

Domande Generali sui Sistemi Operativi

1. *Quale delle seguenti affermazioni descrive meglio il concetto di "kernel mode"?*

- A) Una modalità in cui i processi utente accedono direttamente all'hardware
- B) Una modalità in cui il sistema operativo esegue codice con privilegi elevati
- C) Una modalità usata solo dai programmi applicativi
- D) Una modalità riservata ai driver di periferica esterni

2. *Quale delle seguenti operazioni è gestita direttamente dal sistema operativo?*

- A) Compilazione del codice sorgente
- B) Accesso ai dispositivi di I/O
- C) Interpretazione di linguaggi di scripting
- D) Debug del codice utente

3. *Qual è lo scopo principale del file system?*

- A) Fornire un'interfaccia per l'accesso ai file e alla memoria secondaria
- B) Gestire la comunicazione di rete tra processi
- C) Tradurre le chiamate di sistema in codice macchina
- D) Eseguire i processi in background

4. *Durante un context switch, quale delle seguenti informazioni deve essere sempre salvata?*

- A) Il contenuto della cache del disco
- B) La tabella delle pagine di tutti i processi
- C) Il contenuto della memoria virtuale condivisa
- D) Il contenuto dei registri della CPU del processo in esecuzione

5. *Cosa rappresenta la system call `exec()` in Unix/Linux?*

- A) Crea un nuovo processo figlio
- B) Termina il processo chiamante
- C) Sostituisce il codice del processo corrente con un nuovo programma
- D) Sospende il processo corrente fino alla terminazione di un figlio

Esercizi su Scheduling della CPU

6. *Calcolare il tempo medio di attesa con l'algoritmo First Come First Serve (FCFS) del seguente sistema:*

Processi e tempi di esecuzione:

- P1: Arrivo = 0, Esecuzione = 5 ms
- P2: Arrivo = 0, Esecuzione = 3 ms
- P3: Arrivo = 0, Esecuzione = 8 ms
- P4: Arrivo = 0, Esecuzione = 6 ms

- A) 5.5 ms
- B) 6.25 ms
- C) 7.25 ms
- D) 7.5 ms

7. Calcolare il tempo medio di attesa con l'algoritmo First Come First Serve (FCFS) del seguente sistema:

Processi e tempi di esecuzione:

P1: Arrivo = 0, Esecuzione = 4 ms
P2: Arrivo = 1, Esecuzione = 6 ms
P3: Arrivo = 2, Esecuzione = 3 ms
P4: Arrivo = 4, Esecuzione = 5 ms

- A) 4.5 ms
- B) 5 ms
- C) 5.25 ms
- D) 6.75 ms

8. Calcolare il tempo medio di attesa con l'algoritmo Round Robin (RR) del seguente sistema:

Processi e tempi di esecuzione (q = 2 ms):

P1: Arrivo = 0, Esecuzione = 5 ms
P2: Arrivo = 0, Esecuzione = 3 ms
P3: Arrivo = 0, Esecuzione = 6 ms
P4: Arrivo = 0, Esecuzione = 4 ms

- A) 9 ms
- B) 9.25 ms
- C) 10 ms
- D) 10.5 ms

9. Calcolare il tempo medio di attesa con l'algoritmo Round Robin (RR) del seguente sistema:

Processi e tempi di esecuzione (q = 3 ms):

P1: Arrivo = 0, Esecuzione = 9
P2: Arrivo = 1, Esecuzione = 5
P3: Arrivo = 7, Esecuzione = 6
P4: Arrivo = 11, Esecuzione = 4

- A) 7.5 ms
- B) 8 ms
- C) 8.25 ms
- D) 9 ms

10. Calcolare il tempo medio di attesa con l'algoritmo Shortest Job First (SJF - non preemptive):

Processi e tempi di esecuzione:

P1: Arrivo = 0, CPU = 8

P2: Arrivo = 0, CPU = 2

P3: Arrivo = 0, CPU = 6

P4: Arrivo = 0, CPU = 4

A) 4.5 ms

B) 4.75 ms

C) 5 ms

D) 5.5 ms

11. Calcolare il tempo medio di attesa con l'algoritmo Shortest Remaining Time First (SRTF - preemptive):

Processi e tempi di esecuzione:

P1: Arrivo = 0, CPU = 8

P2: Arrivo = 2, CPU = 5

P3: Arrivo = 4, CPU = 2

A) 3 ms

B) 3.33 ms

C) 3.67 ms

D) 4.16 ms

12. Calcolare il tempo medio di attesa con l'algoritmo Round Robin (RR) del seguente sistema:

Processi, tempi di esecuzione e di I/O (arrivo = 0 per tutti; q = 4 ms):

P1: CPU 6, I/O 4, CPU 4

P2: CPU 5, I/O 2, CPU 3

P3: CPU 4, I/O 3, CPU 2

A) 8.33 ms

B) 8.67 ms

C) 9.25 ms

D) 9.33 ms

Creazione di Processi e Thread

13. In un sistema con 10 processi, ciascuno di essi esegue una `fork()`. Supponendo che tutte le chiamate vadano a buon fine, quanti processi saranno presenti in totale nel sistema?

A) 10

B) 20

C) 100

D) Dipende dal sistema operativo

14. Cosa accade se un processo chiama `wait()` senza figli attivi?

- A) Rimane in attesa indefinita
- B) Termina automaticamente
- C) Il sistema operativo genera un errore immediato
- D) Si blocca finché un altro processo lo risveglia

15. In un processo multithreaded, cosa condividono tutti i thread?

- A) Stack
- B) Registro program counter
- C) Spazio di indirizzamento (memoria e variabili globali)
- D) Identificatore di thread

16. Qual è lo scopo della chiamata `pthread_join()`?

- A) Creare un nuovo thread
- B) Terminare un thread
- C) Forzare la sincronizzazione dei thread
- D) Attendere la terminazione di un thread specifico

17. Quale situazione può verificarsi se due thread accedono contemporaneamente a una variabile globale senza sincronizzazione?

- A) Starvation
- B) Deadlock
- C) Race condition
- D) Context switch

Sincronizzazione (Lock, Semafori, Monitor)

18. Quale proprietà è essenziale per garantire la mutual exclusion?

- A) Tutti i processi devono poter entrare contemporaneamente nella sezione critica
- B) Al massimo un processo per volta può trovarsi nella sezione critica
- C) Ogni processo deve entrare nella sezione critica entro un tempo finito
- D) Un processo non deve mai poter accedere alla sezione critica

19. Un semaforo con valore iniziale 3 può essere usato per:

- A) Consentire a tre thread per volta di accedere contemporaneamente a una risorsa condivisa
- B) Consentire a un solo thread per volta di accedere a una risorsa condivisa
- C) Bloccare indefinitamente tutti i thread in attesa
- D) Implementare una comunicazione one-to-one tra due thread

20. *Quale delle seguenti tecniche di sincronizzazione evita il busy waiting?*

- A) Spinlock
- B) Test-and-set lock
- C) Monitor
- D) Preemption