

**ESAME DI SISTEMI OPERATIVI**  
Prova parziale del 21/11/2022 - Traccia 1

**Esercizio 1 (12 punti)**

Un sistema avente 2 processori (Proc1, Proc2) adotta una strategia di CPU scheduling a coda multilivello con feedback. Nel sistema sono allocate 2 code, ognuna dedicata allo scheduling di uno dei processori. Lo scheduling di Proc1 è Round-Robin con quanto pari a 4, quello di Proc2 è SJF con preemption. Il tempo di context-switching è pari a 1. Quando un processo nasce, viene associato a Proc1, dove esegue il suo primo burst, poi esegue operazioni di I/O su dispositivi ad esso dedicati, e quindi esegue il suo secondo burst su Proc2.

Mostrare le carte di Gantt e calcolare il tempo di attesa medio, al variare di  $x$ , quando in tale sistema arrivano tre processi con le seguenti caratteristiche:

	P1	P2	P3
Istante di arrivo	0	7	8
Primo CPU burst	9	5	2
I/O burst	4	4	3
Secondo CPU burst	8	$x$	4

**Esercizio 2 (10 punti)**

Lo studio di un docente universitario può ospitare, oltre al docente stesso, solamente uno studente alla volta. Ci sono  $M$  sedie in corridoio, davanti all'ufficio, dove gli studenti si possono sedere ed attendere il proprio turno se il docente è occupato con un altro studente. Quando non ci sono studenti, il docente si addormenta sulla sua scrivania. Se uno studente arriva e trova il docente che dorme, deve svegliarlo. Se invece il docente sta parlando con un altro studente, si siede su una delle sedie in corridoio. Se non ci sono sedie disponibili, va via.

Individuare le risorse condivise del sistema e poi descrivere i processi coinvolti garantendone, ove necessario, la mutua esclusione e la sincronizzazione mediante uso di semafori.

**Esercizio 3 (8 punti)**

- Come cambia il diagramma degli stati di un processo in un sistema dotato di Medium-Term Scheduler? E perché?
- Perché per lo scheduler della CPU non è conveniente considerare algoritmi molto complessi mentre per lo scheduler che immette nel sistema processi "New" può essere conveniente?
- Esiste un contesto nel quale una soluzione alla mutua esclusione mediante Busy Form of Waiting sia da preferire rispetto ad una soluzione con semafori?

**Esercizio facoltativo**

Spiegare perché l'algoritmo del fornaio (riportato qui di seguito) soddisfa uno dei tre requisiti di: mutua esclusione, progresso e attesa limitata.

*Repeat*

*choosing[i] := true ;*

*number[i] := max (number[0], ..., number [n - 1])+1;*

*choosing[i] := false;*

*for j := 0 to n - 1*

*do begin*

*while choosing[j] do no-op;*

*while number[j]  $\neq$  0 and*

*(number[j],j) < (number[i], i) do no-op;*

*end;*

*sezione critica*

*number[i] := 0;*

*sezione "normale"*

*until false;*



**ESAME DI SISTEMI OPERATIVI**  
**Prova parziale del 21/11/2022 - Traccia 2**

**Esercizio 1 (12 punti)**

Un sistema avente 2 processori (Proc1, Proc2) adotta una strategia di CPU scheduling a coda multilivello con feedback. Nel sistema sono allocate 2 code, ognuna dedicata allo scheduling di uno dei processori. Lo scheduling di Proc1 è Round-Robin con quanto pari a 3, quello di Proc2 è SJF con preemption. Il tempo di context-switching è pari a 1. Quando un processo nasce, viene associato a Proc1, dove esegue il suo primo burst, poi esegue operazioni di I/O su dispositivi ad esso dedicati, e quindi esegue il suo secondo burst su Proc2.

Mostrare le carte di Gantt e calcolare il tempo di attesa medio, al variare di  $x$ , quando in tale sistema arrivano tre processi con le seguenti caratteristiche:

	P1	P2	P3
Istante di arrivo	0	5	6
Primo CPU burst	7	4	1
I/O burst	3	3	2
Secondo CPU burst	8	$x$	4

**Esercizio 2 (10 punti)**

Un dottore visita un paziente alla volta. Ci sono  $M$  sedie nella sala d'attesa dove i pazienti si possono sedere ed attendere il proprio turno se il dottore è occupato con un altro paziente. Quando non ci sono pazienti, il dottore si addormenta sulla sua scrivania. Se un paziente arriva e trova il dottore che dorme, deve svegliarlo. Se invece il dottore sta visitando un altro paziente, si siede su una delle sedie nella sala d'attesa. Se non ci sono sedie disponibili, va via. Individuare le risorse condivise del sistema e poi descrivere i processi coinvolti garantendone, ove necessario, la mutua esclusione e la sincronizzazione mediante uso di semafori.

**Esercizio 3 (8 punti)**

- In un sistema multithreaded può esistere un Thread Control Block, analogo al Process Control Block? E perché?
- Dimostrare con un esempio perché il codice di una Signal su un semaforo deve essere eseguito in maniera atomica.
- Perché è impossibile stimare la lunghezza dei CPU burst di un processo solo esaminando il codice del processo stesso?

**Esercizio facoltativo**

Determinare quanti processi vengono creati dal codice riportato di seguito, mostrando come si è ottenuto il risultato.

```
void main()
{
    int i;

    for(i=0; i<4; i++)
        fork();
}
```