

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
PROVA SCRITTA DI SISTEMI OPERATIVI
ANNO ACCADEMICO 2021/2022
01 giugno 2022

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 che fornisce due procedure entry:

```
int wait_tick(int nticks)
void tick(void)
```

La procedure entry tick è pensata per essere richiamata periodicamente (es. ogni secondo o ora o giorno) da un processo.

Quando un processo chiama la wait_tick deve attendere un numero di chiamate della tick pari al parametro nticks. Per esempio se un processo chiama wait_tick(2) deve fermarsi e verrà riattivato alla seconda successiva chiamata di tick.

La funzione wait_tick ha come valore di ritorno il numero di processi che erano bloccati al momento della tick che ha sbloccato il chiamante.

Esempio: P chiama wait_tick(2) e si blocca. Q chiama wait_tick(3) e si blocca. T chiama tick() non succede nulla. R chiama wait_tick(2) e si blocca. T chiama tick(), viene sbloccata la wait_tick di P e il valore ritornato è 3. T chiama tick(), vengono sbloccate le wait_tick di Q e R e il valore ritornato per entrambi i processi è 2

Esercizio c.2: Un servizio di message passing asincrono non fifo (nfasend/nfarecv) consegna in tempo finito tutti i messaggi spediti ma non è garantito che i messaggi vengano ricevuti nell'ordine nel quale sono stati spediti.

```
void nfasend(msg_t msg, pid_t dest)
msg_t nfarecv(pid_t sender)
```

Dato un servizio di message passing asincrono non fifo scrivere una libreria che implementi il servizio di message passing asincrono fifo:

```
void asend(msg_t msg, pid_t dest)
msg_t arecv(pid_t sender)
```

Nota: sia il servizio dato (non fifo) sia quello da implementare (fifo) consentono la ricezione solo da mittente specificato (non supportano ANY/*).

Esercizio g.1: Si consideri l'algoritmo del banchiere a tre valute. Si prenda in considerazione una situazione nella quale tre processi p1, p2, p3 abbiano un massimo credito disponibile rispettivamente di (4, 2, 4), (4, 4, 2), (2, 4, 4), cioè per esempio p1 può prendere in prestito 4 unità della prima e della terza valuta ma solo 2 della seconda. In un certo istante p1 ha 2 unità della prima valuta, p2 ha 2 unità della seconda, p3 ha 4 unità della terza e nessuna altra risorsa è stata assegnata. Qual è il capitale iniziale minimo che consente allo stato di essere safe? (considerare tutti i casi possibili e spiegare bene il procedimento).

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

- Perché viene usata la paginazione per implementare la memoria virtuale?
- Perché lo scheduler round robin è poco adatto alle elaborazioni batch?
- Perché il file system FAT è più efficiente di un file system che utilizzi l'allocazione concatenata coi puntatori nei blocchi dati?
- Quale tipo di attacco viene reso arduo dal meccanismo del sale (salt) per la codifica delle password. Perché?