

**Esercizi**

1. Mostrare una possibile implementazione di un semaforo a valori interi senza ricorrere al *busy waiting* (o riducendolo al minimo indispensabile). [5 punti]
2. Si consideri il seguente algoritmo di scheduling basato su priorità variabili dinamicamente, nel quale valori alti di priorità implicano valori alta priorità (cioè 0 =minimo,  $\infty$  =max).

Quando un processo è in attesa della CPU (non in esecuzione), la sua priorità aumenta linearmente ad una velocità  $\alpha$ . Quando è in esecuzione, la sua priorità aumenta linearmente ad una velocità  $\beta$ . I parametri  $\alpha$  e  $\beta$  determinano il tipo di algoritmo.

Tutti i processi entrano inizialmente nel sistema con una priorità pari a 0.

Qual è l'algoritmo corrispondente alla seguente assegnazione:

- $\beta > \alpha > 0$
- $\alpha > \beta > 0$

Giustificare il risultato.

[7 punti]

3. Si descriva il meccanismo della segmentazione paginata, indicando chiaramente uno **schema a blocchi**, ed un esempio di traduzione degli indirizzi. [6 punti]
4. Si consideri un sistema che utilizzi la paginazione su domanda per la gestione della memoria. La tabella delle pagine è memorizzata nella memoria principale, avente un tempo di accesso di  $T_{mem} = 50ns$ . Il costo di gestione di un page fault è di  $8ms$  nel caso in cui la pagina rimpiazzata non è modificata, e di  $20ms$  nel caso in cui sia modificata. Assumendo che la pagina rimpiazzata risulti modificata il 70% delle volte, qual è il massimo tasso di page fault accettabile per avere un tempo di memoria effettivo non superiore a  $100ms$ ?

[4 punti]

5. Descrivere la struttura del file system (su disco) di UNIX.

[5 punti]

6. Descrivere la differenza tra user-level e kernel-level thread.

[5 punti]