

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2007/2008
 CONCORRENZA - 16 Gennaio 2008

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

Esercizio 1: Il guardaroba russo

Per accedere alle sale di un noto museo russo e' obbligatorio lasciare la giacca al guardaroba, dove lavora un solo dipendente. Per lasciare o ritirare la propria giacca e' necessario mettersi in attesa in un'unica estenuante coda. Per consegnare la giacca si attende il proprio turno in coda, si attende il guardarobiere, si consegna la giacca e si attende che questo ci consegni un contrassegno. Per ritirare la giacca si attende il proprio turno in coda, si attende il guardarobiere, si consegna il contrassegno e si attende la giacca. Il guardaroba ha capacita' finita e puo' contenere solo MAX giacche. Quando il guardaroba e' pieno, il guardarobiere fa passare in testa il primo cliente in coda che deve ritirare l'indumento.

a) Scrivere il monitor **guardaroba_russo** tenendo presente la seguente vita dei processi:

// un item e' una union fra una giacca e un contrassegno

```

process guardarobiere() {
    while(1) {
        item x = guardaroba_russo.prendi();
        << scambia x (una giacca/contrassegno) con
            y (un contrassegno/giacca) >>
        guardaroba_russo.dai(y);
    }
}

process visitatore(giacca g) {
    contrassegno c = guardaroba_russo.consegna(g);
    << visita il museo >>
    g = guardaroba_russo.ritira(c);
}

```

b) Cosa potrebbe succedere se il guardarobiere non facesse passare avanti chi deve ritirare le giacche quando il guardaroba e' pieno?

Esercizio 2:

Implementare un meccanismo di message passing sincrono facendo uso di semafori:

```
void ssend(T msg,pid_t dest)
```

```
void sreceive(T *pmsg,pid_t sender)
```

Il sistema provvede la chiamata getpid per poter leggere l'identita' del processo corrente.

Per la receive il mittente e' sempre dichiarato in modo esplicito (non e' previsto il caso di sreceive *)

Esercizio 3:

La seguente funzione puo' fare le veci della Test&Set per la realizzazione di sezioni critiche?

```
F(x,y)=<x1=x; x = y % 2; y = y % x1>
```


UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2007/2008
 PARTE GENERALE - 16 Gennaio 2008

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

Esercizio 1: Consideriamo un sistema caratterizzato da: 4 frame di memoria virtuale, condivisi fra tutti i processi; politica LOOK di gestione del disco; politica LRU di rimpiazzamento delle pagine in memoria virtuale. Supponiamo che in un determinato momento ogni pagina logica n sia mantenuta su disco nel blocco n . Infine supponiamo che i frame di memoria virtuale siano inizialmente non occupati, che i tempi di CPU burst siano trascurabili e che i processi $P1$ e $P2$ vengano lanciati in parallelo, ove le stringhe di riferimento di $P1$ e $P2$ sono le seguenti:

$P1$: R4, R4, W5, R2, R8

$P2$: R3, R9, W7, R10, R3

Mostrare l'evoluzione del sistema con una sequenza di colonne simili alla seguente:

contenuto frame 1 e altre info utili (e.g. dirty bit, caricamento in corso)

contenuto frame 2 e altre info utili (e.g. dirty bit, caricamento in corso)

contenuto frame 3 e altre info utili (e.g. dirty bit, caricamento in corso)

contenuto frame 4 e altre info utili (e.g. dirty bit, caricamento in corso)

posizione testina e direzione

coda direzione corrente

coda prossima direzione

Esercizio 2:

1) Si consideri un banchiere B che lavora con due valute $V1$ e $V2$ e due banchieri $B1$ e $B2$ che lavorano rispettivamente solo con $B1$ e solo con $B2$ (ignorando l'altra valuta).

1.a) E' vero che uno stato safe per B e' safe sia per $B1$ che per $B2$? Fornire una prova o un controesempio

1.b) E' vero che uno stato safe sia per $B1$ che per $B2$ e' safe anche per B ? Fornire una prova o un controesempio

2) Si consideri un banchiere che risponda a richieste di tre tipi:

ACQUIRE(v,n) richiede di acquisire n risorse della classe (valuta) v

RELEASE(v,n) richiede di rilasciare n risorse della classe (valuta) v

MAX(v,n) dichiara che il numero massimo di risorse della classe (valuta) v di cui necessiterà e' n ; deve essere invocata almeno una volta prima della prima ACQUIRE; può essere invocata in qualunque momento per dichiarare un nuovo massimo; tuttavia se il banchiere ritiene che lo stato raggiunto divenga unsafe ha il diritto di uccidere il processo recuperando le risorse assegnate; dichiarare un nuovo massimo inferiore al numero di risorse di classe v in possesso e' un errore

Mostrare come si comporta il banchiere con COH iniziale (8,8) a seguito delle seguenti richieste:

$P1$:MAX(0,5) $P2$:MAX(0,5) $P3$:MAX(0,5) $P1$:MAX(1,3) $P1$:ACQUIRE(1,2) $P2$:ACQUIRE(0,4) $P3$:ACQUIRE(0,3)

$P3$:ACQUIRE(0,1) $P2$:MAX(1,4) $P2$:ACQUIRE(1,3) $P2$:MAX(0,4) $P1$:ACQUIRE(0,3) $P2$:MAX(0,6)

NOTA: dopo ogni richiesta al banchiere mostrare la coda dei processi sospesi, il COH e le risorse già assegnate e ancora da richiedere per ogni processo. Mostrare inoltre tutti i controlli di safety compiuti dal banchiere.

Esercizio 3:

Sia x l'ultima e y la penultima cifra del vostro numero di matricola e sia $n = (y*10 + x)\%4$. Elencare tutti gli interrupt e le trap gestiti dal componente n del sistema operativo, illustrando anche le interazioni (dirette o indirette) con l'hardware collegate alla gestione di tali interrupt e trap:

0) memory manager

1) disk manager

2) scheduler

3) file system manager



