

Iniziato	venerdì, 5 settembre 2025, 14:11
Stato	Completato
Terminato	venerdì, 5 settembre 2025, 14:50
Tempo impiegato	38 min. 57 secondi
Valutazione	8,0 su un massimo di 20,0 (40%)

Domanda 1

Risposta corretta

Punteggio ottenuto 2,0 su 2,0

Un hard disk ha la dimensione di 512 Gigabyte, è formattato in blocchi da 0x400 byte e adotta una qualche forma di allocazione indicizzata dello spazio su disco. Sull'hard disk è memorizzato un file A della dimensione di 600 Kbyte. Quante operazioni di I/O sono necessarie per leggere l'ultimo blocco di dati del file, assumendo già in RAM tutti gli attributi del file? (selezionate l'opzione di risposta che riporta il ragionamento numerico corretto)

Scegli un'alternativa:

- a. $2^{39}/2^{11} = 2^{28}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 512 Kbyte di dati. Assumendo una allocazione indicizzata a più livelli, ci vogliono 3 livelli di indirezione per registrare l'ultimo blocco di dati del file, e dunque, se è già in RAM il numero del blocco indice più esterno, saranno necessarie in tutto 4 operazioni di lettura su disco
- b. $2^{39}/2^{10} = 2^{29}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 256 Kbyte di dati. Assumendo una allocazione indicizzata a più livelli, ci vogliono due livelli di indirezione per registrare l'ultimo blocco di dati del file, e dunque, se è già in RAM il numero del primo blocco indice, saranno necessarie in tutto 3 operazioni di lettura su disco
- c. $2^{39}/2^{10} = 2^{29}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 128 Kbyte di dati. Assumendo una allocazione indicizzata a schema concatenato, ci vogliono 5 blocchi indice per registrare l'ultimo blocco di dati del file, e dunque, se è già in RAM il numero del blocco indice più esterno, saranno necessarie in tutto 6 operazioni di lettura su disco
- d. $2^{39}/2^{11} = 2^{28}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 512 Kbyte di dati. Assumendo una allocazione indicizzata a schema concatenato, ci vogliono 2 blocchi indice per registrare l'ultimo blocco di dati del file, e dunque, se è già in RAM il numero del primo blocco indice, saranno necessarie in tutto 3 operazioni di lettura su disco

Risposta corretta.

La risposta corretta è: $2^{39}/2^{10} = 2^{29}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 256 Kbyte di dati. Assumendo una allocazione indicizzata a più livelli, ci vogliono due livelli di indirezione per registrare l'ultimo blocco di dati del file, e dunque, se è già in RAM il numero del primo blocco indice, saranno necessarie in tutto 3 operazioni di lettura su disco

Domanda 2

Risposta errata

Punteggio ottenuto 0,0 su 3,0

In un sistema paginato è noto che lo spreco di memoria primaria dovuto alla frammentazione interna è in media di circa 2 Kbyte per processo, e un indirizzo logico è scritto su 30 bit. Se la tabella delle pagine più grande di questo sistema è grande 512 Kilobyte, quanto può essere grande al massimo lo spazio di indirizzamento fisico del sistema?

Scegli un'alternativa:

- a. 512 megabyte X
- b. 128 megabyte
- c. 256 megabyte
- d. 1 Gigabyte

Risposta errata.

La risposta corretta è: 256 megabyte

Domanda 3

Risposta errata

Punteggio ottenuto 0,0 su 2,0

Nel codice di wait e signal un solo processo per volta può aggiornare la variabile semaforica e la coda di wait del semaforo. Come può essere ottenuta questa condizione?

Scegli un'alternativa:

- a. wait e signal sono a loro volta sezioni critiche, e devono essere implementate con meccanismi di busy waiting o con la disabilitazione degli interrupt
- b. wait e signal sono a loro volta sezioni critiche, il che garantisce automaticamente che un solo processo alla volta possa accedere alla struttura dati che implementa il semaforo
- c. nei sistemi single-core un solo processo alla volta è in esecuzione, e quindi la condizione richiesta è automaticamente garantita. Nei sistemi multi-core il problema non esiste perché ogni core usa una copia diversa della variabile semaforica
- d. sono le system call sleep() e wakeup(P) usate per implementare wait e signal a garantire l'accesso mutuamente esclusivo alla struttura dati che implementa il semaforo X

Risposta errata.

La risposta corretta è: wait e signal sono a loro volta sezioni critiche, e devono essere implementate con meccanismi di busy waiting o con la disabilitazione degli interrupt

Domanda 4

Risposta corretta

Punteggio ottenuto 2,0 su 2,0

In un sistema operativo che adotta uno scheduling senza diritto di prelazione, quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella

sottostante

Processo	T. di arrivo	Burst
Pa	0	4
Pb	2	3
Pc	3	2
Pd	5	1

se si usa l'algoritmo di scheduling non preemptive che fornisce le migliori prestazioni possibili per schedulare i 4 processi in tabella:

il waiting time medio è: ✓

il turnaround medio è: ✓

il diagramma di GANTT è: ✓

l'algoritmo usato per le risposte precedenti potrebbe non portare mai a termine un processo?

✓

Risposta corretta.

La risposta corretta è:

In un sistema operativo che adotta uno scheduling senza diritto di prelazione, quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella

sottostante

Processo	T. di arrivo	Burst
Pa	0	4
Pb	2	3

Pc	3	2
Pd	5	1

se si usa l'algoritmo di scheduling non preemptive che fornisce le migliori prestazioni possibili per schedulare i 4 processi in tabella:

il waiting time medio è: [7/4]

il turnaround medio è: [17/4]

il diagramma di GANTT è: [(0) ... Pa ... (4) ... Pc ... (6) ... Pd ... (7) ... Pb ... (10)]

l'algoritmo usato per le risposte precedenti potrebbe non portare mai a termine un processo? [si]

Domanda 5

Risposta errata

Punteggio ottenuto 0,0 su 2,0

Quali sono gli hard link permessi ad una directory D in Unix?

Scegli un'alternativa:

a.

- 1) gli hard link ".." delle sottocartelle di D
- 2) l'hard link "." dentro D
- 3) "D" dentro la cartella che contiene D stessa

b.

- 1) gli hard link ".." delle sottocartelle di D
- 2) l'hard link "." dentro D
- 3) "D" dentro la cartella che contiene D stessa

c.

- 1) gli hard link ".." delle sottocartelle di D
- 2) l'hard link ".." dentro D
- 3) "D" dentro la cartella che contiene D stessa

d. ✖

- 1) gli hard link ".." delle sottocartelle di D
- 2) l'hard link ".." dentro D
- 3) "D" dentro la cartella che contiene D stessa

Risposta errata.

La risposta corretta è:

- 1) gli hard link ".." delle sottocartelle di D
- 2) l'hard link "." dentro D
- 3) "D" dentro la cartella che contiene D stessa

Domanda 6

Risposta corretta

Punteggio ottenuto 2,0 su 2,0

Di un sistema è noto che la tabella delle pagine più grande del sistema occupa esattamente un frame, il numero di un frame è scritto su 4 byte usando solo i primi 26 bit, la frammentazione interna media prodotta da un processo del sistema è di circa 4 Kilobyte.

quanto sono grandi lo spazio di indirizzamento fisico e logico del sistema?

(selezionate l'opzione di risposta che riporta il ragionamento aritmetico e il risultato corretti)

Scegli un'alternativa:

- a. La tabella delle pagine più grande del sistema ha $2^{12}/4$ entry,e dunque lo spazio logico del sistema è grande 1 Megabyte.

Lo spazio fisico è suddiviso in 2^{26} frame, e dunque ha una dimensione di 256 Megabyte

- b. La tabella delle pagine più grande del sistema ha $2^{13}/4$ entry,e dunque lo spazio logico del sistema è grande 64 Megabyte.

Lo spazio fisico è suddiviso in 2^{26} frame, e dunque ha una dimensione di 512 Gigabyte

- c. La tabella delle pagine più grande del sistema ha $2^{13}/4$ entry,e dunque lo spazio logico del sistema è grande 16 ✓ Megabyte.

Lo spazio fisico è suddiviso in 2^{26} frame, e dunque ha una dimensione di 512 Gigabyte

- d. La tabella delle pagine più grande del sistema ha $2^{13}/4$ entry,e dunque lo spazio logico del sistema è grande 16 Megabyte.

Lo spazio fisico è suddiviso in 2^{32} frame, e dunque ha una dimensione di 4 Gigabyte

Risposta corretta.

La risposta corretta è: La tabella delle pagine più grande del sistema ha $2^{13}/4$ entry,e dunque lo spazio logico del sistema è grande 16 Megabyte.

Lo spazio fisico è suddiviso in 2^{26} frame, e dunque ha una dimensione di 512 Gigabyte

Domanda 7

Risposta errata

Punteggio ottenuto 0,0 su 3,0

Si consideri questa variante del problema dei produttori e consumatori:

Semaphore full =0; empty =N;

codice consumatore:

```
repeat
  wait(full);
  <preleva dato dal buffer>
  signal(empty);
  <consuma dato>
forever
```

codice produttore:

```
repeat
  <producি dato>
  wait(empty)
  <inserisci dato nel buffer>
  signal(full)
  forever
```

Questa soluzione funziona? Se sì, a quali condizioni? Se no, perché?

Scegli un'alternativa:

- a. La soluzione proposta non funziona perchè se ci sono almeno due posizioni libere nel buffer, ossia se empty ≥ 2 , ci potrebbero essere almeno 2 consumatori che superano la wait su empty e tentano di operare non in mutua esclusione sul buffer condiviso. ✗
- b. La soluzione proposta funziona a condizione che siano presenti un solo produttore e un solo consumatore. Manca infatti la mutua esclusione rispetto all'operazione di inserimento e prelievo di un elemento nel buffer condiviso.
- c. La soluzione proposta non funziona perchè se ci sono almeno due dati nel buffer, ossia se full ≥ 2 , ci potrebbero essere almeno 2 produttori che superano la wait su full e tentano di operare non in mutua esclusione sul buffer condiviso.
- d. La soluzione proposta funziona a condizione che produttori e consumatori accedano al buffer condiviso i momenti diversi. Manca infatti la mutua esclusione rispetto all'operazione di inserimento e prelievo di un elemento nel buffer condiviso.

Risposta errata.

La risposta corretta è: La soluzione proposta funziona a condizione che siano presenti un solo produttore e un solo consumatore. Manca infatti la mutua esclusione rispetto all'operazione di inserimento e prelievo di un elemento nel buffer condiviso.

Domanda 8

Risposta errata

Punteggio ottenuto 0,0 su 2,0

Dopo l'esecuzione dei seguenti comandi in un ambiente Unix (come visti a lezione):

- 1: cd /tmp
- 2: mkdir newfolder
- 3: echo "ciao" > pippo // crea un nuovo file di nome *pippo* contenente la stringa *ciao*
- 4: cd newfolder
- 5: ln/pippo paperino
- 6: ln -s /tmp/newfolder folder2
- 7: cp paperino topolino
- 8: echo "salve" >> topolino // aggiunge "salve" a fondo file
- 9: rm pippo
- 10: cat paperino // *cat* stampa il contenuto del file passato come argomento
- 11: mkdir folder3

Scegli un'alternativa:

a. 1. il link-counter dell'i-node di *paperino* è: 1

2. il link counter di *newfolder* è: 3

3. l'output del comando 10 è: "ciao"

4. il link counter di tmp è: aumentato di 2

b. 1. il link-counter dell'i-node di *paperino* è: 2

2. il link counter di *newfolder* è: 3

3. l'output del comando 10 è: no such file or directory

4. il link counter di tmp è: aumentato di 1

c. 1. il link-counter dell'i-node di *paperino* è: 2

2. il link counter di *newfolder* è: 3

3. l'output del comando 10 è: "ciao"

4. il link counter di tmp è: aumentato di 1

d. 1. il link-counter dell'i-node di *paperino* è: 2 ✗

2. il link counter di *newfolder* è: 2

3. l'output del comando 10 è: "ciao"

4. il link counter di tmp è: aumentato di 1

Risposta errata.

La risposta corretta è:

1. il link-counter dell'i-node di *paperino* è: 2

2. il link counter di *newfolder* è: 3

3. l'output del comando 10 è: "ciao"

4. il link counter di tmp è: aumentato di 1

Domanda 9

Risposta corretta

Punteggio ottenuto 2,0 su 2,0

I **vantaggi** delle librerie dinamiche sono:

Scegli un'alternativa:

- a. 1. Possono essere condivise tra più processi, per cui limitano lo spazio occupato in RAM rispetto a quelle statiche
2. Vengono caricate in RAM solo se sono chiamate, è sufficiente copiarle preventivamente in un segmento di shared memory
3. Se vengono aggiornate non occorre ricompilare i programmi che le usano
- b. 1. Possono essere condivise tra più processi, per cui limitano lo spazio occupato in RAM rispetto a quelle statiche
2. Vengono caricate in RAM solo se sono chiamate, per cui i processi partono più velocemente e non occupano inutilmente spazio in RAM
3. Se vengono aggiornate rendono più efficienti i programmi che le usano
- c. 1. Possono essere condivise tra più processi, per cui limitano lo spazio occupato in RAM rispetto a quelle statiche
2. Vengono caricate in RAM solo se sono chiamate, per cui i processi partono più velocemente e non occupano inutilmente spazio in RAM
3. Se vengono aggiornate non occorre ricompilare i programmi che le usano ✓
- d. 1. Possono essere condivise tra più processi, per cui non hanno bisogno di usare un segmento esplicito di shared memory, ma solo l'area di swap
2. Vengono caricate in RAM solo se sono chiamate, per cui i processi partono più velocemente e non occupano inutilmente spazio in RAM
3. Se vengono aggiornate non occorre ricompilare i programmi che le usano

Risposta corretta.

La risposta corretta è:

- 1. Possono essere condivise tra più processi, per cui limitano lo spazio occupato in RAM rispetto a quelle statiche
- 2. Vengono caricate in RAM solo se sono chiamate, per cui i processi partono più velocemente e non occupano inutilmente spazio in RAM
- 3. Se vengono aggiornate non occorre ricompilare i programmi che le usano