

int isNumber: Παίρνει τον αριθμό που δίνω απο το πληκτρολόγιο και εξετάζει αν είναι όντως αριθμός.

sem_wait: Αλλάζει την τιμή του σεμαφόρου από 1 σε 0, αν είναι 1 και κάνει επιτρέπρι να γίνουν οι περαιτέρω ενέργειες αλλιώς αν είναι 0 τις μπλοκάρει από το να γίνουν.

sem_signal: Αλλάζει την τιμή του σεμαφόρου από 0 σε 1 έτσι ώστε άμα κάτι περιμένει για να γίνει να του δώσει άδεια να γίνει.

sem_Init: Αρχικοποιεί τους σεμαφόρους και κάνει ένα τυπικό έλεγχο για να δει αν έγινε σωστά η αρχικοποίηση του σεμαφόρου.

CreateResources: Είναι μια συνάρτηση που έφτιαξα και καλώ στην main και αρχικοποιώ όλα τα απαραίτητα Resources δηλαδή τις 3 shared memory και τους 6 σεμαφόρους. Αρχικά τις φτιάχνω και μετά τις κάνω attach με τις αντίστοιχες εντολές shmget και shmat. Για τους σεμαφόρους τους δημιουργώ με την semget. Κάνω και κάποιους τυπικούς ελέγχους για όλα και αν πάει κάτι λάθος τότε απελευθερώνω τα Resources που έχω δημιουργήσει με κατάλληλη συνάρτηση.

FreeResources: Την χρησιμοποιώ στους ελέγχους που έκανα παραπάνω και κάθε φορά που πάει κάτι στραβά απελευθερώνει τους πόρους που έχουν ήδη φτιαχθεί δηλαδή στην περίπτωση που κάτι πάει στραβά με τον σεμαφόρο sem_Constructor θα απελευθερωθούν όλες οι shared memory και αν πάει κάτι στραβά με την δημιουργία της 3ης shared memory θα απελευθερωθεί η 1η και η 2η. Για να τις απελευθερώσω χρησιμοποιώ την shmctl.

PutinSharedMemory: Συνάρτηση που παίρνει ένα Component και το βάζει στον pointer της shared memory

takefromSharedMemory: Συνάρτηση που παίρνει το περιεχόμενο του pointer(ένα Component) της shared memory και το βάζει σε ένα άλλο Component.

Constructor: Η Constructor δέχεται ως ορίσματα το type και το n όπου ο τύπος δίνεται απο μία εξωτερική φορ που κάνει ουσιαστικά rotate τους τύπους και το n δίνεται από το πληκτρολόγιο και είναι ο αριθμός των ξεχωριστών component που φτιάχνονται για κάθε τύπο(N τύποι 1,N τύποι 2,N τύποι 3). Μέσα στην Constructor κάνω ένα φορ στο οποίο φτιάχνω τα N Component του κάθε τύπου. Ανάλογα με τον τύπο του Component κοιμάμαι και τον αντίστοιχο χρόνο και μετά φτιάχνω το Id του κάθε Component με την IdMaker και το τοποθετώ στο counter. Στη συνέχεια καλώ την ComponentConstructor με ορίσματα τον τύπο του Component,το counter και τη διεύθυνση του Component c έτσι ώστε να μπορώ να αρχικοποιήσω και τις υπόλοιπες τιμές του εντός της συνάρτησης. Μετά καλώ μία την sem_wait γιατί είμαι έτοιμος να βάλω Component στη shared memory και αν υπάρχει κάτι άλλο δεν πρέπει. Αν δεν

υπάρχει κάτι τότε βάζει το Component στη shared memory που μοιράζεται ο constructor με το painter και κάνει ουσιαστικά σινιάλο να το πάρει ο painter.

Painter: Ο painter παίρνει αν μπορεί (sem_wait) απο τη shared memory ένα Component και το “βάφει”. Στη πραγματικότητα το μόνο που κάνει είναι να αρχικοποιήσει το στοιχείο του Component paint_in_time που λέει πότε μπήκε το component στο painter. Μετά κάνω σινιάλο στο Constructor να συνεχίσει τη δικιά του διαδικασία.. Ακολούθως αν δεν έχει κάτι η shared memory(sem_wait) που μοιράζεται ο painter με το controller βάζει ένα Component και κάνει σινιάλο στο Controller να ξεκινήσει τη δικιά του διαδικασία.

Controller: Ο controller κρατά ένα counter το οποίο όταν γίνει N θα σταματήσει τη διαδικασία. Επίσης έχει ένα queue το οποίο θα κρατάει τα component τα οποία δεν μπορούν να μπουν στη shared memory που μοιράζεται ο Controller και η Composition. Κάνω δύο while, ένα το οποίο ουσιαστικά λειτουργεί μέχρι ο painter να μεταφέρει όλα τα Components στο Controller και ένα μέχρι να μεταφερθούν όλα τα Components στο Composition που βρίσκονται στο αντίστοιχο queue(ανάλογα με το τύπο τους). Μέσα το while έχει κάποιες περιπτώσεις, η πρώτη είναι αν ο τύπος του Component της shared memory είναι ίδιου τύπου με αυτό των άλλων στην ουρά, αν ναι το προσθέτω αλλιώς συνεχίζω. Η δεύτερη περίπτωση είναι αν έχουν φτάσει όλα τα Components (N) από το Βαφείο, τότε κάνει break. Μετά περιμένει και αν του δοθεί σινιάλο, κοιτάει αν το Component που έχει είναι κατάλληλο για τη shared memory που μοιράζεται ο controller με το composition το βάζει αλλιώς απλά κάνει σινιάλο στο composition και συνεχίζει το while. Αντίστοιχη διαδικασία και για το while του queue απλά εκεί σταματάει όταν αδειάσει η ουρά.

Composition: Ο Composition κρατά κάποιους χρόνους που τους χρειαζόμαστε για τελικά print όπως ο μέσος χρόνος αναμονής και ο μέσος χρόνος δημιουργίας ενός προϊόντος. Επίσης μέσα σε αυτόν βάζω ένα Component c1,c2,c3,temp_c που κρατάει ένα Component κάθε τύπου και ένα προσωρινό στο οποίο βάζω το “χαζό” Component που έχω στη shared memory για να μου προσδιορίζει ποιος τύπος Component πρέπει να μπει μετά σε αυτή(Έχω ορίσει τη σειρά που μπαίνουν να είναι 1,2,3). Αν πρέπει να μπει το 1 αντιγράφω τα στοιχεία του ,κάνω μια πρόσθεση για να κρατήσω το χρόνο που μου χρειάζεται για το μέσο χρόνο αναμονής και αλλάζω το τύπο του “χαζού” Component της shared memory σε 2 ώστε να μπει στη συνέχεια Component τύπου 2, αντίστοιχα αν είναι 2 το αλλάζω σε 3, αν είναι 3 σε 1 και μετά πάλι από την αρχή. Στο τέλος πριν ξαναξεκινήσει η διαδικασία πάλι απο το 1, βρίσκω το μέσο χρόνο δημιουργίας του προϊόντος κρατώντας το μεγαλύτερο χρόνο(αυτό που φτιάχτηκε πρώτο τελοσπάντων) και τον αφαιρώ από το χρόνο της σύνθεσης. Μετά κάνω σινιάλο στο Controller να στείλει και άλλο Component στη sharedmemory ουσιαστικά. Τέλος κάνω τα print του μέσου χρόνου αναμονής και του μέσου χρόνου προϊόντος και τελειώνω.

ComponentConstructor: Ο ComponentConstructor πέρα από το να συμπληρώνει τα $c \rightarrow \text{type}$ και $c \rightarrow \text{creation_time}$ μετατρέπει και τα $c \rightarrow \text{id}$ απο αριθμούς που είναι το counter σε string.

Main: Δήλωνω το i που ουσιαστικά είναι ο τύπος των components, το N και το status που είναι μου δείχνει ποιες διεργασίες τέλειωσαν και αν τέλειωσαν οι σωστές(3 Constructor, 1 Painter, 3 Controller, 1 Composition = 8 αν είναι 8 όλα κομπλέ). Δηλώνω το dummy Component για τη shared memory controller-composition που αργότερα θα το αρχικοποιήσω με τύπο=1. Κάνω ελέγχους για τα ορίσματα που δίνω απο το πληκτρολόγιο και εν συνέχεια φτιάχνω shared memory,semaphores με τη CreateResources και κάνω initialize τα semaphores με την InitSemaphores. Ακολουθως δίνω το dummy item στη shared memory Controller_Composition. Κάνω ένα φορ από ένα μέχρι 3(τύποι Components) όπου κάνω ένα shallow tree κρατώντας με το if (childpid==0). Μετά κάνω ένα fork() για το Painter με όρισμα το N (αλλά θα γίνει $3*N$ φορές) και ανίστοιχα για τον Έλεγχο 3 fork χρησιμοποιώντας for και ένα fork για τον Composition που θα γίνει και αυτός $3*N$ φορές. Τέλος απελευθερώνω τα πάντα και τελειώνω.