

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 PROVA SCRITTA DI SISTEMI OPERATIVI
 ANNO ACCADEMICO 2023/2024
 29 maggio 2024

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 `urgentsem` che implementi un servizio di semafori con supporto di P urgente. il monitor ha tre procedure entry e un costruttore:
`void init(int initvalue)`
`void P(void)`
`void UP(void)`
`void V(void)`
 Vale l'invariante $initvalue + NV \geq NP + NUP$ (dove NX indica il numero di processi che hanno completato l'operazione X). UP è una variante della P di tipo urgente. Quando rispettando l'invariante sia possibile scegliere fra un processo in attesa per una chiamata P e uno in attesa per UP questo ultimo deve essere riattivato. Fra processi in attesa per P così come fra processi in attesa per UP la riattivazione deve avvenire in ordine FIFO.

Esercizio c.2: Facendo uso di semafori scrivere una funzione `syncvalue`:
`int syncvalue(int key)`
 la funzione `syncvalue` è sempre bloccante. Quando il valore del parametro `key` è diverso da quello della precedente chiamata il processo prima di bloccarsi riattiva tutti i processi in attesa. Il valore di ritorno è il numero di processi con lo stesso valore `key` sbloccati. Per esempio:
 P chiama `syncvalue(42)`, si blocca.
 Q chiama `syncvalue(42)`, si blocca.
 R chiama `syncvalue(44)` sblocca P e Q poi si blocca. Il valore di ritorno per P e Q è 2.
 T chiama `syncvalue(46)`, sblocca R che ritorna 1 e si blocca.
 Q chiama `syncvalue(46)`, si blocca.
 P chiama `syncvalue(46)`, si blocca
 V chiama `syncvalue(0)`, sblocca T, Q e P (valore di ritorno: 3) poi si blocca...

Esercizio g.1: Sia dato un file system FAT con 32 blocchi. La tabella FAT e' la seguente (FF è il terminatore)
 (la prima riga è l'indice della tabella, la seconda il contenuto)

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
FF	FF	03	04	0F	01	07	08	FF	0B	FF	0C	0D	0E	0F	FF	09	1D	13	11	14	1B	17	15	19	1A	06	FF	10	1E	1F	1C

La directory radice è nel blocco 00 e la testa della lista dei blocchi liberi è 02. Questi sono i contenuti di alcuni blocchi:
 blocco 00: directory: fileA->05 dir->0A fileE->18
 blocco 01: 'XXXXXXXXXXXX'
 blocco 02: 'YYYYYYYYYYYY'
 blocco 05: 'ZZZZZZZZZZZ'
 blocco 0A: directory: fileB->12, fileC->16, fileD->05
 1) Disegnare l'albero dei contenuti (file e directory)
 2) Qual è il contenuto del file denominato fileA?
 3) Quanti blocchi compongono il file che ha nome fileC?
 4) In file system è incoerente per 3 errori, quali sono e come possono essere corretti?

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

- Un controller DMA per i dischi migliora le prestazioni di un servizio di memoria virtuale con paginazione a richiesta?
- Il meccanismo del sale (salt) serve per evitare gli attacchi di tipo "dizionario". Perché un "sale" di un solo byte può non essere sufficiente?
- Per migliorare la gestione dei processi interattivi un progettista decide di implementare uno scheduler round robin dimezzando la durata del prossimo time slice a ogni processo ogni volta che lo consuma completamente senza fare I/O. Quali problemi potrebbero emergere?
- Che differenza c'è fra deadlock prevention e deadlock avoidance? Il metodo dell'allocazione gerarchica è un metodo di prevention o avoidance? perché?