

## Esami di Laboratorio Sistemi Operativi del 5/9/2008

### Esercizio 1(20/30) (Analisi e scelte fatte 8 ,Gestione Thread e sincronizzazione 8 ,Codice 4)

In un piccolo museo per motivi di sicurezza è permesso l'accesso ad un numero  $K$  limitato di persone. Inoltre nella stanza principale, in cui è conservato il gioiello del museo, è consentito l'accesso singolo. I visitatori seguono il seguente iter:

1. Arrivano al museo in gruppino da 1 a 10 con scadenza randomica da  $t$  a  $4*t$ ;
2. Se è possibile accedere al museo tutti insieme cominciano a visitare il museo in un tempo randomico  $t_1$  da 50 ms a 200 ms , viceversa aspettano.
3. Se disponibile ogni singolo visitatore del gruppo visita la stanza principale in un tempo  $t_2$  da 10ms a 40ms.
4. Infine continuano la visita per un tempo randomico  $t_3$  da 50 ms a 200 ms ed infine escono dal museo,

Per una migliore gestione del museo la coda per entrare deve essere gestita in maniera da massimizzare le presenze. Quindi ad esempio se ne museo ci sono ancora 5 posti disponibili, ed arriva prima un gruppo di 7 persone e successivamente un gruppo di 4 persone, si deve consentire al gruppo arrivato dopo di entrare nel museo

Si analizzi la situazione descritta considerando i gruppi di visitatori come thread separati, e si sviluppi in linguaggio C un programma che simuli la situazione descritta commentando le scelte fatte e utilizzando opportuni elementi di sincronizzazione ,

### Esercizio 2(10/30)

Scrivere un programma in linguaggio C in ambiente linux che data una stringa di una operazione su interi come riga di comando ne calcoli il risultato utilizzando processi distinti per ogni operazione, ad esempio:

**5 - 7 + 6 \* 3 / 6    deve dare come risultato    2**

Il processo principale

- crea il primo processo figlio passando l'operatore - e i due parametri 5 e 7, e attendere la fine del primo figlio;
- crea il secondo figlio passando l'operatore + e i due parametri, il risultato precedente -2 e 6, e attendere la fine del secondo figlio;
- crea il terzo figlio passando l'operatore \* e i due parametri, il risultato precedente 4 e 3 , e attendere la fine del terzo figlio;
- crea il quarto figlio passando l'operatore / e i due parametri, il risultato precedente 12 e 6

Ogni figlio deve calcolare l'operazione data e inviarla al padre

(si utilizzi la funzione exit nei figli per restituire al padre il risultato, e la funzione WEXITSTATUS() nel padre per ricevere la soluzione ;