

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
PROVA SCRITTA DI SISTEMI OPERATIVI
ANNO ACCADEMICO 2024/2025
28 maggio 2025

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 è 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, è 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 `maxlenbb` che fornisce le seguenti procedure entry:

```
void maxlenbb.add(int bufnum, objtype obj)
```

```
objtype maxlenbb.get(void)
```

Il monitor gestisce `N` buffer ognuno dei quali contiene `NELEM` elementi.

La funzione `add` inserisce `obj` come elemento nel buffer indicato da `bufnum`.

La funzione `get` cerca il buffer che contiene più elementi e da questo toglie in ordine FIFO un elemento e lo restituisce.

In caso più buffer abbiano lo stesso numero massimo di elementi viene scelto il buffer di indice minore.

Esercizio c.2: dato un sistema di message passing asincrono implementare *facendo uso di un processo server un servizio* di message passing asincrono anti stalker che fornisce le seguenti funzioni:

```
bool asasend(msg_t msg, pid_t dest)
```

```
msg_t asarecv(pid_t sender)
```

La funzione `asasend` restituisce `true` se la spedizione ha successo, `false` se fallisce.

Un processo `S` viene considerato stalker da un destinatario `D` se ci sono almeno 7 messaggi spediti da `S` a `D` che non sono ancora stati ricevuti. Ogni ulteriore tentativo di `S` di spedire messaggi a `D` fallisce (`asasend` restituisce `false`) e il messaggio viene perso.

Hint: non è necessario che il server smista i messaggi, può limitarsi a contare quelli in attesa.

Esercizio g.1: Sia dato un sistema monoprocesso con due unità di I/O. La CPU viene gestita tramite uno scheduler preemptive a priorità statica. La priorità dei processi è usata anche per l'accesso alle unità di I/O.

Esistono nel sistema tre tipi di processi periodici (che vengono riattivati allo scadere di ogni periodo).

P1: priorità massima: 1ms CPU, 1ms I/O (unità 2), periodo=4ms

P2: priorità media. 1ms CPU, 1ms I/O (unità 1), periodo=2ms

P3: priorità minima: 1ms CPU, 2ms I/O (unità 2), 1ms CPU, 2ms I/O (unità 1), periodo=x

Trovare per quali valori di `x` è possibile calcolare uno schedule di durata infinita (o meglio indefinita) e produrre il diagramma di Gantt della soluzione con il minimo valore di `x`. Spiegare il procedimento seguito per trovare la soluzione.

(nota: il tempo di inizio dei periodi dei processi può essere scelto opportunamente)

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

- perché è difficile revocare un diritto di accesso se espresso come capability che revocare lo stesso diritto in una access control list?
- si può implementare la memoria virtuale con paginazione a richiesta in un sistema che ha una Memory Management Unit priva di TLB (Translation lookaside buffer)?
- Perché non si può implementare la funzionalità di link fisico di file in un file system di tipo FAT?
- L'algoritmo del Banchiere evita che si verifichino situazioni di deadlock. Sembra una funzione desiderabile eppure viene raramente usato nei sistemi reali, perché?