

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2003/2004
MIDTERM PARTE GENERALE - 16 Dicembre 2005

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:

Si consideri uno scheduler a priorit  dinamica con 32 livelli di priorit  e time slice pari a 3ms. A ogni processo sono associate una base priority e una current priority. La current priority e' inizialmente uguale alla base priority e viene indicata al momento della creazione del processo. Un processo che termini il proprio time slice decrementa di 1 la sua priorit , senza mai scendere sotto la sua base priority. Un processo che venga risvegliato riceve un incremento di priorit  che dipende dal motivo del risveglio: +1 disco, +6 tastiera.

Sia data la storia esecutiva dei seguenti processi:

P1: creazione a t=0ms, basepriority = 16;

CPU 5ms; disco 2ms; CPU 5ms; disco 2ms; CPU 1ms; tastiera 1s; CPU 2ms; disco 3ms

P2: creazione a t=6ms, basepriority = 4;

CPU 1ms; tastiera 1s; CPU 30ms;

P3: creazione a t=40ms, basepriority = 8;

CPU 35ms; disco 8ms; CPU 2ms;

Esercizio 2:

Dato il seguente brano di codice in linguaggio C:

```
#define MAX 10
void fooswap(struct elem v[])
{
    register int i,j;
    struct elem tmp;
    for (i=0,j=MAX-1; i<j; i++,j--) {
        tmp=v[i];
        v[i]=v[j];
        v[j]=tmp;
    }
}
```

considerato che la struct elem sia lunga esattamente quanto una pagina di memoria (1024 byte nella macchina presa in considerazione), scrivere la sequenza dei riferimenti in memoria, e mostrare come l'algoritmo MIN si comporterebbe prendendo in considerazione una memoria di 4 frame. (Non considerare la gestione della memoria per cio' che riguarda il codice eseguibile, ne' ovviamente per le variabili mappate nei registri ma solamente per la gestione dei dati e dello stack).

Esercizio 3:

Sia **x** l'ultima cifra del vostro numero di matricola e **y** la penultima cifra del vostro numero di matricola. Rispondete alla domanda $(y*10+x)\%6$

- 0) Spiegare un meccanismo di deadlock prevention
- 1) Spiegare un meccanismo di deadlock avoidance
- 2) Spiegare un meccanismo per aumentare l'affidabilit  dei sistemi di memorizzazione in memoria secondaria
- 3) Spiegare un meccanismo per evitare il trashing
- 4) Spiegare un meccanismo per la verifica di coerenza di un file system
- 5) Spiegare un meccanismo hardware per la protezione della memoria

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2003/2004
MIDTERM PARTE GENERALE - 16 Dicembre 2005

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:

Si consideri uno scheduler a priorit  dinamica con 32 livelli di priorit  e time slice pari a 3ms. A ogni processo sono associate una base priority e una current priority. La current priority e' inizialmente uguale alla base priority e viene indicata al momento della creazione del processo. Un processo che termini il proprio time slice decrementa di 1 la sua priorit , senza mai scendere sotto la sua base priority. Un processo che venga risvegliato riceve un incremento di priorit  che dipende dal motivo del risveglio: +1 disco, +6 tastiera.

Sia data la storia esecutiva dei seguenti processi:

P1: creazione a t=10ms, basepriority = 16;

CPU 5ms; disco 2ms; CPU 5ms; disco 2ms; CPU 1ms; tastiera 1s; CPU 2ms; disco 3ms

P2: creazione a t=0ms, basepriority = 4;

CPU 1ms; tastiera 1s; CPU 30ms;

P3: creazione a t=50ms, basepriority = 8;

CPU 35ms; disco 8ms; CPU 2ms;

Esercizio 2:

Dato il seguente brano di codice in linguaggio C:

```
#define MAX 10
void fooswap(struct elem v[])
{
    register int i,j;
    struct elem tmp;
    for (i=0,j=MAX-1; i<j; i++,j--) {
        tmp=v[i];
        v[i]=v[j];
        v[j]=tmp;
    }
}
```

considerato che la struct elem sia lunga esattamente quanto una pagina di memoria (1024 byte nella macchina presa in considerazione), scrivere la sequenza dei riferimenti in memoria, e mostrare come l'algoritmo LRU si comporterebbe prendendo in considerazione una memoria di 4 frame. (Non considerare la gestione della memoria per cio' che riguarda il codice eseguibile, ne' ovviamente per le variabili mappate nei registri ma solamente per la gestione dei dati e dello stack).

Esercizio 3:

Sia x l'ultima cifra del vostro numero di matricola e y la penultima cifra del vostro numero di matricola. Rispondete alla domanda $(y*10+x)\%6$

0) Spiegare un meccanismo di deadlock prevention

1) Spiegare un meccanismo di deadlock avoidance

2) Spiegare un meccanismo per aumentare l'affidabilit  dei sistemi di memorizzazione in memoria secondaria

3) Spiegare un meccanismo per evitare il trashing

4) Spiegare un meccanismo per la verifica di coerenza di un file system

5) Spiegare un meccanismo hardware per la protezione della memoria