

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 PROVA SCRITTA DI SISTEMI OPERATIVI
 ANNO ACCADEMICO 2021/2022
 06 settembre 2022

Esercizio -1: Essere iscritti su AlmaEsami per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione in tutti i fogli prima di svolgere ogni altro esercizio. Scrivere esclusivamente a penna senza abrasioni. E' vietato l'uso delle penne cancellabili, della matita, dei coprenti bianchi per la correzione (bianchetto) e la scrittura in colore rosso (riservato alla correzione).

Il compito e' formato da tre fogli, sei facciate compresa questa. Le soluzioni che si vogliono sottoporre per la correzione devono essere scritte negli spazi bianchi di questi fogli. Non verranno corretti altri supporti.

E' obbligatorio consegnare il compito, e' possibile chiedere che esso non venga valutato scrivendo "NON VALUTARE" in modo ben visibile nella prima facciata.

Per svolgere questo compito occorre solo una penna e un documento di identità valido. La consultazione o anche solo la disponibilità di altro materiale comporterà l'annullamento del compito (verrà automaticamente valutato gravemente insufficiente).

Esercizio c.1: I bit di un numero intero rappresentano condizioni di un sistema. Se lo stato attuale è 6 (0110) vuole dire che attualmente sono vere le condizioni 2 (0010) e 4 (0100).

Scrivere un monitor bitcond che fornisca le seguenti procedure entry:

`void set(int bit2set);` accende nello stato attuale i bit di bit2set

`void unset(int bit2unset)` spegne nello stato attuale i bit di bit2unset

`void statuswait(int bit2wait)` attende che lo stato attuale soddisfi tutti le condizioni indicate in bit2wait (cioè che tutti i bit in bit2wait siano accesi nello stato attuale).

Le richieste `statuswait` devono essere servite in ordine FIFO (cioè un processo anche se sono presenti tutte le condizioni necessarie deve attendere se un processo che ha chiamato `statuswait` prima è in attesa).

Lo stato iniziale è zero (nessuna risorsa disponibile)

Esercizio c.2: Usando i semafori implementare un servizio che preveda due funzioni:

`void sumstop(int v)`

`int sumgo(void)`

La funzione `sumstop` deve mettere il processo chiamante in attesa.

La funzione `sumgo` deve sbloccare tutti i processi messi in attesa con la `sumstop` e restituire la somma algebrica dei valori passati come parametro alla `sumstop` dai processi che sono stati sbloccati (zero se la `sumgo` viene richiamata quando non c'è nessun processo bloccato).

Esercizio g.1: Sia dato questo programma:

Program P:

```
for (i=0; i<2; i++) {
    long_compute();
    io();
}
short_compute();
```

`long_compute` impiega 5 ms, `short_compute` 2 ms e `io` impiega 4 ms. Il programma usa un device condiviso gestito in modalità FIFO.

Considerando in un sistema di elaborazione biprocessore dove sono in esecuzione quattro istanze del programma P che sono state

attivate ai tempi 0, 2ms, 4ms e 7ms e che il sistema usa uno scheduler round robin per l'accesso alla CPU disegnare il diagramma di Gantt

dell'esecuzione e spiegarne i passaggi. (time slice=3 ms)

Esercizio g.2: rispondere alle seguenti domande (motivando opportunamente le risposte):

a) Perché il compilatore C è scritto in C? Come si fa a compilare il compilatore C se per compilarlo occorre il compilatore C?

b) Come si calcola la lunghezza massima di un file in un file system di tipo UNIX?

c) Perché può essere difficoltoso revocare le capability di accesso a risorse?

d) Qual è la differenza fra una funzione di libreria (e.g. `sprintf`) e una system call (e.g. `write`).