

Università degli Studi di Trento - Esame Sistemi Operativi 1
23 Gennaio 2018

ISTRUZIONI: Scrivere in modo chiaro e leggibile. Scrivere il proprio nome, cognome e matricola su ogni foglio. Tempo a disposizione 120 minuti. Restituire il testo dell'esame. Totale 6 domande.

1. Spiegare i principi su cui si basa la tecnologia RAID. Spiegare in dettaglio come funziona, i vantaggi e gli svantaggi del RAID di livello 4. **(5 punti)**
2. Spiegare in dettaglio cosa è l'attesa attiva (busy waiting). Perché è un problema e come è possibile risolverlo. **(5 punti)**
3. **(5 punti)** Si consideri la situazione in cui vi siano in esecuzione quattro processi, P_A , P_B , P_C e P_D , con la matrice delle risorse ($R_1 R_2 R_3 R_4 R_5$) allocate e la matrice del numero massimo di risorse di cui possono disporre come segue:

$$\text{ALLOC} = \begin{bmatrix} 1 & 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad \text{MAX} = \begin{bmatrix} 1 & 1 & 2 & 1 & 3 \\ 2 & 2 & 2 & 1 & 1 \\ 2 & 1 & 3 & 1 & 0 \\ 1 & 1 & 2 & 2 & 1 \end{bmatrix} \begin{matrix} P_A \\ P_B \\ P_C \\ P_D \end{matrix}$$

Se il vettore delle risorse disponibili è $A=(0,0, x,1,1)$ qual è il minimo valore di x che rende lo stato attuale sicuro (mostrare il processo che ha portato a stabilire il valore di x)?

4. In un disco di 80 Gbyte diviso in cluster di 4 Kbyte quanti byte (o Kbyte o Mbyte) occupa la bitmap che rappresenta la posizione delle aree libere? Quanto occupa complessivamente un indice FAT32? **(6 punti)**

5. (6 punti)

Si consideri la seguente stringa di riferimenti a pagine:

4,8,3,11,5,7,2,8,3,1,2,8,11,8,4,5,6,7,11,9

Assumendo uno schema di demand paging con 4 frames, quanti page fault si verificano con i seguenti algoritmi di rimpiazzamento delle pagine?

- (a) LRU
- (b) L'algoritmo ottimale
- (c) FIFO

6. La situazione descritta di seguito con due processi A e B e due risorse può generare una situazione di stallo. Come modificare il codice per evitare che questo avvenga? **(6 punti)**

```
typedef int semaforo;  
semaforo risorsa_1;  
semaforo risorsa_2;
```

```
void processo_A(void) {  
    presa(&risorsa_1);  
    presa(&risorsa_2);  
    usa_entrambe_risorse();  
    rilascio(&risorsa_2);  
}
```

```
        rilascio(&risorsa_1);
    }

    void processo_B(void) {
        presa(&risorsa_2);
        presa(&risorsa_1);
        usa_entrambe_risorse();
        rilascio(&risorsa_1);
        rilascio(&risorsa_2);
    }
```