Λειτουργικά Συστήματα (Κ22) / Περίοδος 2018 – 2019
1η Εργασία

Στοιχεία Φοιτητή :

Όνομα : Μακρυγεώργος Νικόλαος
Α.Μ. : 1115201500238

Μεταγλώτιση και εκτέλεση :

Η μεταγλώττιση γίνεται μέσω του makefile γράφοντας make.
Διαγραφή εκτελέσιμου και αντικειμενικών αρχείων γράφοντας make clean.

Εκτέλεση γράφοντας π.χ. main 1000
Είναι απαραίτητο να δωθεί όρισμα στη γραμμή εντολής (π.χ. 100,1000,2000,...) διαφορετικά τερματίζει με μήνυμα λάθους.

Αποτελέσματα Προσομοίωσης :

Το πρόγραμμα εκτυπώνει το μέσο χρόνο κατάληψης του βαφείου,το μέσο χρόνο κατασκευής προιόντος και τα στοιχεία κάθε προιόντος που προκύπτει απο την συναρμολόγση.
Παρακάτω εκτυπώνω μόνο τους μέσους χρόνος λόγο όγκου αποτελεσμάτων.Στο πρόγραμμα που παραδίδω εκτυπώνονται και τα προιόντα μετά τους χρόνους.

Μεταγλώττιση και εκτέλεση στο linux01 του εργαστηρίου linux.

```
linux01:/home/users/sdi1500238>
linux01:/home/users/sdi1500238>make
gcc -c ask1.c
gcc -c main.c
gcc -c lista.c
gcc ask1.o main.o lista.o -o main
linux01:/home/users/sdi1500238>main 1000

Average time for painting is : 0.526000 msec.
Average time for each product is : 3.8310 msec.

linux01:/home/users/sdi1500238>main 2000

Average time for painting is : 0.541833 msec.
Average time for each product is : 3.8590 msec.

linux01:/home/users/sdi1500238>make clean
rm -f main ask1.o main.o lista.o
```

Παραδοτέα Αρχεία :

main.c	lista.c	ask1.c	lista.h
ask1.h	makefile	readme	

Περιγραφή Προγράμματος :

Για την υλοποίηση της προσομοίωσης χρησιμοποιούνται 8 διεργιασίες.Η main είναι η αρχική διεργασία και παράγει μέσω της συνάρτησης fork() 7 διεργασίες.Η main έχει το ρόλο της συναρμολόγησης,οι τρείς πρώτες που παράγονται το ρόλο των κατασκευαστών,οι τρείς επόμενες το ρόλο των ελέγχων και η τελευταία το ρόλο του βαφείου.Ο διαχωρισμός τους γίνεται ως εξής : μέσα σε ένα for που εκτελείται 7 φορές έχουμε την εντολή pid = fork() μετα την εντολή αυτη το pid για τη διεργασία παιδί θα είναι 0 και για τον γωνέα ένας αριθμός διάφορος του 0.Αρα δίνουμε στο pid που είναι 0 το δείκτη του for δηλαδή το i αρα και στον πατέρα δίνουμε 0.Δηλαδή στο τέλος το pid για κάθε διεργασία παιδί θα έχει τιμή απο 1 μέχρι 7 και ο γωνέας 0.

Έχουν χρησιμοποιηθεί 2 κοινές μνήμες και 12 σημαφόροι συνολικά.

Η πρώτη κοινή μνήμη είναι μίας θέσης,μεγέθους όσο και το εξάρτημα (struct) και είναι για την επικοινωνία μεταξύ κατασκευαστών – βαφείου και βαφείου – ελέγχων.
Η δεύτερη κοινή μνήμη είναι 3ον θέσεων,μεγέθους (3 * struct) και είναι για την επικοινωνία μεταξύ ελέγχων και συναρμολόγησης.

Οι 6 πρώτοι σημαφόροι είναι για τον συγχρωνισμό κατασκευαστές – βαφείο – ελέγχοι και οι υπόλοιποι 6 για συγχρονισμό ελέγχων – συναρμολόγσης.

Οι καθηστερήσεις που χρειάζεται γίνονται μέσω την συνάρτησης usleep() η οποία δεν παράγει busy waiting.

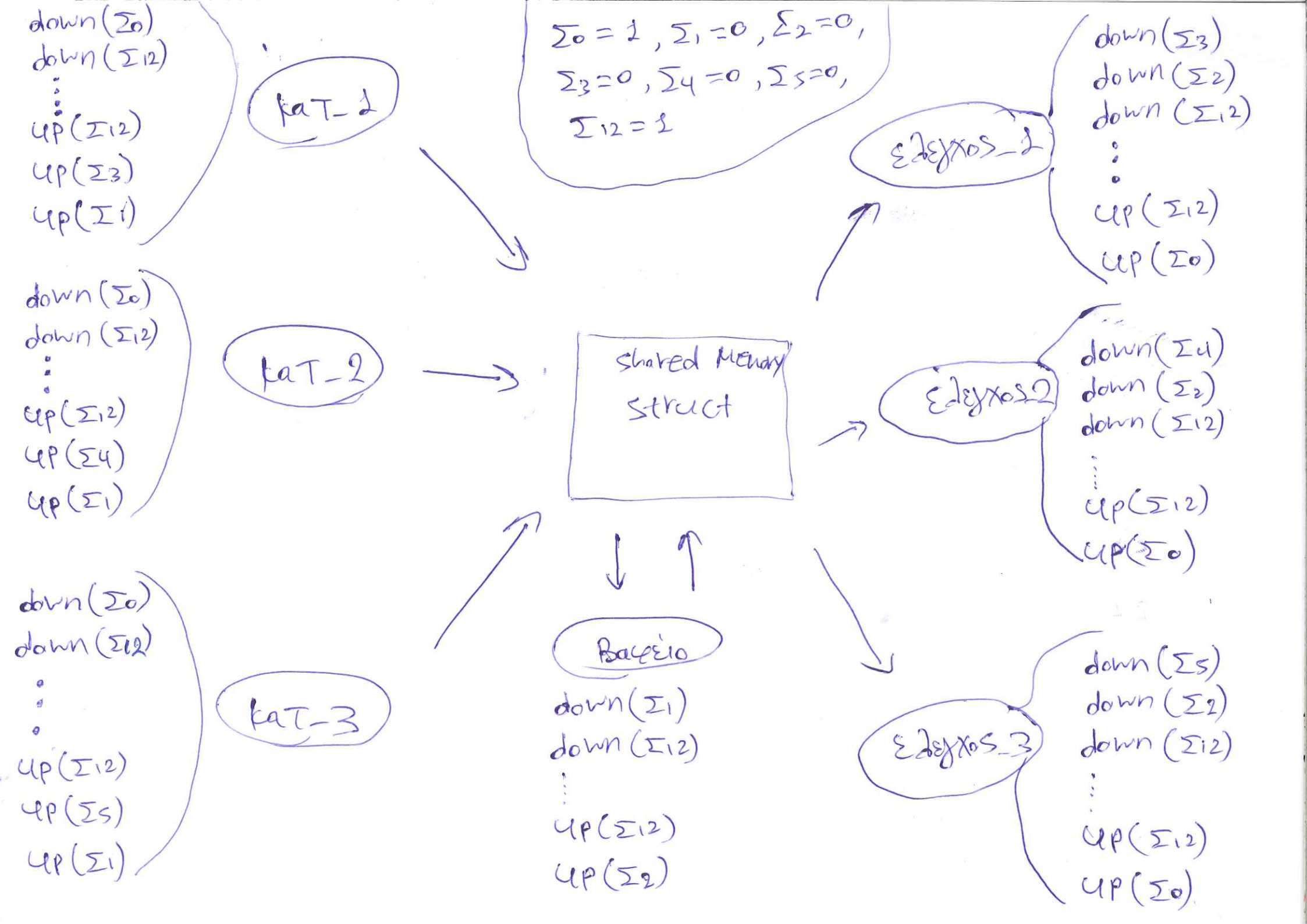
Οι 3 κατασκευαστές – παραγωγοί παράγουν απο Υ εξαρτήματα ο καθένας.Κάθε κατασκευαστής προσπαθεί να έχει πρόσβαση στην κοινή μνήμη.Οταν μπεί τελικά στην κοινή μνήμη γράφει(αποθηκεύει το εξάρτημα) στη συνέχει κάνει up() τον αντίστοιχο σημαφόρο ελέγχου για να “προετοιμαστεί” ο έλεγχος και τον σημαφόρο για το βαφείο.Το βαφείο παίρνει το εξάρτημα για επεξεργασία το ξαναδίνει απο εκεί που το πείρε και κάνει up() ένα 3ο σημαφόρο για να αναλάβει ο έλεγχος.Ο έλεγχος παίρνει το εξάρτημα το “ελέγχει” και στην συνέχεια αν μπορεί το δίνει στην δεύτερη κοινή μνήμη για να το πάρει η συναρμολόγηση διαφορετικά το αποθηκεύει σε μία λίστα.Στο τέλος κάνει up() τον σημαφόρο για να μπορέσει ο επόμενος κατασκευαστής να δώσει το εξάρτημα του.Στην περίπτωση που ο έλεγχος έχει αποθηκεύσει εξαρτήματα στην λίστα όταν τελειώσει την λήψη των εξαρτημάτων δίνει τα εξαρτήματα που έχει αποθηκεύση στην δεύτερη κοινή μνημη για να τα πάρει η συναρμολόγηση.

Η συναρμολόγηση πρώτα παίρνει το εξάρτημα τύπου 1 μετά το τύπου 2 και στο τέλος το τύπου 3. Αν δεν υπάρχει εξάρτημα τύπου 1 τότε μπλοκάρετε μέχρι ο έλεγχος να δώσει στην κοινή μνημη. Το ίδιο ισχύει για τύπου 2 και 3.Όταν πάρει όλα τα εξαρτήματα τότε αρχίζει την συναρμολόγηση. Οταν πάρει ένα εξάρτημα ανεξαρτήτου τύπου τότε κάνει up() τον κατάλληλο σημαφόρο ώστε ο έλεγχος να μπορέσει να προωθήσει το επόμενο.

Στο τέλος εκτυπώνονται ο μέσος χρόνος για την κατάληψη του βαφείου,ο μέσος χρόνος για την παραγωγή ενός προιόντος και τα τελικά προιόντα που δημιουργήθηκαν.

Έχω υλοποιήση μια συνδεδεμένη λίστα για την αποθήκευση των εξαρτημάτων σε κάθε έλεγχο.

Παρακάτω παρουσιάζω μια πρόχειρη σχηματική απεικονιση των διεργασιών κατασκευαστές-βαφείο – έλεγχοι και του συγχρονισμού με την πρώτη shared memory.


down(Σ0)
down(Σ12)
up(Σ12)
up(Σ3)
up(Σ1)
καΤ_1
Σ0 = 1, Σ1 = 0, Σ2 = 0,
Σ3 = 0, Σ4 = 0, Σ5 = 0,
Σ12 = 1
Σχεδιος_1
down(Σ3)
down(Σ2)
down(Σ12)
up(Σ12)
up(Σ0)
down(Σ0)
down(Σ12)
up(Σ12)
up(Σ4)
up(Σ1)
καΤ_2
shared Memory
struct
Σχεδιος2
down(Σ4)
down(Σ2)
down(Σ12)
up(Σ12)
up(Σ0)
down(Σ0)
down(Σ12)
up(Σ12)
up(Σ5)
up(Σ1)
καΤ_3
Βαρέιο
down(Σ1)
down(Σ12)
up(Σ12)
up(Σ2)
Σχεδιος_3
down(Σ5)
down(Σ2)
down(Σ12)
up(Σ12)
up(Σ0)