

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 PROVA SCRITTA DI SISTEMI OPERATIVI  
 ANNO ACCADEMICO 2020/2021  
 26 maggio 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:** Un buffer sincrono strampalato (bss) ha due procedure entry:

```
void put(T value)
list of T get(void)
```

La entry put viene utilizzata per aggiungere un elemento e la entry get per leggere tutti quelli disponibili.

Se più processi chiamano la put quando nessun processo è in attesa per una get, rimangono tutti bloccati.

Quando successivamente un processo chiama la get riceve la lista di tutti gli elementi inseriti con le put e tutti i processi vengono sbloccati.

Se il buffer è vuoto tutti i processi che chiamano la get rimangono bloccati. quando un processo chiama successivamente la put tutti i processi in attesa per la get si sbloccano e ricevono lo stesso valore di ritorno: una lista contenente il solo valore passato come parametro alla put.

**Esercizio c.2:** Dato un servizio di message passing asincrono implementare un *servizio* di message passing asincrono a selezione di mittenti (somp) senza fare uso di processi server.

Il servizio somp ha due primitive:

```
sasend(message , destination)
sarecv(senders)
```

dove senders è un insieme.

la sarecv restituisce il primo messaggio che uno dei processi in senders ha spedito al processo ricevente usando la sasend (o spedito da qualsiasi processo se senders è vuoto).

**Esercizio g.1:** Lo scheduler RM è a priorità statica, preemptive e gestisce processi periodici (processi che si riattivano periodicamente, allo scadere di un periodo se ne attiva una nuova istanza). Sono processi che fanno solo calcolo. Al fine di questo esercizio il costo del context switch sia considerato nullo.

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

Processo	Periodo	Priorità	tempo di elaborazione
p1	15	massima	5
p2	20	media	5
p3	25	minima	6

Sistema B: sono presenti 3 processi:

Processo	Periodo	Priorità	tempo di elaborazione
p1	8	massima	2
p2	16	media	4
p3	32	minima	16

I processi del sistema A sono schedulabili con RM? (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:

a) perché una situazione di trashing si può risolvere sospendendo l'esecuzione di processi?

b) come fa il sistema operativo a sospendere i processi che sono in attesa del completamento di una richiesta di I/O?

c) quale tipo di crittografia è necessaria per implementare un servizio di autorizzazione basato su capability? perché? (NB la domanda non fa riferimento alle POSIX capability)

d) Perché con due dischi si preferisce usare RAID1 piuttosto che RAID5? (infatti RAID5 è definito con un minimo di 3 dischi).

