Progetto SO 2022/23
Transito navale di merci

Versione provvisoria

Bini, Radicioni, Schifanella C.

December 2022

Indice

1 Composizione gruppo di studenti

Il progetto sarà sviluppato da un gruppo, composto da 1 a **massimo 3 da componenti**. Il gruppo dovrebbe essere composto da studenti dello **stesso turno**, i quali discuteranno con il docente del proprio turno. È tuttavia consentita anche la discussione del progetto di laboratorio da parte di un gruppo di studenti di turni diversi. In questo caso, **tutti** gli studenti del gruppo discuteranno con **lo stesso docente**. Esempio: Tizio (turno T1) e Caio (turno T2) decidono di fare il progetto insieme. Lo consegnano e vengono convocati dal prof. Sempronio il giorno X. Tale giorno Tizio e Caio si presentano e ricevono entrambi una valutazione dal Prof. Sempronio (docente del T1, anche se Caio fa parte del turno T2, il cui docente di riferimento è il prof. Calpurnio).

2 Consegna

Il progetto è costituito da:

1. il codice sorgente
2. una breve relazione che sintetizzi le scelte progettuali compiute

Il progetto si consegna compilando la seguente Google Form, cui si accede con credenziali istituzionali,

- `https://forms.gle/kDDaGRoW5mrdhi696`

la quale richiederà il caricamento di:

- progetto stesso (un unico file in formato .tgz o .zip NON .rar) e
- cognome, nome, matricola, email di ciascun componente del gruppo.

Dopo il caricamento del progetto, verrete convocati dal docente con cui discuterete (si veda Sezione 1 in caso di gruppo composto da studenti di turni diversi). Attenzione: **compilare una sola form per progetto** (e **non una per ogni componente del gruppo**). Una eventuale ulteriore consegna prima dell'appuntamento **annullerà la data dell'appuntamento**.

La consegna deve avvenire almeno **10 giorni prima** degli appelli scritti per dare modo al docente di pianificare i colloqui:

- se consegnate con almeno 10 giorni di anticipo rispetto alla data di un appello, il docente propone una data per la discussione entro l'appello seguente
- altrimenti, la data sarà successiva all'appello seguente.

Esempio: per avere la certezza di un appuntamento per la discussione di progetto entro l'appello del 24/01/2023, si deve consegnare entro le ore 24:00 del **14/01/2023**.

3 Valutazione e validità

Il progetto descritto nel presente documento potrà essere discusso se consegnato **entro il 30 Novembre 2023**. Dal Dicembre 2023 sarà discusso il progetto assegnato durante l'anno accademico 2023/24.

Tutti i membri del gruppo devono partecipare alla discussione. La valutazione del progetto è **individuale** ed espressa in 30-esimi. Durante la discussione

- verrà chiesto di illustrare il progetto;
- verranno proposti quesiti sul programma "Unix" del corso anche non necessariamente legati allo sviluppo del progetto.

È necessario ottenere una votazione di almeno **18** su 30 per poter essere ammessi allo scritto. In caso di superamento della discussione del progetto, la votazione conseguita consentirà di partecipare allo scritto per i **cinque appelli successivi** alla data di superamento. Non sono ammesse eccezioni o deroghe a tale periodo di validità.

In caso di mancato superamento, lo studente si potrà ripresentare soltanto dopo almeno **un mese** dalla data del mancato superamento

Si ricorda che il voto del progetto ha un peso di $\frac{1}{4}$ sulla votazione finale di Sistemi Operativi.

4 Pubblicazione del proprio progetto, plagio, sanzioni

Copiare altri progetti o parte di essi impedisce una corretta valutazione. Per questo motivo, gli studenti che consegnano il progetto sono consapevoli che:

- la condivisione con altri gruppi attraverso canali pubblici o privati (a titolo di esempio: google drive, canali telegram, github, etc.) di tutto o parte del progetto non è consentita fino a tutto Novembre 2023;
- la copiatura di tutto o parte del progetto non è consentita;
- eventuali frammenti di codice estratti da parti di codice visto a lezione o da altri repository pubblici devono essere opportunamente dichiarati.

Nel momento in cui uno studente non rispettasse le sopra citate norme di comportamento, dopo essere stato convocato ed aver avuto modo di illustrare la propria posizione, potrà essere oggetto delle seguenti sanzioni:

- se lo studente avrà nel frattempo superato l'esame di Sistemi Operativi anche successivamente alla data di discussione del progetto, la verbalizzazione del voto verrà annullata
- se lo studente avrà soltanto superato la discussione del progetto ma non l'esame, la valutazione del progetto verrà annullata e lo studente non potrà accedere ad ulteriore discussione di progetto prima dei due appelli successivi alla data di evidenza di copiatura.

5 Descrizione del progetto: versione minima (voto max 24 su 30)

Si intende simulare il traffico di navi cargo per il trasporto di merci di vario tipo, attraverso dei porti. Questo viene realizzato tramite i seguenti processi

- un processo *master* incaricato di creare gli altri processi e gestire la simulazione, ove necessario;
- un numero `SO_NAVI` (≥ 1) di processi *nave*; e
- un numero `SO_PORTI` (≥ 4) di processi *porto*.

Nella descrizione del progetto si farà riferimento a:

- il *tempo simulato*, ovvero il tempo che trascorre nella simulazione (esempio: un giorno per trasportare una merce)
- il *tempo reale*, ovvero il tempo di durata dell'esecuzione della simulazione (esempio: dopo l'avvio, la simulazione termina dopo 30 secondi, pur avendo simulato una durata di 30 giorni).

Nella simulazione, un giorno del tempo simulato dura un secondo di tempo reale.

5.1 Le merci

Nella simulazione esistono `SO_MERCI` tipi diversi di merce. Se ritenuto utile, si può identificare ogni tipo di merce con un nome oppure con un identificativo numerico. Un lotto di ogni tipo di merce è caratterizzato da:

- peso del lotto (in ton), estratto casualmente fra 1 e `SO_SIZE` all'inizio della simulazione e
- tempo di vita (in giorni), estratto casualmente fra `SO_MIN_VITA` e `SO_MAX_VITA`, misurato in *giorni*.

Offerta e domanda di merci vengono generate presso i porti (si veda descrizione in seguito). Ogni volta che una merce di un certo tipo viene generata a run-time, avrà sempre le stesse caratteristiche di cui sopra.

Quando il tempo di vita di una merce è trascorso, la merce viene persa e svanisce da ovunque essa sia (nave o porto). Il tempo di vita di una merce è relativo al momento della sua creazione presso un certo porto.

Tutta la merce generata durante la simulazione si classifica in:

- merce presente in un porto (disponibile per il carico)
- merce presente su una nave
- merce consegnata ad un porto che la richiede
- merce scaduta in porto
- merce scaduta in nave

5.2 La mappa

La mappa del mondo è rappresentata da un quadrato di lato `SO_LATO` di tipo `(double)`, misurata in kilometri (Terra piatta). Una posizione sulla mappa è rappresentata dalla coppia di coordinate (come su piano cartesiano). Sia porti che navi hanno una posizione sulla mappa. La navigazione fra due punti della mappa può sempre avvenire in linea retta: non esistono grandi masse terrestri che richiedono di essere aggirate.

5.3 Il processo nave

Ogni nave ha

- una *velocità* `SO_SPEED` misurata in kilometri al giorno (identica per tutte le navi),
- una *posizione*, ovvero una coppia di coordinate di tipo `(double)` all'interno della mappa,
- una *capacità* `SO_CAPACITY` (in ton) che misura il carico totale trasportabile (identica per tutte le navi). Una nave non può trasportare una quantità di merci che ecceda la propria capacità.

Le navi nascono da posizioni casuali e senza merce. Non c'è limite al numero di navi che possono occupare una certa posizione sulla mappa, mentre il numero di navi che possono svolgere contemporaneamente operazioni di carico/scarico in un porto è limitato dal numero di banchine.

Lo spostamento della nave da un punto ad un altro è realizzato tramite una `nanosleep` che simuli un tempo simulato di

$$\frac{\text{distanza fra posizione di partenza e di arrivo}}{\text{velocità di navigazione}}.$$

Nel calcolare questo tempo, si tenga conto anche della parte frazionaria (per esempio, se si deve dormire per 1.41 secondi non va bene dormire per 1 o 2 secondi). Si ignorino le eventuali collisioni durante il movimento sulla mappa.

Le navi possono sapere tutte le informazioni sui porti: posizione, merci offerte/richieste, etc. Ogni nave, in maniera autonoma:

- si sposta sulla mappa
- quando si trova nella posizione coincidente con il porto, può decidere se accedere ad una banchina e, se questo avviene, può decidere lo scarico o il carico della merce ancora non scaduta.

La negoziazione fra nave e porto su tipologia e quantità di merci da caricare o scaricare avviene in un momento a discrezione del progettista (prima di partire per la destinazione, quando arrivata nel porto, altro, etc.)

Le navi non possono comunicare fra loro, né sapere il contenuto di quanto trasportato dalle altre navi.

5.4 Il processo porto

Il porto è localizzato in una certa posizione della mappa e gestisce un numero casuale di banchine compreso fra 1 e `SO_BANCHINE`. Esistono sempre almeno quattro porti (`SO_PORTI`≥ 4), uno per ogni angolo della mappa.

Quando i processi porto vengono creati, viene creata casualmente anche domanda e offerta di merci presso di essi. Sia la domanda che l'offerta totale di tutte le merci presso tutti i porti è pari a `SO_FILL` (in ton). Sia offerta che richiesta di merci sono caratterizzate da

- il tipo di merce

- la quantità di merce.

Non ci può essere sia offerta che richiesta di una stessa merce nello stesso porto. Non appena la merce è stata creata viene marcata con la propria data di scadenza allo scadere dalla quale, la merce risulta inutilizzabile e viene dichiarata sprecata.

Le banchine del porto sono gestite come risorse condivise protette da semafori che impediscano l'utilizzo da parte di un numero di navi maggiore delle banchine presenti. Quando una nave raggiunge un porto (sia per carico che scarico) chiede l'accesso ad una banchina. Se lo ottiene, tiene occupata (con `nanosleep(...)`) la banchina per un tempo pari a

$$\frac{\text{quantità di merce scambiata (in ton)}}{\text{velocità carico/scarico (in ton/giorno)}}$$

con "velocità" uguale al parametro `SO_LOADSPEED`, misurato in ton/giorno. La velocità di carico e scarico è identica. Quando viene scaricata una merce di cui c'è richiesta, allora tale richiesta viene soddisfatta e conteggiata come tale.

5.5 Dump stato simulazione

Al trascorrere di ogni giorno, deve essere visualizzato un report provvisorio contenente:

- Totale delle merci suddivise per tipologia e stato (disponibile, consegnato, etc. Si veda la Sezione 5.1 per la descrizione dello stato di una merce)
- Numero di navi:
  - in mare con un carico a bordo,
  - in mare senza un carico,
  - in porto, facendo operazioni di carico/scarico.
- Per ogni porto si indichi
  - la quantità di merce presente, spedita, e ricevuta
  - il numero di banchine occupate/totali

5.6 Terminazione della simulazione

La simulazione termina in una delle seguenti circostanze

- dopo un tempo simulato di `SO_DAYS`
- quando per ogni tipo di merce
  - l'offerta è pari a zero oppure
  - la richiesta è pari a zero.

Il report finale deve indicare:

- Numero di navi ancora in mare con un carico a bordo
- Numero di navi ancora in mare senza un carico
- Numero di navi che occupano una banchina
- Totale delle merci suddivise per tipologia e stato (disponibile, consegnato, etc. Si veda la Sezione 5.1 per la descrizione dello stato di una merce)
- Per ogni porto si indichi la quantità di merce
  - presente, spedita, e ricevuta.

- Per ogni tipo di merce si indichi:
  - la quantità totale generata dall'inizio della simulazione e quanta di essa
    - è rimasta ferma in porto
    - è scaduta in porto
    - è scaduta in nave
    - è stata consegnata da qualche nave.
  - Si indichi il porto che
    - ha offerto la quantità maggiore della merce
    - e quello che ha richiesto la quantità maggiore di merce.

6 Descrizione del progetto: versione "normal" (max 30)

La generazione delle merci avviene durante tutto il corso della simulazione (e non solo all'inizio). All'inizio di ogni giorno, viene generata una quantità `SO_FILL/SO_DAYS` di merce di tipo casuale presso porti casuali. Si ricordi sempre che non ci possono essere sia domanda che offerta della stessa merce in uno stesso porto.

Inoltre, si intende simulare alcune problematiche che possono rendere più difficoltose le operazioni di navigazione, carico e scarico delle merci. Un nuovo processo chiamato *meteo* si occupa di generare in maniera randomica una serie di inconvenienti:

- tempesta: ogni giorno colpisce casualmente una nave in movimento, fermandola sul posto per `SO_STORM_DURATION` ore
- mareggiata: ogni giorno colpisce casualmente un porto, fermando tutte le operazioni per `SO_SWELL_DURATION` ore
- maelstrom: ogni `SO_MAELSTROM` ore, una nave viene affondata e il suo eventuale carico è disperso.

I dump dello stato della simulazione e il report finale daranno conto degli effetti meteo sul traffico marittimo, specificando (oltre a quanto elencato nelle Sezioni 5.5 e 5.6):

- quante navi sono state rallentate dalla tempesta
- quali porti sono stati interessati dalla mareggiata
- quante navi sono state affondate a causa del maelstrom.

In aggiunta a quanto descritto in Sezione 5.6, la simulazione termina anche quando tutte le navi sono affondate per il maelstrom.

7 Configurazione

Tutti i parametri di configurazione sono letti a **tempo di esecuzione**, da file, da variabili di ambiente, o da `stdin` (a discrezione del progettista). Quindi, un cambiamento dei parametri non deve determinare una nuova compilazione dei sorgenti.

La seguente tabella elenca valori per alcune configurazioni di esempio da testare. Si tenga presente che il progetto deve poter funzionare anche con altri parametri fattibili.

Configurazione	Parametri															
	SO_NAVI	SO_PORTI	SO_MERCI	SO_SIZE (ton)	SO_MIN_VITA (giorni)	SO_MAX_VITA (giorni)	SO_LATO (km)	SO_SPEED (km/giorno)	SO_CAPACITY (ton)	SO_BANCHINE	SO_FILL (ton)	SO_LOADSPEED (ton/giorno)	SO_DAYS (giorni)	SO_STORM_DURATION (ore)	SO_SWELL_DURATION (ore)	SO_MAELSTROM (ore)
"dense, small ships"	1K	100	1	1	50	50	1K	500	10	2	500K	200	10	6	24	1
"as above + trashing"	1K	100	10	1	3	10	1K	500	10	2	500K	200	10	6	24	1
"Born to run"	10	1K	100	100	3	10	1K	2K	1K	10	1M	500	10	6	24	60
"cargos, big stuff"	100	5	10	100	3	10	1K	500	1K	10	1M	200	10	6	24	24
"unlucky cargos"	100	5	10	100	3	10	1K	500	1K	10	1M	200	10	12	10	1

Ci possono essere valori dei parametri di configurazione che generano comportamenti bizzarri. In questi casi è sufficiente che gli studenti siano in grado di identificare tali comportamenti e suggerire delle configurazioni dei parametri che siano in grado di rendere il comportamento più naturale.

8 Linee guida per la valutazione

- Minimizzazione della quantità di merci sprecate
- Tempo di CPU impegnato (evitare attese attive)

9 Requisiti implementativi

Il progetto (sia in versione "minimal" che "normal") deve

- utilizzare almeno memoria condivisa, semafori e un meccanismo di comunicazione fra processi a scelta fra code di messaggi o pipe,
- essere realizzato sfruttando le tecniche di divisione in moduli del codice (per esempio, i vari processi devono essere lanciati da eseguibili diversi con `execve(...)`),
- essere compilato mediante l'utilizzo dell'utility `make`
- massimizzare il grado di concorrenza fra processi
- deallocare le risorse IPC che sono state allocate dai processi al termine del gioco
- essere compilato con almeno le seguenti opzioni di compilazione:

  ```
  gcc -std=c89 -Wpedantic
  ```

- poter eseguire correttamente su una macchina (virtuale o fisica) che presenta parallelismo (due o più processori).

Per i motivi introdotti a lezione, ricordarsi di definire la macro `_GNU_SOURCE` o compilare il progetto con il flag `-D_GNU_SOURCE`.