

CORSO DI LAUREA IN INFORMATICA
Sistemi Operativi e Reti
ESONERO 2 - 16.02.2018 - A.A. 2017/2018

Cognome:	Nome:	<input type="checkbox"/> Sistemi operativi e reti (12 CFU) <input type="checkbox"/> Sistemi operativi con lab. (6 CFU) <input type="checkbox"/> Reti di calcolatori (6 CFU)	Firma:
-----------------	--------------	---	---------------

Sistemi Operativi

1. Con riferimento ai processi P1, P2 e P3 aventi istanti di arrivo e durate di CPU burst specificate nella seguente tabella:

Processo	Istante di arrivo	Durata cpu burst
P1	0	60
P2	10	30
P3	15	15

- A) disegnate il diagramma temporale che mostra l'esecuzione dei processi in base all'algoritmo di scheduling della CPU SRTF (Shortest Remaining Time First);
- B) per tale algoritmo di scheduling calcolate il tempo medio di completamento (turnaround time) e il tempo medio di attesa dei tre processi. (6 punti)
2. In un sistema 4 processi, P1...P4, condividono 4 risorse, R1...R4, ciascuna di tipo diverso. In un dato istante la situazione è la seguente: P1 alloca R1 e richiede R2, P2 alloca R2 e richiede R3, P3 alloca R3 e R4 e P4 richiede R4 e R1. Determinate, utilizzando il grafo di allocazione delle risorse e motivando la risposta, se il sistema si trova in stallo e, in caso affermativo, quali sono i processi e le risorse coinvolti. Nel caso in cui in sistema non si trovasse in stallo, indicate una possibile sequenza temporale con la quale i processi terminano. (6 punti)
3. Un sistema operativo con memoria virtuale utilizza la paginazione con le seguenti caratteristiche: indirizzi virtuali e fisici a 32 bit, pagine di 4 KB, descrittori di pagina di 8 byte. Determinate:
- A) il numero di pagine di cui sono costituiti rispettivamente lo spazio di indirizzamento virtuale e quello fisico; B) il numero di bit di cui sono costituiti l'indice di pagina e l'offset; C) il numero di righe e la dimensione in byte che occupa la tabella delle pagine di un processo P con spazio virtuale di 4002 KB; D) se l'allocazione in memoria del processo P del punto C produce frammentazione di memoria. (8 punti)
4. Realizzate un programma multi-thread con modello *master-worker* in C, che svolga quanto segue:
Il thread main (master) crea 20 nuovi thread (worker) ad intervalli di tempo compresi tra 1 e 3 secondi e quindi si sospende. I worker creati possono essere di due tipi: *Tipo1* e *Tipo2*. La scelta del tipo è casuale. Un worker di *Tipo1* aggiunge, ogni secondo, ad una variabile TOT condivisa, un valore casuale compreso tra 1 e 100. Un worker di *Tipo2*, invece, sottrae, ogni secondo, al valore di TOT un valore casuale compreso tra 1 e 50. Se un worker estrae il numero 17 si blocca e può essere riattivato solo da un altro thread del suo stesso tipo. L'applicazione termina quando un thread estrae il valore 13. Prima di terminare il thread main stampa sullo schermo il valore di TOT. (10 punti)