Κ22: Λειτουργικά Συστήματα

Ονοματεπώνυμο: Κωνσταντίνος Πασχόπουλος

Αριθμός Μητρώου: 1115201500127

configfile:

Στο configfile υπάρχουν οι πληροφορίες για την δομή του λιμανιού, δηλαδή πόσες θέσεις υπάρχουν για την κάθε κατηγορία πλοίου και πόσο κοστίζει η στάθμευση σε αυτές τις θέσεις. Ο αριθμός των θέσεων θα πρέπει να έχει την μορφή:

spaces «Τύπος πλοίου» «Αριθμός διαθέσιμων θέσεων»

Το κόστος της κάθε θέσης θα πρέπει να έχει την μορφή:

cost «Τύπος πλοίου» «Κόστος στάθμευσης»

myport.c:

Η διεργασία myport αρχικά αναλαμβάνει την δημιουργία και αρχικοποίηση της κοινής μνήμης, καθώς και την αρχικοποίηση των απαραίτητων σηματοφόρων. Ανάλογα με το πόσες θέσεις χρειάζεται το κάθε λιμάνι δεσμεύει τον απαραίτητο χώρο στην κοινή μνήμη για να χωράει τον τμήμα του Public ledger που δείχνει την τωρινή κατάσταση του λιμανιού (parking_spaces). Στη συνέχεια μέσω πολλαπλών fork δημιουργεί όλες τις διεργασίες που θα λάβουν μέρος στο λιμάνι, όπως ο port-master, ο monitor και τα διάφορα vessels στα οποία δίνει τυχαία ορίσματα. Τέλος δίνει σήμα στις διεργασίες του port-master και του monitor να ολοκληρώσουν την λειτουργία τους και φροντίζει να καταστρέψει τους σηματοφόρους, την κοινή μνήμη και ό,τι άλλους πόρους χρησιμοποίησε.

portmaster.c:

Ο port-master είναι ο μόνος που γράφει στον Public ledger. Ο Public ledger αποτελείτε από δύο μέρη, το παρόν και το παρελθόν. Στην υλοποίηση μου επέλεξα το παρελθόν του

λιμανιού να το αποθηκεύω τοπικά στον port-master σε μία συνδεδεμένη λίστα και να το εκτυπώνω όταν ο port-master ολοκληρώνει την λειτουργία του. Το παρόν υπάρχει μέσα στην κοινή μνήμη και μπορούν οι υπόλοιπες διεργασίες να το διαβάζουν όποτε χρειάζονται. Ο port-master λειτουργεί μέχρι να του έρθει κάποιο σήμα για να σταματήσει, αυτό το σήμα είναι όταν η μεταβλητή vessel_action στην κοινή μνήμη γίνει -1. Τότε σταματάει να δέχεται καινούργια πλοία και περιμένει μέχρι όλα τα πλοία που βρίσκονται μέσα στο λιμάνι έχουν αναχωρήσει για να σταματήσει να λειτουργεί. Κάθε φορά που ο port-master κάνει ένα loop κάνει sem_wait στο σηματοφόρο portmaster και περιμένει εκεί μέχρι να τον «ξυπνήσει» κάποιο πλοίο. Αν το πλοίο έχει κάνει την μεταβλητή vessel_action 0 σημαίνει ότι θέλει να παρκάρει κάπου στο λιμάνι. Αν την κάνει 1 σημαίνει ότι θέλει να φύγει από το λιμάνι. Όταν κάποιο πλοίο ζητάει να παρκάρει ο port-master κοιτάζει πως μπορεί να το εξυπηρετήσει. Αν το λιμάνι δεν υποστηρίζει πλοία του συγκεκριμένου τύπου κάνει την μεταβλητή portmaster_action 0 για να το διώξει. Αν δεν υπάρχει αυτή τη στιγμή κάποια διαθέσιμη θέση για αυτό το πλοίο την κάνει 1 και κάνει continue προκειμένου να μην αφήσει κάποιο άλλο πλοίο να μπει στην θέση του. Είναι δηλαδή first come first serve και περιμένουν τα πλοία στην ουρά μέχρι το πλοίο που είναι πρώτο στην ουρά έχει βρει θέση και έχει μπει στο λιμάνι. Αν βρει μια διαθέσιμη θέση κάνει την portmaster_action 2 και ενημερώνει τον Public ledger. Στην περίπτωση που κάποιο πλοίο έχει βρει θέση ο port-master μπορεί να εξυπηρετήσει το επόμενο πλοίο που έχει κολλήσει στην ουρά, για αυτό κάνει sem_post στον σηματοφόρο approaching με τον οποίο έχω προσομοιώσει την FIFO ουρά. Αν κάποιο πλοίο θέλει να φύγει ενημερώνει τον Public ledger, ενημερώνει το ιστορικό του λιμανιού και ενημερώνει τα στατιστικά του λιμανιού. Και στις δύο περιπτώσεις ο port-master κάνει sem_wait στον σηματοφόρο port, για να σιγουρευτεί ότι κανένα πλοίο δεν κινείται μέσα στο λιμάνι.

vessel.c:

Αρχικά κάθε vessel κολλάει στον σηματοφόρο approaching. Η δουλειά αυτού του σηματοφόρου είναι να προσομοιώσει μια FIFO ουρά. Στη συνέχεια χρησιμοποιώ τον σηματοφόρο mutex με τον οποίο το κάθε πλοίο «κλειδώνει» την κοινή μνήμη για να μπορέσει να γράψει εκεί τα στοιχεία του, τα οποία θα διαβάσει ο port-master και θα απαντήσει αναλόγως. Αφού το κάνει αυτό κάνει sem_post στον σηματοφόρο portmaster έτσι ώστε να ειδοποιήσει τον port-master και μετά περιμένει στον σηματοφόρο answer μέχρι να αποφασίσει ο port-master τι θα κάνει το πλοίο. Αφού του απαντήσει ανοίγει την κοινή για να μπορέσουν να την χρησιμοποιήσουν και άλλες διεργασίες και κάνει sem_post τον portmaster για να του δείξει ότι διάβασε το μήνυμα του. Αν του απαντήσει στην μεταβλητή portmaster_action 0 τότε κάνει detach την κοινή μνήμη και φεύγει. Αν η μεταβλητή είναι 1 τότε περιμένει στον σηματοφόρο stuck_vessel και συνεχίζει όταν ο port-master κάνει sem_post σε αυτόν τον σηματοφόρο επειδή κάποια θέση έχει ελευθερωθεί. Αυτό το έκανα για να αποφύγω το busy waiting, καθώς το πλοίο ρωτάει τον port-master που να παρκάρει μόνο

όταν ελευθερώνεται κάποια θέση στο λιμάνι. Αν του απαντήσει 2 τότε συνεχίζει την λειτουργία του και μπαίνει στο λιμάνι. Αφού ολοκληρώσει την μανούβρα του για να πάει στην θέση παρκινγκ κάνει sem_post στην μεταβλητή port έτσι ώστε να δείξει ότι δεν κινείται πια. Για λόγους ευκολίας όταν περάσει ένα δευτερόλεπτο πάει στον Public ledger που βρίσκεται στην κοινή μνήμη και βρίσκει ποια είναι η τωρινή του χρέωση. Τέλος ζητάει με παρόμοιο τρόπο από τον port-master να φύγει.

myfunctions.c:

Εδώ έχω κάποιες βοηθητικές συναρτήσεις με τις οποίες διαχειρίζομαι το logfile που χρησιμοποιούν όλες οι διεργασίες και τον public ledger που βρίσκεται τοπικά στον port-master.