

Domanda 1

Risposta errata

Punteggio ottenuto 0,0 su 2,0

In quale/i caso/i un processo in coda di ready può decidere di passare allo stato running?

Scegli un'alternativa:

- ☒ a. quando un processo si trova in testa alla coda di ready ✕
- ☐ b. in tutti gli algoritmi di scheduling non preemptive, dove un processo può decidere autonomamente quanto tempo passare in coda di ready
- ☐ c. mai, è il Sistema Operativo che sposta i processi da uno stato all'altro.
- ☐ d. quando un processo esce da un qualsiasi stato di waiting

Risposta errata.

La risposta corretta è: mai, è il Sistema Operativo che sposta i processi da uno stato all'altro.

Domanda 2

Risposta errata

Punteggio ottenuto 0,0 su 2,0

Ricostruite il codice del generico lettore nel problema dei lettori-scrittori:

```
semaphore mutex = 1, scrivi = 1;
```

```
int numlettori = 0;
```

```
Processo lettore {
```

```
wait(mutex);
```

```
if numlettori > 0 wait(scrivi);
```

✘

```
numlettori--;
```

✘

```
signal(mutex);
```

```
... leggi il file ...
```

```
wait(mutex);
```

```
if numlettori <= 1 wait(scrivi);
```

✘

```
numlettori++;
```

✘

```
signal(mutex)
```

```
}
```

```
if numlettori == 0 wait(scrivi);
```

```
if numlettori == 1 signal(scrivi);
```

```
if numlettori == 0 signal(scrivi);
```

```
if numlettori >= 1 wait(scrivi);
```

```
if numlettori == 1 wait(scrivi);
```

Risposta errata.

La risposta corretta è:

Ricostruite il codice del generico lettore nel problema dei lettori-scrittori:

```
semaphore mutex = 1, scrivi = 1;
```

```
int numlettori = 0;
```

```
Processo lettore {
```

```
wait(mutex);
```

```
[numlettori++;
```

```
[if numlettori == 1 wait(scrivi);]
signal(mutex);
... leggi il file ...
wait(mutex);
[numlettori--;]
[if numlettori == 0 signal(scrivi);]
signal(mutex)
}
```

Domanda 3

Risposta non data

Punteggio max.: 2,0

In quale caso, e perché, l'allocazione concatenata (senza FAT) dello spazio in memoria secondaria è particolarmente **svantaggiosa** rispetto ad altre tecniche di allocazione viste?

Scegli un'alternativa:

- ☐ a. Nel caso di file molto grandi, perché se l'hard disk su cui deve essere memorizzato il file è quasi completamente occupato, il SO potrebbe impiegare molto tempo a trovare un insieme di blocchi liberi in cui allocare il file
- ☐ b. Nel caso di file molto grandi, perché in ogni blocco di dati una parte del blocco viene sprecata per memorizzare in puntatore al blocco successivo, risultando così in un notevole spreco di memoria
- ☐ c. Nel caso di file molto grandi, perché l'accesso sequenziale ai vari dati del file risulta molto più lento che nelle altre forme di allocazione dello spazio su disco
- ☐ d. Nel caso di file molto grandi, perché per leggere i dati contenuti nell'ultima parte del file occorre seguire la catena di blocchi sull'hard disk, ossia effettuare molte operazioni di I/O su un dispositivo lento.

Risposta errata.

La risposta corretta è: Nel caso di file molto grandi, perché per leggere i dati contenuti nell'ultima parte del file occorre seguire la catena di blocchi sull'hard disk, ossia effettuare molte operazioni di I/O su un dispositivo lento.

Domanda 4

Risposta corretta

Punteggio ottenuto 2,0 su 2,0

In quale caso l'accesso in lettura ad un file memorizzato su un sistema RAID non è più veloce che se il file fosse memorizzato su un normale hard disk?

Scegli un'alternativa:

- ☒ a. quando il file è memorizzato su uno più blocchi appartenenti a strip contenuti sullo stesso disco del RAID (e il RAID usato non è di tipo 01/10, poiché in questo caso, se il file è memorizzato su almeno due strip, si può sfruttare il disco di mirroring) ✓
- ☐ b. quando il file è memorizzato su dischi che contengono anche strip di parità, come nel livello 5, perché in questo caso la ricostruzione dei dati di cui è composto il file richiede una certa quantità di lavoro in più da parte del controller del RAID
- ☐ c. quando il file occupa molti strip, perché allora neppure la presenza di dischi di mirroring può mitigare la necessità di accedere più volte al contenuto dei vari dischi per leggere tutti gli strip in cui è suddiviso il file
- ☐ d. mai: infatti, i sistemi RAID sono stati progettati proprio per fornire più elevate velocità di accesso ai dati che contengono (oltre che per garantire una maggiore affidabilità)

Risposta corretta.

La risposta corretta è: quando il file è memorizzato su uno più blocchi appartenenti a strip contenuti sullo stesso disco del RAID (e il RAID usato non è di tipo 01/10, poiché in questo caso, se il file è memorizzato su almeno due strip, si può sfruttare il disco di mirroring)

Domanda 5

Risposta errata

Punteggio ottenuto -1,5 su 3,0

In un sistema paginato è noto che lo spreco di memoria primaria dovuto alla frammentazione interna è in media di circa 1 Kbyte per processo, e un indirizzo logico è scritto su 28 bit. Se la tabella delle pagine più grande di questo sistema è grande 256 Kilobyte, quanto può essere grande al massimo lo spazio di indirizzamento fisico del sistema?

Scegli un'alternativa:

- ☐ a. 64 Megabyte
- ☐ b. 128 Megabyte
- ☒ c. 512 Megabyte ✕
- ☐ d. 256 Megabyte

Risposta errata.

La risposta corretta è: 128 Megabyte

Domanda 6

Parzialmente corretta

Punteggio ottenuto 0,5 su 2,0

In un sistema che adotta la paginazione della memoria, un indirizzo logico è scritto su "P" bit, e il numero di pagina è scritto su "k" bit. Lo spazio fisico è invece suddiviso in 2^{24} frame.

Possiamo quindi dire che:

un frame del sistema è grande: $2^{(P-k)}$ ✓ byte

lo spazio fisico del sistema è grande: $(2^{24}) * 3$ ✗ byte

la tabella delle pagine più grande del sistema ha una dimensione di: $2^{(24-k)}$ ✗ byte

il sistema dovrà implementare la memoria virtuale solo se: $P < (24 + P-k)$ ✗

$$2^{(24 + P-k)}$$

$$(2^k) * 3$$

$$2^{(24-P+k)}$$

$$2^{(P+24)}$$

$$P > (24 + P-k)$$

Risposta parzialmente esatta.

Hai selezionato correttamente 1.

La risposta corretta è:

In un sistema che adotta la paginazione della memoria, un indirizzo logico è scritto su "P" bit, e il numero di pagina è scritto su "k" bit. Lo spazio fisico è invece suddiviso in 2^{24} frame.

Possiamo quindi dire che:

un frame del sistema è grande: $[2^{(P-k)}]$ byte

lo spazio fisico del sistema è grande: $[2^{(24 + P-k)}]$ byte

la tabella delle pagine più grande del sistema ha una dimensione di: $[(2^k) * 3]$ byte

il sistema dovrà implementare la memoria virtuale solo se: $[P > (24 + P-k)]$

Domanda 7

Risposta corretta

Punteggio ottenuto 2,0 su 2,0

tra gli **svantaggi** della paginazione della memoria troviamo:

Scegli un'alternativa:

- ☐ a.
 - 1. le page table dei processi occupano spazio in RAM
 - 2. il SO lavora di più perché deve gestire le PT dei processi e l'elenco dei frame liberi in RAM
 - 3. il context switch tra processi richiede più tempo perché il SO deve ogni volta ricostruire la PT del processo che entra in esecuzione
- ☐ b.
 - 1. le page table dei processi occupano spazio nell'area di SWAP
 - 2. il SO lavora di più perché deve gestire le PT dei processi e l'elenco dei frame liberi in RAM
 - 3. il context switch tra processi richiede più tempo perché il SO deve attivare la PT del processo che entra in esecuzione
- ☒ c.
 - 1. le page table dei processi occupano spazio in RAM ✓
 - 2. il SO lavora di più perché deve gestire le PT dei processi e l'elenco dei frame liberi in RAM
 - 3. il context switch tra processi richiede più tempo perché il SO deve attivare la PT del processo che entra in esecuzione
- ☐ d.
 - 1. le page table dei processi occupano spazio in RAM
 - 2. il SO lavora di più per cercare di allocare le pagine dei processi nei frame liberi in modo da produrre la frammentazione interna minima
 - 3. il context switch tra processi richiede più tempo perché il SO deve attivare la PT del processo che entra in esecuzione

Risposta corretta.

La risposta corretta è:

- 1. le page table dei processi occupano spazio in RAM
- 2. il SO lavora di più perché deve gestire le PT dei processi e l'elenco dei frame liberi in RAM
- 3. il context switch tra processi richiede più tempo perché il SO deve attivare la PT del processo che entra in esecuzione

Domanda 8

Risposta errata

Punteggio ottenuto -1,0 su 2,0

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

- 1: cd /tmp
- 2: mkdir newfolder
- 3: 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: 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
- ☐ b.
 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
- ☒ c.
 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
- ☐ d.
 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

Risposta errata.

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 9

Risposta corretta

Punteggio ottenuto 3,0 su 3,0

In un sistema operativo che adotta uno scheduling con 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	8
Pb	2	8
Pc	4	2
Pd	6	1

Qual è il waiting time medio ottenuto per lo scheduling dei quattro processi della tabella se si usa l'algoritmo di scheduling preemptive che fornisce il miglior turnaround time possibile? Qual è il corrispondente diagramma di GANTT?

Scegli un'alternativa:

- ☐ a. Diagramma di GANTT: (0) ... Pa ... (4) ... Pc ... (6) ... Pa ... (10) ... Pd ... (11) ... Pc ... (19)

Waiting time medio = 4

- ☒ b. Diagramma di GANTT: (0) ... Pa ... (4) ... Pc ... (6) ... Pd ... (7) ... Pa ... (11) ... Pb ... (19) ✓

Waiting time medio = 3

- ☐ c. Diagramma di GANTT: (0) ... Pa ... (4) ... Pb ... (7) ... Pd ... (9) ... Pa ... (11) ... Pc ... (19)

Waiting time medio = 3

- ☐ d. Diagramma di GANTT: (0) ... Pa ... (4) ... Pb ... (7) ... Pc ... (10) ... Pa ... (11) ... Pd ... (19)

Waiting time medio = 4