INFORMATICA INDUSTRIALE E SISTEMI OPERATIVI IDI
Prima prova scritta in itinere, 23 marzo 2004

Nome: _____ Cognome: _____

Esercizio 1

Siano dati 4 processi, caratterizzati dai tempi di arrivo e di utilizzo della CPU indicati nella tabella seguente:

pid	t_{arrivo}	t_{CPU}
1	0	5
2	0	4
3	1	1
4	2	2

Si illustrino graficamente le tracce di esecuzione per i 4 processi in esame, considerando gli algoritmi di scheduling *first come first served* (FCFS) e *shortest job first* (SJF) con prelazione. Si calcolino inoltre, nei due casi, i tempi di attesa e completamento medi. Infine, si indichi per quale dei due casi si verifica il maggior numero di commutazioni di contesto. Nel caso due processi risultino essere equivalenti rispetto ai criteri di scelta implicati dai suddetti algoritmi, si effettui la selezione sulla base dell'identificatore di processo, accordando la preferenza al processo con l'identificatore più basso.

Esercizio 2

Si definisca una classe Java che realizzi un thread in grado di calcolare la produttoria $P(j,k)=\prod_{(i=j)}^{(j+k)}(i)$. Utilizzando la suddetta classe si realizzi il calcolo del fattoriale di un numero intero M = N x Q ($M!=\prod_{(i=1)}^{(M)}(i)$) attraverso un programma Java che 1) distribuisce ad un insieme di N threads il calcolo delle produttorie parziali, 2) quindi si blocca in attesa del completamento di tutti i threads e 3) infine determina il valore del fattoriale moltiplicando i prodotti parziali.

Per gestire valori interi in precisione arbitraria, si utilizzi la classe *BigInteger* (inclusa nel package *java.math*). La classe dispone di un costruttore

> *public BigInteger(String v)*, per l'inizializzazione di un oggetto di tipo BigInteger data la rappresentazione sotto forma di stringa di un valore intero,

e di un metodo

> *public BigInteger multiply(BigInteger v)*, per eseguire il prodotto di un oggetto di tipo BigInteger con un altro oggetto *v* dello stesso tipo. Il metodo restituisce una nuova istanza della classe.