

Domande Generali sui Sistemi Operativi

1. Qual è la funzione principale di un sistema operativo?
 - **A)** Eseguire direttamente il codice macchina
 - **B)** Gestire e coordinare l'hardware e il software del sistema
 - **C)** Tradurre il codice di alto livello in codice macchina
 - **D)** Aumentare la velocità della CPU
 - **Risposta corretta: B**

2. Quale dei seguenti *non* è un componente del sistema operativo?
 - **A)** Gestore della memoria
 - **B)** Gestore delle periferiche
 - **C)** File system
 - **D)** Compilatore
 - **Risposta corretta: D**

3. Qual è la funzione principale di una chiamata di sistema (*system call*)?
 - **A)** Permettere all'utente di accedere direttamente alla CPU
 - **B)** Permettere ai programmi utente di richiedere servizi al sistema operativo
 - **C)** Organizzare la struttura della memoria
 - **D)** Gestire la rete
 - **Risposta corretta: B**

4. In un sistema operativo, quale tra questi livelli di protezione hardware è il più privilegiato?
 - **A)** Livello utente
 - **B)** Livello kernel
 - **C)** Livello intermedio
 - **D)** Livello di applicazione
 - **Risposta corretta: B**

5. Il registro *program counter* (PC) contiene:
 - **A)** L'indirizzo in memoria della prima istruzione di ogni processo
 - **B)** L'indirizzo in memoria degli operandi di un'istruzione
 - **C)** L'indirizzo in memoria della prossima istruzione da eseguire
 - **D)** Il quanto temporale associato a ciascun processo
 - **Risposta corretta: C**

Esercizi su Scheduling della CPU

6. Supponiamo di avere i seguenti processi con tempi di arrivo e tempi di esecuzione. Qual è il tempo medio di attesa con algoritmo **First Come First Serve (FCFS)**?
- **Processi e tempi di esecuzione:**
 - P1: Arrivo = 0, Esecuzione = 6 ms
 - P2: Arrivo = 0, Esecuzione = 8 ms
 - P3: Arrivo = 0, Esecuzione = 7 ms
 - P4: Arrivo = 0, Esecuzione = 3 ms
 - **A) 8.75 ms**
 - **B) 10.5 ms**
 - **C) 9.25 ms**
 - **D) 10.25 ms**
 - **Risposta corretta: B**
7. Supponiamo di avere i seguenti processi con tempi di arrivo e tempi di esecuzione. Qual è il tempo medio di attesa con algoritmo **First Come First Serve (FCFS)**?
- **Processi e tempi di esecuzione:**
 - P1: Arrivo = 0 ms, Esecuzione = 6 ms
 - P2: Arrivo = 2 ms, Esecuzione = 8 ms
 - P3: Arrivo = 4 ms, Esecuzione = 7 ms
 - P4: Arrivo = 6 ms, Esecuzione = 3 ms
 - **A) 5.75 ms**
 - **B) 6.5 ms**
 - **C) 7.25 ms**
 - **D) 8 ms**
 - **Risposta corretta: C**
8. Con un **time quantum di 2 ms** in **Round Robin**, quale sarà l'ordine di esecuzione e il tempo medio di attesa per i seguenti processi?
- **Processi e tempi di esecuzione:**
 - P1: Arrivo = 0 ms, Esecuzione = 3 ms
 - P2: Arrivo = 0 ms, Esecuzione = 5 ms
 - P3: Arrivo = 0 ms, Esecuzione = 2 ms
 - P4: Arrivo = 0 ms, Esecuzione = 7 ms
 - **A) Tempo medio di attesa: 5.5 ms**
 - **B) Tempo medio di attesa: 7.25 ms**
 - **C) Tempo medio di attesa: 6.75 ms**
 - **D) Tempo medio di attesa: 8 ms**

- **Risposta corretta: B**

9. Con un **time quantum di 4 ms** in **Round Robin**, quale sarà l'ordine di esecuzione e il tempo medio di attesa per i seguenti processi?

- **Processi e tempi di esecuzione:**
 - P1: Arrivo = 0 ms, Esecuzione = 8 ms
 - P2: Arrivo = 1 ms, Esecuzione = 5 ms
 - P3: Arrivo = 2 ms, Esecuzione = 10 ms
 - P4: Arrivo = 3 ms, Esecuzione = 4 ms
- **A)** Tempo medio di attesa: 12.75 ms
- **B)** Tempo medio di attesa: 13.25 ms
- **C)** Tempo medio di attesa: 11.75 ms
- **D)** Tempo medio di attesa: 12.25 ms
- **Risposta corretta: A**

10. Con l'algoritmo **Shortest Job First (SJF)**, quale sarà l'ordine di esecuzione e il tempo medio di attesa?

- **Processi e tempi di esecuzione:**
 - P1: Arrivo = 0 ms, Esecuzione = 8 ms
 - P2: Arrivo = 0 ms, Esecuzione = 4 ms
 - P3: Arrivo = 0 ms, Esecuzione = 2 ms
 - P4: Arrivo = 0 ms, Esecuzione = 6 ms
- **A)** Tempo medio di attesa: 5.5 ms
- **B)** Tempo medio di attesa: 5 ms
- **C)** Tempo medio di attesa: 4.5 ms
- **D)** Tempo medio di attesa: 6 ms
- **Risposta corretta: B**

11. Con l'algoritmo **Shortest Job First (SJF)**, quale sarà l'ordine di esecuzione e il tempo medio di attesa?

- **Processi e tempi di esecuzione:**
 - P1: Arrivo = 0 ms, Esecuzione = 7 ms
 - P2: Arrivo = 1 ms, Esecuzione = 4 ms
 - P3: Arrivo = 2 ms, Esecuzione = 1 ms
 - P4: Arrivo = 3 ms, Esecuzione = 4 ms
- **A)** Tempo medio di attesa: 5.5 ms
- **B)** Tempo medio di attesa: 5 ms
- **C)** Tempo medio di attesa: 4.5 ms
- **D)** Tempo medio di attesa: 5.25 ms

- **Risposta corretta: D**

12. Con un algoritmo **Shortest Remaining Time First (SRTF)**, qual è il tempo medio di attesa?

- **Processi e tempi di esecuzione:**
 - P1: Arrivo = 0 ms, Esecuzione = 6 ms
 - P2: Arrivo = 2 ms, Esecuzione = 4 ms
 - P3: Arrivo = 4 ms, Esecuzione = 3 ms
 - P4: Arrivo = 6 ms, Esecuzione = 5 ms
- **A) 3.5 ms**
- **B) 4 ms**
- **C) 2.7 ms**
- **D) 4.6 ms**
- **Risposta corretta: B**

13. Con un algoritmo **Shortest Remaining Time First (SRTF)**, qual è il tempo medio di attesa?

- **Processi e tempi di esecuzione:**
 - P1: Arrivo = 0 ms, Esecuzione = 8 ms
 - P2: Arrivo = 1 ms, Esecuzione = 4 ms
 - P3: Arrivo = 2 ms, Esecuzione = 9 ms
 - P4: Arrivo = 3 ms, Esecuzione = 5 ms
- **A) 6.5 ms**
- **B) 4.25 ms**
- **C) 5.8 ms**
- **D) 6.25 ms**
- **Risposta corretta: A**

14. Supponiamo che i processi alternino CPU e I/O in un sistema **Round Robin** con un time quantum di 3 ms. Quale sarà il tempo medio di attesa per i processi?

- **Processi e tempi di esecuzione CPU/I/O:**
 - P1 = 5 ms (CPU), 2 ms (I/O), 4 ms (CPU)
 - P2 = 7 ms (CPU), 3 ms (I/O)
 - P3 = 4 ms (CPU), 5 ms (I/O), 3 ms (CPU)
- **A) 9.67 ms**
- **B) 10.5 ms**
- **C) 10.67 ms**
- **D) 11 ms**
- **Risposta corretta: C**

Creazione di Processi e Thread

15. Cosa restituisce la funzione **fork()** in un programma C se il processo figlio è stato creato con successo?

- **A)** 0 al figlio, PID del figlio al genitore
- **B)** PID del figlio sia al genitore che al figlio
- **C)** 0 al genitore, PID del genitore al figlio
- **D)** 0 al figlio, 0 al genitore
- **Risposta corretta: A**

16. Cosa significa la terminazione di un thread in un processo multithreaded?

- **A)** Il processo padre termina automaticamente
- **B)** Tutti i thread del processo terminano
- **C)** Il processo continua a funzionare con i thread rimanenti
- **D)** Il sistema operativo riavvia il thread
- **Risposta corretta: C**

17. Quale chiamata di sistema permette al processo padre di attendere il completamento del processo figlio?

- **A)** exec()
- **B)** wait()
- **C)** join()
- **D)** sleep()
- **Risposta corretta: B**

18. Quando un thread accede a una variabile condivisa, qual è il rischio senza opportune misure di sincronizzazione?

- **A)** Deadlock
- **B)** Race condition
- **C)** Starvation
- **D)** Mutual exclusion
- **Risposta corretta: B**

19. Perché l'uso di thread può migliorare le prestazioni in un'applicazione?

- **A)** I thread richiedono più memoria del processo
- **B)** I thread consentono l'esecuzione parallela all'interno di un processo

- **C)** I thread riducono l'uso della CPU
 - **D)** I thread sostituiscono il processo
 - **Risposta corretta: B**
-

Sincronizzazione (Lock, Semafori, Monitor)

20. Quale delle seguenti affermazioni è vera riguardo ai **semafori binari**?

- **A)** Possono assumere solo valori 0 o 1
- **B)** Sono più efficienti dei lock
- **C)** Non sono utilizzabili per il controllo di accesso
- **D)** Non richiedono sincronizzazione
- **Risposta corretta: A**

21. Qual è il principale vantaggio dell'uso dei **monitor** per la sincronizzazione?

- **A)** Consentono accesso non esclusivo alle risorse
- **B)** Evitano problemi di race condition in maniera più semplice
- **C)** Sono più veloci dei semafori
- **D)** Non richiedono blocchi di accesso
- **Risposta corretta: B**

22. Quale dei seguenti meccanismi di sincronizzazione utilizza un contatore per gestire l'accesso a una risorsa multipla?

- **A)** Monitor
- **B)** Semaforo contatore
- **C)** Mutex
- **D)** Lock binario
- **Risposta corretta: B**

23. Cos'è un **deadlock** in termini di sincronizzazione dei processi?

- **A)** Un processo in attesa di terminazione
- **B)** Un'area di memoria occupata da più processi
- **C)** Una situazione in cui nessun processo può avanzare perché bloccato
- **D)** Un accesso simultaneo a una variabile condivisa
- **Risposta corretta: C**

24. Qual è la differenza principale tra **mutex** e **semaforo**?

- **A)** I mutex sono contatori, i semafori no
- **B)** I semafori proteggono solo variabili globali
- **C)** I mutex consentono solo un singolo processo alla volta, i semafori più di uno
- **D)** I semafori non possono causare deadlock
- **Risposta corretta: C**