Nome_____ Cognome _____N. di matricola (10 cifre) _____ Riga ____ Col ____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
PROVA SCRITTA DI SISTEMI OPERATIVI
ANNO ACCADEMICO 2023/2024
15 gennaio 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 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: Il monitor mbb gestisce un servizio di bounded buffer con scrittura e lettura di elementi multipli.
mbb ha due procedure entry:
void write(obj_t *data, int ndata)
void read(obj_t *data, int ndata)
il parametro "data" è un vettore di elementi da scrivere o leggere. Il parametro ndata indica il numero di elementi del vettore. (le richieste devono essere gestite in modo FIFO). Il buffer contiene al massimo MAX elementi di tipo obj_t.

Esercizio c.2: Sia dato un servizio di message passing asincrono broadcast che fornisce due funzioni:
void absend(msg_type msg)
msg_type abrecv(void)
I messaggi spediti con la absend vengono ricevuti da tutti i processi e la abrecv riceve i messaggi spediti da ogni mittente. I messaggi di ogni mittente vengono ricevuti il ordine FIFO.
E’ possibile con il servizio a diffusione implementare un servizio di message passing asincrono senza fare uso di processi server? Se sì fornire l’implementazione, se no (di)mostrare l’impossibilità.

Esercizio g.1: Siano dati due processi in esecuzione in un sistema monoprocessore e gestiti da uno scheduler round-robin. I due processi PA e PB sono gli unici nel sistema iniziano contemporaneamente l'esecuzione e usano la stessa unità di I/O gestita in modo FIFO.
PA: 2ms CPU, 1ms I/O, 1ms CPU, 6ms I/O, 1ms CPU
PB: 1ms CPU, 1ms I/O, 6ms CPU, 1ms I/O, 1ms CPU
Qual è il tempo minimo e quale il tempo massimo impiegato dal sistema per completare l’elaborazione dei due processi al variare della lunghezza del quanto di tempo e della posizione iniziale dei processi nella ready queue (PA precede PB o viceversa).

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

a) Un i-node di un file system tipo ext2 per un errore viene riportato un numero di link maggiore del reale. Cosa può succedere se si continua ad usare il file system? (perché?) E se il numero di link errato fosse al contrario inferiore al reale cosa potrebbe succedere? (perché?)
b) Perché è necessario usare spinlock in sistemi multiprocessore per implementare kernel di tipo simmetrico (SMP)?
c) Perché nei sistemi reali l’algoritmo di rimpiazzamento second chance (orologio) viene preferito a LRU sebbene il primo non sia a stack e il secondo sì?
d) Perché revocare un’autorizzazione espressa come capability è più difficile che revocare lo stesso diritto quando espresso come access control list?