

CORSO DI LAUREA IN INFORMATICA
Sistemi Operativi e Reti
Esonero 1 - 25.01.2019 - A.A. 2018/2019

Cognome:	Nome:	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. (3 punti)
2. In relazione al problema della sostituzione delle pagine, in un sistema con memoria paginata, in un determinato istante i primi 8 bit d'uso (U) e di modifica (M) della tabella delle pagine fisiche hanno i seguenti valori: P0:U=1,M=0; P1:U=0,M=1; P2:U=1,M=1; P3:U=1,M=0; P4:U=0,M=1; P5:U=0,M=0; P6:U=1,M=1; P7:U=0,M=1. Al prossimo page-fault il valore della prossima pagina da esaminare per il rimpiazzamento è quella con indice 1 (P1). Indicate quale sarà la pagina da sostituire e scrivete i nuovi valori che assumeranno i primi 8 elementi dei bit d'uso e di modifica della tabella delle pagine fisiche nel caso in cui il sistema utilizza per il rimpiazzamento delle pagine l'algoritmo *second chance* (algoritmo dell'orologio). (3 punti)
3. A) Relativamente all'organizzazione fisica del file system, descrivete sinteticamente la tecnica di allocazione ad indice. B) Considerate un semplice file system che adotti la tecnica di allocazione ad indice a un livello, nel quale la dimensione del blocco è di 4 KB e ogni blocco è indirizzato da 32 bit. Calcolate la dimensione massima di un file. (4 punti)
4. Realizzate un programma multi-thread in C, completo di commento, che svolga quanto segue:
Il thread main crea una matrice di numeri interi di dimensione NxM assegnando a ciascun elemento della matrice un valore casuale compreso tra 0 e 255. Dopo aver creato la matrice, il thread main crea 3 thread *worker*. Il primo thread ha il compito di eseguire la somma delle righe dispari della matrice, il secondo la somma delle righe pari. La somma di tutte le righe della matrice deve essere memorizzata in una variabile TOT che il terzo thread visualizzerà sullo schermo prima che l'applicazione termini. (5 punti)