα:

```
The average waiting time for the PaintShop is 0.008796 seconds.
The average time for the fabrication of a Final Product is 0.024711
seconds.
```

Για 100 εξαρτήματα:

```
The average waiting time for the PaintShop is 0.008691 seconds.
The average time for the fabrication of a Final Product is 0.023909
seconds.
```

Για 1000 εξαρτήματα:

```
The average waiting time for the PaintShop is 0.008996 seconds.
The average time for the fabrication of a Final Product is 0.023626
seconds.
```

Για 2000 εξαρτήματα:

```
The average waiting time for the PaintShop is 0.008612 seconds.
The average time for the fabrication of a Final Product is 0.025075
seconds.
```

Παρατηρώ ότι ακόμα και με αλλαγές στις καθυστερήσεις και μετά από αρκετά πειράματα στον προσωπικό μου υπολογιστή, ο λόγος των δύο χρόνων που επιστρέφονται κυμαίνεται στο 1/3.