Iniziato	martedì, 17 giugno 2025, 14:12
Stato	Completato
Terminato	martedì, 17 giugno 2025, 14:49
Tempo impiegato	37 min. 10 secondi
Valutazione	**20,00** su un massimo di 23,00 (**86,96**%)

Domanda **1**

Risposta corretta

Punteggio ottenuto 2,00 su 2,00

I problemi che si possono presentare se non viene correttamente risolto il problema della sezione critica sono:

Scegli un'alternativa:

a.
1. due o più processi possono trovarsi contemporaneamente in sezione critica
2. quando un processo lascia la sua sezione critica, nessun altro processo riesce ad entrarvi
3. un processo spreca il suo quanto di CPU cercando di entrare in una sezione critica occupata

b.
1. due o più processi possono trovarsi contemporaneamente in sezione critica
2. un processo non esce più dalla sua sezione critica
3. un processo non riesce ad entrare nella sua sezione critica in tempo finito

c. ✔
1. due o più processi possono trovarsi contemporaneamente in sezione critica
2. quando un processo lascia la sua sezione critica, nessun altro processo riesce ad entrarvi
3. un processo non riesce ad entrare nella sua sezione critica in tempo finito

d.
1. in un certo momento della computazione la sezione critica è vuota
2. quando un processo lascia la sua sezione critica, nessun altro processo riesce ad entrarvi
3. un processo non riesce ad entrare nella sua sezione critica in tempo finito

Risposta corretta.

La risposta corretta è:
1. due o più processi possono trovarsi contemporaneamente in sezione critica
2. quando un processo lascia la sua sezione critica, nessun altro processo riesce ad entrarvi
3. un processo non riesce ad entrare nella sua sezione critica in tempo finito

Domanda **2**

Risposta corretta

Punteggio ottenuto 2,00 su 2,00

Assumendo che siano presenti in un sistema almeno due processi utente, quattro possibili ragioni per cui, in un moderno sistema operativo time sharing che implementa la memoria virtuale si può verificare un context switch tra due processi utente sono:

Scegli un'alternativa:

a.
1. scade il quanto di tempo del processo running
2. entra in coda di ready un processo con priorità maggiore di quello in stato running
3. il processo running inizia una operazione di I/O
4. il processo running esegue una wait e si addormenta sul semaforo

b.
1. scade il quanto di tempo del processo running
2. entra in coda di ready un processo con priorità maggiore di quello in stato running
3. il processo running ha appena completato una operazione di I/O
4. il processo running esegue una wait e si addormenta sul semaforo

c.
1. scade il quanto di tempo del processo running
2. entra in coda di ready un processo con priorità maggiore di quello in stato running
3. il processo running inizia una operazione di I/O
4. il processo running esegue una signal e si addormenta sul semaforo

d.
1. scade il quanto di tempo del processo running
2. entra in coda di ready un processo con priorità inferiore di quello in stato running
3. il processo running inizia una operazione di I/O
4. il processo running esegue una wait e si addormenta sul semaforo

Risposta corretta.

La risposta corretta è:
1. scade il quanto di tempo del processo running
2. entra in coda di ready un processo con priorità maggiore di quello in stato running
3. il processo running inizia una operazione di I/O
4. il processo running esegue una wait e si addormenta sul semaforo

Domanda **3**

Risposta corretta

Punteggio ottenuto 2,00 su 2,00

Tra le ragioni per cui un sistema operativo può finire in una condizione di thrashing troviamo:

Scegli un'alternativa:

a. un elevato numero di processi CPU bound rispetto alla dimensione della memoria principale, e l'adozione di una politica di allocazione globale o locale dei frame

b. un elevato numero di processi I/O bound rispetto alla dimensione della memoria principale, e l'adozione di una politica di allocazione globale o locale dei frame

c. un elevato numero di processi attivi rispetto alla dimensione della memoria principale, e l'adozione di una politica di allocazione globale o locale dei frame ✔

d. un elevato numero di processi attivi rispetto alla dimensione della memoria principale, e l'adozione di un algoritmo di sostituzione delle pagine di tipo FIFO

Risposta corretta.

La risposta corretta è: un elevato numero di processi attivi rispetto alla dimensione della memoria principale, e l'adozione di una politica di allocazione globale o locale dei frame

Domanda **4**

Risposta corretta

Punteggio ottenuto 2,00 su 2,00

Di un sistema è noto che la tabella delle pagine più grande del sistema occupa esattamente 2 frame, il numero di un frame è scritto su 4 byte usando solo i primi 26 bit, e nel sistema sono presenti in media 4 processi che insieme producono una frammentazione interna complessiva media di 8 Kilobyte.

lo spazio logico del sistema è grande: 8 Megabyte ✔

lo spazio fisico del sistema è grande: 256 Gigabyte ✔

512 Gigabyte	non si può ricavare dai dati del problema
	16 Megabyte
	1 Megabyte
4 Gigabyte	nessuno dei valori proposti

Risposta corretta.

La risposta corretta è:

Di un sistema è noto che la tabella delle pagine più grande del sistema occupa esattamente 2 frame, il numero di un frame è scritto su 4 byte usando solo i primi 26 bit, e nel sistema sono presenti in media 4 processi che insieme producono una frammentazione interna complessiva media di 8 Kilobyte.

lo spazio logico del sistema è grande: [8 Megabyte]
lo spazio fisico del sistema è grande: [256 Gigabyte]

Domanda **5**

Risposta corretta

Punteggio ottenuto 3,00 su 3,00

si consideri l'esecuzione della seguente porzione di codice che utilizza la system call fork:

```
int a, b, c, d, n, pid1, pid2, pid3;
a = 50, b = 60, c = 70, d = 80;
n = fork();
      if ( n == 0)
         {a = 55; b = 65;
          pid1 = getppid();
          printf("%d", pid1);
          exit(0);}
      else
         {c = 75; d = 85;
          pid2 = getpid();
          printf("%d",pid2);
          pid3 = wait(NULL);
          exit(0);}
```

il valore della variabile a vista dal processo figlio subito prima della sua exit è [55] ✔

il valore della variabile c vista dal processo figlio subito prima della sua exit è [70] ✔

il valore della variabile b vista dal processo padre subito prima della sua exit è: [60] ✔

il valore della variabile d vista dal processo padre subito prima della sua exit è: [85] ✔

all'esecuzione delle due printf vale la seguente relazione: pid1 [=] ✔ pid2

del risultato della wait possiamo dire che: pid2 [<] ✔ pid3

65 | 70 | 50 | 60 | 85 | = | 55 | > | 75 | < | 80

Risposta corretta.

La risposta corretta è:
si consideri l'esecuzione della seguente porzione di codice che utilizza la system call fork:

```
int a, b, c, d, n, pid1, pid2, pid3;
a = 50, b = 60, c = 70, d = 80;
n = fork();
      if ( n == 0)
         {a = 55; b = 65;
          pid1 = getppid();
          printf("%d", pid1);
          exit(0);}
      else
         {c = 75; d = 85;
          pid2 = getpid();
          printf("%d",pid2);
          pid3 = wait(NULL);
          exit(0);}

il valore della variabile a vista dal processo figlio subito prima della sua exit è [55]
il valore della variabile c vista dal processo figlio subito prima della sua exit è [70]
il valore della variabile b vista dal processo padre subito prima della sua exit è: [60]
il valore della variabile d vista dal processo padre subito prima della sua exit è: [85]
all'esecuzione delle due printf vale la seguente relazione: pid1 [=] pid2
del risultato della wait possiamo dire che: pid2 [<] pid3
```

Domanda **6**

Risposta corretta

Punteggio ottenuto 2,00 su 2,00

In hard disk grande 512 Gigabyte, per scrivere il numero di un blocco vengono usati 28 bit, arrotondati al minimo numero di byte necessario. L'hard disk adotta una allocazione indicizzata semplice, e di un file A si sa che nel suo blocco indice 16 byte vengono usati per tenere traccia dei blocchi di dati di A. Quanto può essere grande al massimo A?

Scegli un'alternativa:

- a. 12 Kilobyte
- b. 20 Kilobyte
- c. 8 Kilobyte ✔
- d. 16 Kilobyte

Risposta corretta.

La risposta corretta è: 8 Kilobyte

Domanda **7**

Parzialmente corretta

Punteggio ottenuto 1,00 su 2,00

Ricostruite il codice del generico consumatore nel problema dei produttori-consumatori:

full = 0 ; empty = SIZE; mutex = 1;

consumatore:
while (true) {

if emtpy == SIZE wait(emtpy); ✗

wait(mutex); ✓

preleva un elemento dal buffer;

signal(mutex); ✓

signal(full); ✗

consuma item;

}

wait(empty);

wait(full);

if full == 0 signal(empty);

if empty == 0 signal(empty);

signal(empty);

Risposta parzialmente esatta.

Hai selezionato correttamente 2.

La risposta corretta è:

Ricostruite il codice del generico consumatore nel problema dei produttori-consumatori:

full = 0 ; empty = SIZE; mutex = 1;

consumatore:
while (true) {

[wait(full);]

[wait(mutex);]

preleva un elemento dal buffer;

[signal(mutex);]

[signal(empty);]

consuma item;

}

Domanda **8**

Risposta corretta

Punteggio ottenuto 2,00 su 2,00

In un moderno sistema time-sharing con uno spazio di indirizzamento logico maggiore di quello fisico, per quali ragioni un processo può dover transire dallo stato "running" allo stato "waiting"?

Scegli un'alternativa:

a.
1. Perché deve compiere una operazione di I/O
2. Perché si è addormentato su un semaforo
3. Perché ha indirizzato una pagina vittima (page fault)

b.
1. Perché deve compiere una operazione di I/O
2. Perché è stato svegliato su un semaforo
3. Perché ha indirizzato una pagina non presente in RAM (page fault)

c. ✔
1. Perché deve compiere una operazione di I/O
2. Perché si è addormentato su un semaforo
3. Perché ha indirizzato una pagina non presente in RAM (page fault)

d.
1. Perché ha completato una operazione di I/O
2. Perché si è addormentato su un semaforo
3. Perché ha indirizzato una pagina non presente in RAM (page fault)

Risposta corretta.

La risposta corretta è:

1. Perché deve compiere una operazione di I/O
2. Perché si è addormentato su un semaforo
3. Perché ha indirizzato una pagina non presente in RAM (page fault)

Domanda **9**

Risposta errata

Punteggio ottenuto 0,00 su 2,00

Quale/quali tra i seguenti algoritmi di sostituzione delle pagine:

FIFO, LRU, seconda chance, seconda chance migliorata, OPT/MIN

soffre/soffrono dell'anomalia di Belady, e perché?

Scegli un'alternativa:

a. seconda chance e LRU. Infatti questi due algoritmi possono, nel caso peggiore, ridursi all'algoritmo di sostituzione FIFO, che è appunto l'algoritmo che manifesta l'anomalia di belady

b. seconda chance. perché nel caso peggiore può ridursi a FIFO. Per questa ragione è stata introdotta la variante della seconda chance migliorata, proprio per eliminare l'anomalia di Belady.

c. seconda chance e seconda chance migliorata. Infatti questi due algoritmi possono, nel caso peggiore, ridursi all'algoritmo di sostituzione FIFO, che è appunto l'algoritmo che manifesta l'anomalia di belady

d. FIFO e LRU. Infatti, FIFO, è appunto l'algoritmo che manifesta l'anomalia di belady, ed LRU viene implementato usando il reference bit, che in alcune situazioni può introdurre l'anomalia. ✖

Risposta errata.

La risposta corretta è: seconda chance e seconda chance migliorata. Infatti questi due algoritmi possono, nel caso peggiore, ridursi all'algoritmo di sostituzione FIFO, che è appunto l'algoritmo che manifesta l'anomalia di belady

Domanda **10**

Risposta corretta

Punteggio ottenuto 2,00 su 2,00

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

1: cd /tmp
2: mkdir newfolder
3: cd newfolder
4: echo “ciao” > pippo // crea un nuovo file di nome *pippo* contenente la stringa *ciao*
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: 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 3

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

c. 1. il link-counter dell'i-node di *paperino* è: 1
   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 3

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 corretta.

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

Domanda **11**

Risposta corretta

Punteggio ottenuto 2,00 su 2,00

Su un hard disk che adotta una allocazione concatenata (senza FAT) è memorizzato un file A della dimensione di 0x8000 byte, e si sa che nell'ultimo blocco di A sono presenti 32 byte del file. Si sa inoltre che per scrivere il numero di un blocco vengono usati 14 bit, arrotondati al minimo numero di byte necessario. Quanto è grosso l'hard disk?

Scegli un'alternativa:

- a. 128 Megabyte
- b. 32 Megabyte ✔
- c. 256 Megabyte
- d. 64 Megabyte

Risposta corretta.

La risposta corretta è: 32 Megabyte