

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 PROVA SCRITTA DI SISTEMI OPERATIVI
 ANNO ACCADEMICO 2020/2021
 23 giugno 2021

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: Scrivere il monitor `delay` con una sola procedure entry:

```
int delay(int value);
```

Il monitor deve sospendere i primi `NDELAY` processi che chiamano la `delay`. Le successive chiamate di `delay` devono mantenere costante il numero di processi sospesi, ogni successiva chiamata deve riattivare il primo processo in attesa prima di sospendersi, la `delay` ritorna il valore passato come parametro dal processo che ne ha riattivato l'esecuzione. e.g. se `NDELAY` è 2:

P1: delay(44) -> P1 si sospende

P2: delay(40) -> P2 si sospende

P3: delay(42) -> P1 si riattiva e ritorna 42, P3 si sospende

P4: delay(22) -> P2 si riattiva e ritorna 22, P4 si sospende

Esercizio c.2: Implementare usando semafori ordinari (fair, fifo) un servizio di semafori a priorità lifo che fornisca le seguenti primitive:

```
void PLP(int prio);
```

```
void PLV();
```

PLP e PLV si devono comportare rispettivamente come P e V. Quando la PLV deve riattivare un processo sceglie fra quelli in attesa quello a priorità massima, nel caso siano presenti più processi a priorità massima sceglie quello in attesa da **meno** tempo.

Esercizio g.1: Lo scheduler ED è a priorità dinamica, preemptive e gestisce processi periodici (processi che si riattivano periodicamente, allo scadere di un periodo se ne attiva una nuova istanza). Lo scheduler sceglie il processo ready che ha la fine del proprio periodo più prossima. Sono processi che fanno solo calcolo. Ai fini di questo esercizio il costo dei context switch sia considerato nullo.

Si considerino due sistemi. Sistema A: sono presenti 3 processi:

Processo	Periodo	tempo di elaborazione
p1	15	5
p2	20	5
p3	30	6

Sistema B: sono presenti 3 processi:

Processo	Periodo	tempo di elaborazione
p1	8	2
p2	16	4
p3	32	16

I processi del sistema A sono schedulabili con ED? (si può costruire uno schedule di durata indefinita che consenta di eseguire tutti i processi). E i processi del sistema B?

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

- perché DMA viene utilizzato per le unità di memoria secondaria (es dischi) e non per terminali?
- cosa succede in un sistema operativo quando un processo utente tenta di eseguire una operazione illegale (es. divisione per zero)? Lo standard POSIX (UNIX) cosa prevede in questo caso?
- L'algoritmo di rimpiazzamento second chance (detto anche dell'orologio) è a stack? Perché viene preferito a LRU?
- Quali sono le caratteristiche negative dell'uso delle Access Control List per la memorizzazione delle informazioni di autorizzazione.

