

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI
ANNO ACCADEMICO 2010/2011 - PROVA DI CONCORRENZA

13 maggio 2011

Esercizio -1: essersi iscritti correttamente 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 due fogli, quattro 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.

Esercizio 1: Scrivere un monitor reqq che gestisca Ncode di richieste.

I richiedenti chiamano la funzione che ha la seguente signature:

```
answer_t reqq.query(request_t request, int type);
```

Query deve fermare il processo richiedente fino a completamento della richiesta da parte di un gestore. Il valore di ritorno e' la risposta del gestore.

Ci sono N gestori, uno per ogni tipo di richiesta, che si comportano come segue:

```
multiq_handler: process[i, i=0,...,N-1] {
    request_t req;
    int type;
    while (1) {
        req=reqq.getquery(i, &type);
        reqq.reply(i,handle(req, type));
    }
}
```

Normalmente le richieste vengono assegnate al gestore del tipo corrispondente ($i==type$), se non vi sono richieste di tale tipo elabora una richiesta del tipo con la piu' lunga coda di richieste pendenti. Quando un gestore termina l'elaborazione (funzione handle, da non implementare!) invia il risultato tramite la reply. Il valore passato alla reply deve essere restituito al richiedente come valore di ritorno della funzione query. Se non vi sono richieste disponibili i gestori si fermano attendendo nuove richieste.

Esercizio 2: Facendo uso di semafori ordinari implementare le funzioni lifocs_enter e lifocs_exit che realizzino una critical section LIFO.

I processi per usare la critical section LIFO usano il seguente protocollo:

```
lifocs_enter()
```

```
... codice critico
```

```
lifocs_exit()
```

quando un processo rilascia la sezione critica e ci sono processi in attesa di entrare, deve venir assegnata la sezione critica al processo che ha fatto l'ultima richiesta (in ordine di tempo).

Esercizio 3: Un servizio di message passing a canale prevede che ogni spedizione e ricezione di un messaggio faccia riferimento a "canale" e non ad un processo.

L'interfaccia e' la seguente:

```
n=channel_join_create(id); /* crea un canale con l'id specificato se non esiste, altrimenti incrementa il numero dei processi che stanno usando il canale, restituisce il numero di processi che condividono il canale. l'operazione e' atomica */
```

```
channel_send(id,msg); /* spedisce in modo asincrono un messaggio sul canale id */
```

```
msg=channel_receive(id); /* riceve un messaggio sul canale specificato, se piu' processi si fermano in attesa di un messaggio dallo stesso canale, ogni messaggio viene recapitato al processo che per primo ha richiesto la channel_receive */
```

```
channel_leave_delete(id); /* decrementa il numero dei processi utilizzatori del canale, il canale viene cancellato quando non ha piu' utilizzatori. L'operazione e' atomica. */
```

Implementare un supporto per sezioni critiche che faccia uso del message passing a canale. (le funzioni da implementare sono ccenter, csexit).

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI
 ANNO ACCADEMICO 2010/2011 – PARTE GENERALE
 20 maggio 2011

Esercizio -1: essersi iscritti correttamente 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 due fogli, quattro facciate compresa questa. Le soluzioni che si vuole 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 (con la stessa penalizzazione di punteggio della grave insufficienza per la prossima esercitazione scritta).

Esercizio 1: Siano dati una memoria con 4 frame {0,1,2,3} e un unico processo che usa 5 pagine {0,1,2,3,4} eseguito tramite memoria virtuale (con paginazione a richiesta) su tale memoria.

a- Usando l'algoritmo di rimpiazzamento LRU si costruisca la più corta stringa di riferimenti che inizi con 4,0,1,2 e che al termine abbia la pagina 0 nel frame 0, la pagina 1 nel frame 1, la pagina 2 nel frame 2, la pagina 3 nel frame 3, o si dimostri che tale stringa non esiste.

b- Usando l'algoritmo di rimpiazzamento FIFO si costruisca la più corta stringa di riferimenti che inizi con 4,0,1,2 e che al termine abbia la pagina 0 nel frame 0, la pagina 1 nel frame 1, la pagina 2 nel frame 2, la pagina 3 nel frame 3, o si dimostri che tale stringa non esiste.

c- Usando l'algoritmo di rimpiazzamento FIFO si costruisca la più corta stringa di riferimenti che inizi con 0,1,3,4 e che al termine abbia la pagina 0 nel frame 0, la pagina 1 nel frame 1, la pagina 2 nel frame 2, la pagina 3 nel frame 3, o si dimostri che tale stringa non esiste.

Esercizio 2: Sia dato il seguente stato dell'algoritmo del Banchiere Multivaluta

Valuta A: COH=x p1: credito=10 assegnati=5 p2: credito=5 assegnati=3 p1: credito=10 assegnati=4	Valuta B: COH=y credito=6 assegnati=4 credito=6 assegnati=3 credito=6 assegnati=2
--	--

Mostrare per quali valori (x, y) lo stato sopra indicato risulta safe.

Esercizio 3: Detti rispettivamente R il numero di riga e C il numero di colonna, soddisfare la richiesta corrispondente a $(R+C)\%9$ (esercizio obbligatorio):

- 0) Spiegare il funzionamento di un Device Driver e in particolare mostrare come operano in sistemi con e senza DMA
- 1) Spiegare il funzionamento di un Memory Manager in un sistema con paginazione a richiesta
- 2) Spiegare il funzionamento di uno Scheduler ed analizzare quali algoritmi di scheduling siano meglio adatti a diverse classi di processi
- 3) Spiegare e descrivere gli algoritmi usati per ottimizzare l'uso della memoria secondaria.
- 4) Spiegare e descrivere gli algoritmi usati per la deadlock avoidance
- 5) Spiegare quali metodi di allocazione possano essere usati nell'implementazione dei file system
- 6) Spiegare le Access Control List e le Capability, mostrare le differenze e i campi di applicazione
- 7) Spiegare la differenza fra thread e processi e come si possa implementare sistemi con supporto multithreading.
- 8) Spiegare i metodi che vengono usati per controllare la coerenza dei dati presenti in un file system

NOTA: tutti gli esercizi verranno valutati solo se le risposte saranno corredate da motivazioni e dimostrazioni scritte in Italiano o in Inglese (corretto o almeno comprensibile). La presenza nella soluzione di un esercizio di solo codice sorgente/tabelle/scarabocchi/simboli vari comporta la non valutazione dell'esercizio.

