

ΕΡΓΑΣΙΑ 1^η

Στο πρόγραμμα αυτό έχουν υλοποιηθεί και τα τέσσερα στάδια, τα οποία είναι απαραίτητα για να γίνει η κατασκευή και η επεξεργασία ενός εξαρτήματος, ώστε να μπορεί να επιτευχθεί η δημιουργία του τελικού προϊόντος .

Ο χρήστης πρέπει να δίνει τον αριθμό Y (αριθμός εξαρτημάτων) από τη γραμμή εντολών, αλλιώς το πρόγραμμα θα εκτυπώσει μήνυμα λάθους στην οθόνη του.

main .c

Στο αρχείο υλοποιούνται οι απαραίτητες ενέργειες για την δημιουργία του προσομοιωτή κατασκευής προϊόντων. Πιο συγκεκριμένα δημιουργεί 8 διεργασίες παιδιά, 3 διαμοιραζόμενες μνήμες (shmem_ConstructionToPainter, shmem_PainterToChecking, shmem_CheckingToCombine), ένα σετ 8 σημαφόρων (καθώς τους αρχικοποιεί κιάλας) για τον συγχρονισμό της όλης διαδικασίας και τέλος “φροντίζει” για την αποδέσμευση τους μόλις ολοκληρωθούν οι διεργασίες παιδιά.

construction .c

Αποτελεί το πρώτο στάδιο, όπου κατασκευάζονται εξαρτήματα διαφορετικού τύπου. Παίρνουν τυχαία id's , για παράδειγμα type1xxxx, καθώς επίσης αρχικοποιούνται και τα υπόλοιπα μέλη του struct Components (int type, char id[10], int painted, int checked, double timer). Για τη ρύθμιση της επικοινωνίας για τα 3 επίπεδα κατασκευής και του βαφείου θα χρειαστούν δύο σημαφόροι (οι 0 και 1) και μία διαμοιραζόμενη μνήμη (shmem_ConstructionToPainter).

Ο σημαφόρος 1, αρχικοποιημένος σε up, χρησιμοποιείται για τον συγχρονισμό των διεργασιών κατασκευής, έτσι ώστε να γίνεται η προώθηση εξαρτήματος στο βαφείο από ένα επίπεδο κατασκευής κάθε φορά. Επομένως όποιο επίπεδο-διεργασία κατασκευής προλάβει πρώτη, κάνει Down τον σημαφόρο και αυτός θα ξαναγίνει up από το βαφείο μόλις ολοκληρώσει την επεξεργασία του εξαρτήματος ώστε να μπορέσει να προωθηθεί στο βαφείο το επόμενο εξάρτημα από κάποια διεργασία κατασκευής.

Ο σημαφόρος 0, αρχικοποιημένος σε down, χρησιμοποιείται ώστε το βαφείο να διαβάζει ένα εξάρτημα από την μνήμη μόνο όταν κάποια διεργασία κατασκευής έχει προωθήσει ένα εξάρτημα προς το βαφείο. Έτσι όταν μια διεργασία κατασκευής γράψει στην μνήμη το εξάρτημα, κάνει up τον σημαφόρο 0 επιτρέποντας στο βαφείο να το διαβάσει.

painter .c

Στο βαφείο γίνεται η επεξεργασία του εξαρτήματος, αφού το παραλάβει από το στάδιο κατασκευής. Μόλις ολοκληρώσει την επεξεργασία του εξαρτήματος το προωθεί στο στάδιο ελέγχου. Για να επιτευχθεί αυτό θα χρειαστούν δυο διαμοιραζόμενες μνήμες (shmem_ConstructionToPainter, shmem_PainterToChecking) και 6 σημαφόροι (0, 1, 2, 3, 4, 5). Οι σημαφόροι 0 και 1 ρυθμίζουν την την επικοινωνία των διεργασιών κατασκευής με το βαφείο (όπως αυτό αναφέρθηκε στο construction.c).

Για την επιτυχημένη προώθηση του εξαρτήματος από το βαφείο, μόλις ολοκληρώσει την επεξεργασία του, στο στάδιο ελέγχου χρησιμοποιείται ο σημαφόρος 2 ο οποίος είναι αρχικοποιημένος up. Μόλις γράψει το εξάρτημα στην μνήμη (shmem_PainterToChecking), ο σημαφόρος 2 γίνεται down ώστε να μην επιτραπεί να γράψει στην μνήμη το επόμενο εξάρτημα προτού το στάδιο ελέγχου παραλάβει το τρέχον εξάρτημα. Μόλις το στάδιο ελέγχου παραλάβει το εξάρτημα ενεργοποιεί τον σημαφόρο 2 σε up.

Οι σημαφόροι 3, 4, 5 είναι αρχικοποιημένοι down και λειτουργούν και οι τρεις με τον ίδιο τρόπο. Πιο συγκεκριμένα ο σημαφόρος 3 αφορά το επίπεδο ελέγχου 1, ο σημαφόρος 4 το επίπεδο ελέγχου 2 και ο σημαφόρος 5 το επίπεδο ελέγχου 3. Έτσι μόλις το βαφείο γράψει στην μνήμη (shmem_PainterToChecking), ανάλογα με το τι τύπου είναι είναι το εξάρτημα ενεργοποιεί έναν εκ των σημαφόρων 3,4 ή 5 σε up ώστε να μπορέσει η διεργασία ελέγχου να διαβάσει το εξάρτημα που της αναλογεί. Ο κύκλος αυτός εκτελείται 3*Υ φορές, λαμβάνοντας έτσι όλα τα εξαρτήματα από κάθε επίπεδο κατασκευής.

checkinglevel .c

Στο σημείο αυτό πραγματοποιείται η επικοινωνία του τμήματος ελέγχου τόσο μεταξύ του βαφείου όσο και του σταδίου συναρμολόγησης. Θα χρειαστούν 2 σημαφόροι για την επικοινωνία του σταδίου ελέγχου με το βαφείο. Το στάδιο ελέγχου αποτελείται από 3 επίπεδα. Έτσι σε κάθε επίπεδο αντιστοιχεί ένας σημαφόρος (3, 4, 5) ανάλογα με την διεργασία που τρέχει, και ο σημαφόρος 2, όπως αυτό εξηγήθηκε στο βαφείο. Επίσης θα χρειαστούν 2 διαμοιραζόμενες μνήμες (shmem_PainterToChecking, shmem_CheckingToCombine).

Στην συνέχεια μόλις το τμήμα ελέγχου επεξεργαστεί το εξάρτημα, το προωθεί στο τμήμα συναρμολόγησης. Για τον συγχρονισμό μεταξύ τους χρησιμοποιούνται 2 σημαφόροι (6, 7).

Ο σημαφόρος 6 είναι αρχικοποιημένος up, και εξασφαλίζει ώστε να γράφει στην μνήμη (shmem_CheckingToCombine) ένα εξάρτημα μία διεργασία ελέγχου κάθε φορά. Ο σημαφόρος 7 είναι αρχικοποιημένος down, κάνοντας το τμήμα συναρμολόγησης να περιμένει έως ότου κάποια διεργασία ελέγχου γράψει στην μνήμη το επεξεργασμένο εξάρτημα, η οποία (διεργασία ελέγχου) ενεργοποιεί τον σημαφόρο σε up.

combine .c

Τέλος, το επίπεδο συναρμολόγησης είναι εκείνο το οποίο θα συναρμολογήσει τα εξαρτήματα ανα τριάδες για να φτιάξει τα προϊόντα.

Όσα εξαρτήματα λαμβάνει αποθηκεύονται σε λίστες που εκπροσωπεί κάθε λίστα έναν από τους τρεις τύπους εξαρτημάτων. Να αναφέρουμε, σε αυτό το σημείο, ότι οι λίστες είναι απαραίτητες για να μην υπάρξει αδιέξοδος. Ενώ, λοιπόν, διαβάζει τα διάφορα εξαρτήματα κάθε φορά από τη μνήμη, τα αποθηκεύει και ελέγχει πάντοτε εάν στις λίστες υπάρχουν μέσα αντικείμενα. Εάν το μέγεθος και των τριών ουρών είναι διάφορο του μηδενός, σημαίνει πως έχει ό,τι απαιτείται για να δημιουργήσει το προϊόν. Κάθε φορά που συμβαίνει αυτό, από την ουρά διαγράφονται τα αντικείμενα που έχουν χρησιμοποιηθεί.

Το πρόγραμμα εκτυπώνει τα παρακάτω.

\$ make

\$/main 4

A product with id : type11o38type21o38type31o38 has been created successfully in 6.001617 seconds!

A product with id : type1wg0ztype2wg0ztype3wg0z has been created successfully in 6.002516 seconds!

A product with id : type1y4tjtype2y4tjtype3y4tj has been created successfully in 6.001780 seconds!

Painter was occupied for : 25.014315 seconds

A product with id : type1883mtype2883mtype3883m has been created successfully in 6.001605 seconds!