

Esercizio 1

Cosa si intende per protezioni a livello hardware o *architetture di protezione*? Motivare l'importanza di tali protezioni. Dare, infine, almeno un esempio di architettura di protezione illustrando come essa viene utilizzata dal Sistema Operativo.

Esercizio 2

Si consideri un sistema con 3 risorse R1,R2,R3 (una istanza per tipo di risorsa). Inizialmente, P1 richiede R1 e P2 richiede R2 e P3 richiede R1. Per terminare, il processo P1 necessita anche di R2 e R3, mentre il processo P2 necessita di R1 e P3 necessita di R2.

1. Discutere, utilizzando il grafo di assegnazione delle risorse, la possibilità di stallo nel sistema sopra descritto; nel caso si individui una situazione di stallo, evidenziare l'insieme dei processi effettivamente in stallo
2. Mostrare come lo stallo viene prevenuto se i processi sono modificati (riprogrammati) in modo da non effettuare mai "possesso e attesa" (hold-and-wait); Spiegare
3. Mostrare, infine, come lo stallo viene evitato tramite l'algoritmo con grafo di assegnazione delle risorse.

Esercizio 3

Quali sono le condizioni necessarie per lo stallo? Discutere se e in che modo tali condizioni possano essere negate al fine di prevenire o evitare lo stallo.

Esercizio 4

Si consideri uno spogliatoio condiviso dai seguenti processi donna e uomo:

```
process donna{                                process uomo{
    sp.entra_donna();                          sp.entra_uomo();
    <cambia l'abito>                            <cambia l'abito>
    sp.esci_donna();                            sp.esci_uomo();
}                                                }

monitor sp {
    int n_uomini=0,n_donne=0;
    condition attendi;

    entra_donna() {          esci_donna() {          entra_uomo() {          esci_uomo() {

}                                }                                }                                }

}
```

1. Nello spogliatoio non è consentita la presenza simultanea di uomini e donne. Completare il codice di tali funzioni (direttamente negli appositi spazi vuoti) spiegando, qui sotto, la soluzione proposta

2. Discutere la possibilità di attesa illimitata (*starvation*) e proporre possibili soluzioni a tale problema

Esercizio 5

Si consideri un processo che fa riferimento a 5 pagine virtuali nel seguente ordine:

1, 2, 3, 4, 2, 4, 5, 4, 5, 6, 5, 6, 5, 1

Si consideri una memoria fisica (inizialmente vuota) di 3 frame e si mostri il funzionamento degli algoritmi seguenti:

1. Ottimo,

	1	2	3	4	2	4	5	4	5	6	5	6	5	1
Frame 1														
Frame 2														
Frame 3														
Page Fault														

Spiegare brevemente:

2. LRU (Least Recently Used),

	1	2	3	4	2	4	5	4	5	6	5	6	5	1
Frame 1														
Frame 2														
Frame 3														
Page Fault														

Spiegare brevemente:

3. Seconda chance (clock).

	1	2	3	4	2	4	5	4	5	6	5	6	5	1
Frame 1														
Frame 2														
Frame 3														
Page Fault														

Spiegare brevemente:

Esercizio 6

Considerare tre processi $P1$, $P2$ e $P3$. I primi due modificano una variabile condivisa x . L'ultimo ha il compito di stampare il valore di x prima e dopo che le modifiche siano state effettuate. Aggiungere le necessarie sincronizzazioni al codice dei tre processi utilizzando i semafori. Spiegare il funzionamento della soluzione indicata (ricordarsi di indicare il valore di inizializzazione dei semafori).

```

P1 {
    x=x+10;
}
P2 {
    x=x+100;
}
P3 {
    print(x); //prima delle mod.
    print(x); //dopo le modifiche
} }
    
```

Spiegazione:

Esercizio 7

Considerare un sistema in cui sono pronti, nell'ordine, 3 processi $P1$, $P2$ e $P3$ che utilizzano la CPU per 10, 15, 20 unità di tempo prima di terminare. È inoltre presente un quarto processo I/O-bound $P4$ che, periodicamente, usa un'unità di tempo di CPU e va in attesa di I/O per 5 unità di tempo prima di chiedere nuovamente la CPU. $P4$ è l'ultimo processo nella coda dei pronti.

- Mostrare, tramite un opportuno schema, come vengono eseguiti tali processi per 20 unità di tempo, nel caso in cui l'algoritmo di scheduling sia:

1. Round Robin senza priorità (quanto = 1)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1																				
P2																				
P3																				
P4																				

Spiegazione:

2. Shortest Job First (SJF o SPN)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1																				
P2																				
P3																				
P4																				

Spiegazione:

3. Shortest Remaining Time First (SRTF)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1																				
P2																				
P3																				
P4																				

Spiegazione:

- Quale dei tre algoritmi garantisce maggiore interattività, quale di essi garantisce maggiore equità e quale, infine, garantirà il minor tempo medio di completamento (Turnaround medio)? Discutere.