

Home ► Anno Accademico 20/21 ► Secondo anno Laurea DM270 ► SOA-20-21 ►

SCRITTI ONLINE DEL CORSO DI SISTEMI OPERATIVI A ►

Scritto parte di teoria del corso A di Sistemi Operativi del 26 gennaio 2021

Iniziato martedì, 26 gennaio 2021, 10:01

Stato Completato

Terminato martedì, 26 gennaio 2021, 11:20

Tempo impiegato 1 ora 19 min.

Valutazione 9,0 su un massimo di 20,0 (45%)

Domanda 1

Risposta corretta

Punteggio
ottenuto 2,0 su
2,0

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) ... Pb ... (7) ... Pd ... (9)
... Pa ... (11) ... Pc ... (19)

Waiting time medio = 3

 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) ... Pc ... (10)
... Pa ... (11) ... Pd ... (19)

Waiting time medio = 4

 d.

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

Waiting time medio = 4

Risposta corretta.

La risposta corretta è:

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

Waiting time medio = 3

Domanda 2

Risposta errata

Punteggio
ottenuto 0,0 su
2,0

Tre processi P_A , P_B e P_C eseguono il seguente codice:

Shared Var semaphore mutex = 1; semaphore done = 1

P_A :

repeat forever:
 wait(done)
 wait(done)
 wait(done)
 wait(mutex)
 A
 signal(mutex)
 signal(done)

P_B :

repeat forever:
 wait(done)
 wait(mutex)
 B
 signal(mutex)
 signal(done)

P_C :

repeat forever:
 wait(mutex)
 C
 signal(mutex)
 signal(done)

L'esecuzione concorrente di P_A , P_B e P_C produce una sequenza (di lunghezza indefinita) di chiamate alle procedure A, B e C. Quale/quali delle sequenze riportate nelle opzioni di risposta possono essere la porzione iniziale di sequenze prodotte dall'esecuzione concorrente di P_A , P_B e P_C ?

Scegli un'alternativa:

a.

[x] C,B,C,C,B,C,A,B,C [x] B,C,C,A,C,C,B,C,A [x] C,C,B,C,A,C,A,C,B
[] C,C,B,A,C,B,A,B,C

b.

[x] C,B,C,C,B,C,A,B,C [] B,C,C,A,C,C,B,C,A [x] C,C,B,C,A,C,A,C,B [] C,C,B,A,C,B,A,B,C X

c.

[] C,B,C,C,B,C,A,B,C [x] B,C,C,A,C,C,B,C,A [x] C,C,B,C,A,C,A,C,B [] C,C,B,A,C,B,A,B,C

d.

[] C,B,C,C,B,C,A,B,C [x] B,C,C,A,C,C,B,C,A [x] C,C,B,C,A,C,A,C,B
[x] C,C,B,A,C,B,A,B,C

Risposta errata.

La risposta corretta è:

[x] C,B,C,C,B,C,A,B,C [x] B,C,C,A,C,C,B,C,A [x] C,C,B,C,A,C,A,C,B [] C,C,B,A,C,B,A,B,C

Domanda 3

Risposta corretta

Punteggio
ottenuto 1,0 su
1,0

Un sistema ha un tempo di accesso in RAM di 100 ns, adotta un TLB con un tempo di accesso di 20 ns e un hit rate del 90%, usa una paginazione a due livelli e non ha bisogno di usare un algoritmo di rimpiazzamento delle pagine. Qual è il tempo medio di accesso alla RAM (*medium access time - mat*) del sistema? (per semplicità in caso di miss si ignori il costo di interrogazione del TLB)

Scegli un'alternativa:

- a. mat = 118 ns
- b. mat = 148 ns
- c. mat = 128 ns
- d. mat = 138 ns ✓

Risposta corretta.

La risposta corretta è: mat = 138 ns

Domanda 4

Risposta errata

Punteggio
ottenuto 0,0 su
1,0

Un hard disk ha la dimensione di 512 Gigabyte, è formattato in blocchi da 0x100 byte e adotta una qualche forma di allocazione indicizzata dello spazio su disco. Sull'hard disk è memorizzato un file A della dimensione di 75 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. $2^{39}/2^9 = 2^{30}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 32 Kbyte di dati. Assumendo una allocazione indicizzata a schema concatenato, ci vogliono 3 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 4 operazioni di lettura su disco
- b. $2^{39}/2^9 = 2^{30}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 32 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 blocco indice più esterno, saranno necessarie in tutto 3 operazioni di lettura su disco
- c. $2^{39}/2^8 = 2^{31}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 16 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
- d. $2^{39}/2^8 = 2^{31}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 16 Kbyte di dati. Assumendo una allocazione indicizzata a schema concatenato, ci vogliono 4 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 5 operazioni di lettura su disco

✗

Risposta errata.

La risposta corretta è: $2^{39}/2^8 = 2^{31}$, dunque un blocco indice di questo hard disk può tenere traccia di un massimo di 16 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 5

Risposta corretta

Punteggio
ottenuto 1,0 su
1,0

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. il processo running esegue una wait e si addormenta sul semaforo
2. il processo running genera page fault
3. il processo running genera una trap e viene terminato
4. il processo running inizia una operazione di I/O



b.

1. il processo running esegue una signal e si addormenta sul semaforo
2. il processo running genera page fault
3. il processo running genera una trap e viene terminato
4. il processo running inizia una operazione di I/O

c.

1. il processo running esegue una wait e si addormenta sul semaforo
2. il processo running genera una pagina vittima
3. il processo running genera una trap e viene terminato
4. il processo running inizia una operazione di I/O

d.

1. il processo running esegue una wait e si addormenta sul semaforo
2. il processo running genera page fault
3. il processo running genera una trap e viene terminato
4. il processo in coda di ready inizia una operazione di I/O

Risposta corretta.

La risposta corretta è:

1. il processo running esegue una wait e si addormenta sul semaforo
2. il processo running genera page fault
3. il processo running genera una trap e viene terminato
4. il processo running inizia una operazione di I/O

Domanda 6

Risposta errata

Punteggio
ottenuto 0,0 su
1,0

Un hard disk ha la dimensione di 512 Gigabyte, è formattato in blocchi da 0x2000 byte e adotta una allocazione concatenata (senza FAT) dello spazio su disco. Sull'hard disk è memorizzato un file A della dimensione di 0x4000 byte. Quanti byte del file sono memorizzati nell'ultimo blocco di dati del file? (selezionate l'opzione di risposta che riporta il calcolo corretto del valore cercato)

Scegli un'alternativa:

- a. $2^{39}/2^{13} = 2^{26} \Rightarrow 4$ byte per scrivere un puntatore a blocco. Dunque, l'ultimo blocco di A conterrà $4 * 2^{14}/2^{13} = 8$ byte di A
- b. $2^{39}/2^{13} = 2^{26} \Rightarrow 4$ byte per scrivere un puntatore a blocco. Dunque, l'ultimo blocco di A conterrà $2^{15}/4 = 8192$ byte di A
- c. $2^{39}/2^{15} = 2^{24} \Rightarrow 3$ byte per scrivere un puntatore a blocco. Dunque, l'ultimo blocco di A conterrà $3 * 2^{15}/2^{14} = 6$ byte di A
- d. $2^{39}/2^{13} = 2^{26} \Rightarrow 4$ byte per scrivere un puntatore a blocco. Dunque, l'ultimo blocco di A conterrà $2^{14}/4 = 4096$ byte di A

✗

Risposta errata.

La risposta corretta è: $2^{39}/2^{13} = 2^{26} \Rightarrow 4$ byte per scrivere un puntatore a blocco. Dunque, l'ultimo blocco di A conterrà $4 * 2^{14}/2^{13} = 8$ byte di A

Domanda 7

Risposta corretta

Punteggio
ottenuto 1,0 su
1,0

Dell'allocazione concatenata con FAT dello spazio sull'hard disk possiamo dire che tra i suoi **vantaggi** e i suoi **svantaggi** troviamo:

Scegli un'alternativa:

 a.

1. che permette di tenere traccia anche dei blocchi liberi dell'hard disk
2. che permette un accesso diretto ai file efficiente
3. che sottrae spazio in RAM ai processi utente

 b.

1. che permette di tenere traccia anche dei blocchi liberi dell'hard disk
2. che permette un accesso diretto ai file efficiente solo per i file molto grossi
3. che sottrae spazio in RAM ai processi utente

 c.

1. che l'accesso diretto ai file è inefficiente perché si deve comunque seguire una catena di puntatori
2. che sottrae spazio in ram ai processi utente
3. che produce molta frammentazione interna

 d.

1. che permette di tenere traccia anche dei blocchi liberi dell'hard disk
2. che permette un accesso diretto ai file efficiente
3. che non è efficiente nel caso di allocazione indicizzata a più livelli

Risposta corretta.

La risposta corretta è:

1. che permette di tenere traccia anche dei blocchi liberi dell'hard disk
2. che permette un accesso diretto ai file efficiente
3. che sottrae spazio in RAM ai processi utente

Domanda 8

Risposta errata

Punteggio

ottenuto 0,0 su

1,0

Che vantaggio da l'uso dei thread al posto dei processi?

Scegli un'alternativa:

- a. un insieme di peer thread condivide lo spazio di indirizzamento. Per questa ragione, i vari thread non hanno bisogno di ricorrere alla memoria virtuale, la creazione di un nuovo peer thread richiede molto meno tempo della corrispondente operazione sui processi.
- b. un insieme di peer thread condivide lo spazio di indirizzamento. Per questa ragione, i vari thread non devono usare meccanismi di sincronizzazione per accedere alle variabili condivise, e il context switch tra peer thread richiede molto meno tempo della corrispondente operazione tra processi. X
- c. un insieme di peer thread condivide lo spazio di indirizzamento. Per questa ragione, i vari thread non hanno bisogno di ricorrere alla memoria virtuale, e l'esecuzione dei vari thread richiede meno tempo di un insieme di processi che esegua lo stesso codice dei thread.
- d. un insieme di peer thread condivide lo spazio di indirizzamento. Per questa ragione, il context switch tra peer thread e la creazione di un nuovo peer thread richiedono molto meno tempo delle corrispondenti operazioni sui processi.

Risposta errata.

La risposta corretta è: un insieme di peer thread condivide lo spazio di indirizzamento. Per questa ragione, il context switch tra peer thread e la creazione di un nuovo peer thread richiedono molto meno tempo delle corrispondenti operazioni sui processi.

Domanda 9

Risposta errata

Punteggio
ottenuto 0,0 su

2,0

In un sistema operativo un indirizzo fisico è scritto su 28 bit, l'offset più grande in una pagina è 3FFF, lo spazio logico è il doppio di quello fisico, e nel sistema possono essere presenti contemporaneamente al massimo 1024 processi.

Se il sistema adottasse una Inverted Page Table, quanto sarebbe grande questa tabella? (selezionate l'opzione di risposta che riporta il ragionamento aritmetico e il risultato corretti)

Scegli un'alternativa:

- a. Ogni entry della IPT è grande 24 bit, ossia 3 byte, e dunque la IPT sarà grande $3 * 2^{14} = 48$ Kbyte (circa) X
- b. Ogni entry della IPT è grande 24 bit, ossia 3 byte, e dunque la IPT sarà grande $3 * 2^{15} = 96$ Kbyte (circa)
- c. Ogni entry della IPT è grande 25 bit, arrotondati a 4 byte, e dunque la IPT sarà grande $4 * 2^{14} = 64$ Kbyte (circa)
- d. Ogni entry della IPT è grande 25 bit, arrotondati a 4 byte, e dunque la IPT sarà grande $4 * 2^{15} = 128$ Kbyte (circa)

Risposta errata.

La risposta corretta è: Ogni entry della IPT è grande 25 bit, arrotondati a 4 byte, e dunque la IPT sarà grande $4 * 2^{14} = 64$ Kbyte (circa)

Domanda 10

Risposta corretta

Punteggio

ottenuto 2,0 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 ../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 *tmp* è: aumentato di 2
3. l'output del comando 10 è: "ciao"
4. il comando 6 da come risultato: un errore perché non sono ammessi hard link tra cartelle

 b.

1. il link-counter dell'i-node di *paperino* è: 1
2. il link counter di *tmp* è: 2
3. l'output del comando 10 è: no such file or directory
4. il comando 6 da come risultato: un errore perché non sono ammessi hard link tra cartelle

 c.

1. il link-counter dell'i-node di *paperino* è: 2
2. il link counter di *tmp* è: aumentato di 1
3. l'output del comando 10 è: "ciao" seguito da "salve"
4. il comando 6 da come risultato: un errore perché non sono ammessi hard link tra cartelle

 d.

1. il link-counter dell'i-node di *paperino* è: 2
2. il link counter di *tmp* è: aumentato di 2
3. l'output del comando 10 è: "ciao"
4. il comando 6 da come risultato: un nuovo collegamento alla cartella *newfolder*

Risposta corretta.

La risposta corretta è:

1. il link-counter dell'i-node di *paperino* è: 1
2. il link counter di *tmp* è: aumentato di 2
3. l'output del comando 10 è: "ciao"
4. il comando 6 da come risultato: un errore perché non sono ammessi hard link tra cartelle

Domanda 11

Risposta errata

Punteggio
ottenuto 0,0 su
1,0

Perché i sistemi operativi moderni non usano l'allocazione contigua dello spazio in RAM a partizioni fisse?

Scegli un'alternativa:

- a. perché limita a priori il grado di multiprogrammazione, e lungo la vita del sistema si producono sempre più "buchi" di RAM liberi ma non adiacenti e sempre più piccoli, e quindi non utilizzabili.
- b. perché limita a priori il grado di multiprogrammazione e soffre in modo eccessivo del problema della frammentazione interna, costringendo periodicamente al ricompattamento dello spazio in RAM.

- c. perché limita a priori il numero massimo di processi che possono essere contemporaneamente presenti nel sistema, e soffre in modo eccessivo del problema della frammentazione interna.
- d. perché si adatta male alla paginazione della memoria, dato che il numero di partizioni prestabilito limita il numero massimo di pagine dei processi che possono stare contemporaneamente in RAM

Risposta errata.

La risposta corretta è: perché limita a priori il numero massimo di processi che possono essere contemporaneamente presenti nel sistema, e soffre in modo eccessivo del problema della frammentazione interna.

Domanda 12

Risposta corretta

Punteggio

ottenuto 1,0 su

1,0

Dire che un processo è nello stato “waiting for page” significa:

Scegli un'alternativa:

- a. che mentre era in esecuzione ha indirizzato una pagina non presente in ram, ed è stato messo nello stato indicato nell'attesa che il SO recuperi e trasferisca in ram la pagina mancante ✓
- b. che mentre era in esecuzione è stato temporaneamente fermato (e messo nello stato indicato) perché un altro processo ha indirizzato una pagina non presente in ram che il SO deve subito recuperare
- c. che mentre era nella coda di ready ha indirizzato una pagina non presente in ram, ed è stato messo nello stato indicato nell'attesa che il SO recuperi e trasferisca in ram la pagina mancante
- d. che il processo ha richiesto al SO l'accesso ad una pagina mancante, e si è messo nello stato indicato in attesa che la pagina sia stata recuperata dal SO

Risposta corretta.

La risposta corretta è: che mentre era in esecuzione ha indirizzato una pagina non presente in ram, ed è stato messo nello stato indicato nell'attesa che il SO recuperi e trasferisca in ram la pagina mancante

Domanda 13

Risposta errata

Punteggio

ottenuto 0,0 su

1,0

Un algoritmo di scheduling a code multiple con retroazione:

Scegli un'alternativa:

- a. usa più code di ready, gestite ciascuna con una diversa politica di scheduling. Un processo può essere promosso a una coda con priorità superiore se ha consumato completamente il suo ultimo quanto di tempo
- b. usa più code di ready, gestite ciascuna con una diversa politica di scheduling. Un processo può essere spostato da una coda all'altra in base a come si è comportato l'ultima volta che gli è stata assegnata la CPU
- c. usa più code di ready, gestite ciascuna con una diversa politica di scheduling. Un processo può essere retrocesso a una coda con priorità inferiore se non ha consumato completamente il suo ultimo quanto di tempo ✗
- d. usa più code di ready, in cui i processi vengono inseriti a seconda che siano processi CPU o I/O bound, e processi in foreground o in background

Risposta errata.

La risposta corretta è: usa più code di ready, gestite ciascuna con una diversa politica di scheduling. Un processo può essere spostato da una coda all'altra in base a come si è comportato l'ultima volta che gli è stata assegnata la CPU

Domanda 14

Risposta corretta

Punteggio

ottenuto 1,0 su
1,0

Sia A un file di testo all'interno di un file system Unix. Viene eseguito con successo il comando:

`ln -s A B`

dove B non esisteva prima dell'esecuzione del comando. Che cosa succede nelle strutture interne al sistema?

Scegli un'alternativa:

- a. nella cartella in cui viene eseguito il comando viene creato un nuovo file di nome B, e Il link counter dell'i-node associato a B viene inizializzato a 1. Dentro all'i-node di A viene scritto il pathname usato come secondo argomento del comando
- b. Una nuova entry di nome B viene inserita nella cartella in cui è stato eseguito il comando, e un nuovo i-node viene associato a B. il pathname usato come primo argomento del comando viene scritto dentro l'i-node associato a B.

- c. Una nuova entry di nome B viene inserita nella cartella in cui è stato eseguito il comando. Il numero dell'i-node associato ad A viene ora associato anche a B. Il link counter di quell'i-node viene incrementato di 1
- d. il contenuto di A viene copiato dentro a B. L'i-node di A viene duplicato e associato a B. Il link counter dell'i-node associato a B viene inizializzato a 1, mentre il link counter dell'i-node associato ad A viene incrementato di 1

Risposta corretta.

La risposta corretta è: Una nuova entry di nome B viene inserita nella cartella in cui è stato eseguito il comando, e un nuovo i-node viene associato a B. il pathname usato come primo argomento del comando viene scritto dentro l'i-node associato a B.

Domanda 15

Risposta errata

Punteggio
ottenuto 0,0 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 8 byte usando però solo i primi 20 bit, la frammentazione interna media prodotta da un processo del sistema è di circa 32 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 $2^{15}/8$ entry, e dunque lo spazio logico del sistema è grande 128 Megabyte.
Lo spazio fisico è suddiviso in 2^{20} frame, e dunque ha una dimensione di 32 Gigabyte
----- 
- b. La tabella delle pagine più grande del sistema ha $2^{16}/8$ entry, e dunque lo spazio logico del sistema è grande 512 Megabyte.
Lo spazio fisico è suddiviso in 2^{20} frame, e dunque ha una dimensione di 64 Gigabyte

- c. La tabella delle pagine più grande del sistema ha $2^{16}/2$ entry, e dunque lo spazio logico del sistema è grande 4 Gigabyte.
Lo spazio fisico è suddiviso in 2^{20} frame, e dunque ha una dimensione di 64 Gigabyte

- d. La tabella delle pagine più grande del sistema ha $2^{16}/2$ entry, e dunque lo spazio logico del sistema è grande 4 Gigabyte.
Lo spazio fisico è suddiviso in 2^{20} frame, e dunque ha una dimensione di 1 Megabyte

Risposta errata.

La risposta corretta è: La tabella delle pagine più grande del sistema ha $2^{16}/8$ entry, e dunque lo spazio logico del sistema è grande 512 Megabyte.

Lo spazio fisico è suddiviso in 2^{20} frame, e dunque ha una dimensione di 64 Gigabyte
