PROGETTO SO

* Processo master
* Processo nave
* Processo porto
* Utility file

PROCESSO MASTER

Si occupa di generazione processi e del report.

Inizializza l’array di struct che forma la mappa tramite apposita funzione esposta dall’Utility file, genera i processi nave/porto e “giornalmente” crea un report.

Il report sarà costruito così:

* Il master crea una pipe(lettura per master,scrittura per navi/porto), i processi porto/nave scrivono sulla pipe in un formato da decidere le informazioni che servono al master per il report ogni fine di un giorno, una volta ricevute il master stampa il report quotidiano(viene usata una wait(giornoSimulato) e poi stampato il report con le informazioni delle pipe).

PROCESSO NAVE

Inizialmente viene inizializzata una struct contenente le informazioni relative alla nave(velocità, posizione, capacità,merci a 0).

Appena creata, la nave effettua una ricerca all’interno della shared memory contenente le posizioni e informazioni del porto e sceglie il porto più vicino a lei, con la maggior quantità di merci offerte/richieste che la nave soddisfa e con la possibilità di accedere ad almeno una banchina.

Questa specifica ricerca va fatta appena la nave viene creata e appena la nave supera il carico/scarico in un porto, verrà utilizzato un boolean “searchPort” per indicare quando effettuare la ricerca.

Successivamente lo spostamento verrà fatto incrementando la posizione ottenuta da questa ricerca con l’attesa specificata per la nanosleep() fino al raggiungimento del porto desiderato.

Attraccata alla banchina la nave effettua la negoziazione con il porto, **NON** quando decide verso che porto andare.

PROCESSO PORTO

-Leggi nel documento del progetto, è l’unica parte che si capisce a prescindere uagliun

UTILITY FILE

* Possiede una funzione che genera la mappa, ovvero una SM composta da un array di struct che per ogni cella contiene una struct così composta:
* lato(per la grandezza della mappa),
* coordinata x,
* coordinata y,
* PID porto,
* Ogni altra informazione relativa al porto;
* Possiede una funzione che permette di verificare se la posizione della nave sia identica alla posizione di un porto e se può essere effettuato il carico/scarico;

COMUNICAZIONE TRA PROCESSI

Il processo nave e porto comunicheranno attraverso una memoria condivisa. I processi porto dopo aver inizializzato la memoria condivisa portArray si metteranno in attesa che si liberi il semaforo semStartComunicationId,siccome quando sarà sbloccato avranno accesso ad una memoria condivisa(creata con shmId del porto che attende) su cui il porto leggerà ciò che le navi hanno scritto per lui e finita questa procedura i porti scriveranno nella stessa cella della memoria condivisa le informazioni che servono alla nave così che lei possa leggerle, infine i porti si riposizionano in attesa del semaforo e le navi partono verso un altro porto.