SISTEMI OPERATIVI – 13 febbraio 2020 corso A

Cognome:_____**Nome:**_____
Matricola:_____

Ricordate che non potete usare calcolatrici o materiale didattico, e che potete consegnare al massimo tre prove scritte per anno accademico.

ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

ESERCIZIO 1 (7 punti)

a) Si consideri il problema dei lettori e scrittori visto a lezione, dove i codici del generico scrittore e del generico lettore sono riportati qui di seguito.

Inserite le istruzioni mancanti necessarie per il funzionamento del sistema secondo la soluzione vista a lezione, indicando il semaforo (o i semafori) mancante/i e il relativo valore di inizializzazione.

semafori e variabili condivise necessarie con relativo valore di inizializzazione:
semaphore mutex = 1;
semaphore scrivi = 1;
int numlettori = 0;

"scrittore"
{
wait(scrivi);
Esegui la scrittura del file
signal(scrivi)
}

"lettore"
{
wait(mutex);

numlettori++;

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

signal(mutex);

... leggi il file ...

wait(mutex);

numlettori--;

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

signal(mutex);
}

b) Elencate almeno quattro 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:

scade il quanto di tempo del processo running
entra in coda di ready un processo con priorità maggiore di quello in stato running
processo running esegue una operazione di I/O
processo running esegue una wait e si addormenta sul semaforo
processo running genera una trap e viene terminato
processo running genera page fault
processo running termina di eseguire il suo codice (in tutti i casi eccetto il secondo, se c'è almeno un altro processo utente in RQ)

c) Che cosa vuol dire che un algoritmo di scheduling soffre di starvation?

Che non garantisce che, in un tempo finito, un processo in coda di ready venga selezionato per entrare in esecuzione e terminare.

d) descrivete brevemente le tre proprietà che deve possedere una corretta soluzione al problema della sezione critica.

Si vedano i lucidi della sezione 6.2

e) Quale ulteriore proprietà è desiderabile in un qualsiasi meccanismo di sincronizzazione, per evitare inutile spreco di tempo di CPU?

Che non sia basato sul busy waiting

f) Perché non è possibile risolvere il problema della sezione critica permettendo al processo utente che deve entrare in sezione critica di disabilitare gli interrupt per tutto il periodo di tempo in cui rimane nella sezione critica?

Perché questo significherebbe lasciare il controllo dell'intera macchina al processo utente, nel caso in quest'ultimo si "dimenticasse" di riabilitare gli interrupt.

ESERCIZIO 2 (7 punti)

a) un sistema paginato usa 40 bit per scrivere un indirizzo fisico e l'offset più grande in un indirizzo è 7FF. Il sistema adotta una paginazione a due livelli, e si sa che la tabella delle pagine esterna più grande del sistema occupa esattamente un frame. Quanto è grande lo spazio di indirizzamento logico del sistema?

Frame e pagine del sistema sono grandi 2^{11} byte, e quindi lo spazio di indirizzamento fisico è suddiviso in $2^{40}/2^{11} = 2^{29}$ frame, e ci vogliono 4 byte per scrivere il numero di un frame in una qualsiasi entry di una tabella delle pagine.

Dunque la tabella esterna più grande del sistema, che ha la dimensione di un frame, contiene $2^{11}/2^2 = 2^9$ entry, corrispondenti al numero di frame occupati dalla tabella delle pagine interna più grande del sistema, che quindi ha dimensione $2^9 * 2^{11} = 2^{20}$ byte. Questa tabella quindi contiene $2^{20}/2^2$ entry, che corrispondono al numero di pagine dello spazio di indirizzamento logico. Tale spazio ha quindi la dimensione di $2^{18} * 2^{11} = 2^{29}$ byte.

b) Si sa che il sistema del punto a) ha un tempo di accesso in RAM di 100 ns, e che è dotato di un TLB perfetto (tempo di accesso pari a 0) e un hit rate medio del 90%. Supponendo che non sia implementata la memoria virtuale, qual è l'effettivo tempo medio di accesso in RAM dovuto alla presenza del sistema di

paginazione? (è sufficiente riportare la formula che esprime il calcolo, e si ricorda che il TLB contiene sempre una porzione della PT interna)

ma = 0,90 * 100ns + 0,1 * (100+100+100)ns = 120ns

c) Cosa rende un algoritmo di rimpiazzamento delle pagine un buon algoritmo di rimpiazzamento?

La capacità di scegliere come pagina vittima una pagina che non verrà più indirizzata, o verrà indirizzata il più in là possibile nel tempo (risposta alternativa: la capacità di minimizzare il numero di page faults)

d) Che cos'è il Thrashing?

Un comportamento degenere dei processi che passano la maggior parte del loro tempo a generare page faults e ad attendere che la pagina mancante arrivi in RAM anziché a portare avanti la loro computazione.

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

Non usano l'allocazione a partizioni fisse perché limita a priori il grado di programmazione, soffre in modo eccessivo del problema della frammentazione interna, e un processo di dimensioni maggiori di quelle della partizione più grande non può comunque essere eseguito.

f) perché i sistemi operativi moderni non usano l'allocazione contigua dello spazio in RAM a partizioni variabili?

Non usano l'allocazione a partizioni variabili perché soffre del problema della frammentazione esterna e costringe periodicamente al ricompattamento dello spazio in RAM.

ESERCIZIO 3 (6 punti)

a) In Unix che cosa succede nelle strutture interne al sistema, quando viene creato un nuovo link fisico ad un file A già esistente?

Viene semplicemente incrementato di 1 il link counter dell'index-node associato al file A. Una entry viene aggiunta nella cartella in cui è stato creato il nuovo link fisico.

b) In Unix che cosa succede nelle strutture interne al sistema, quando viene creato un nuovo link simbolico ad un file A già esistente?

Viene creato un nuovo index node associato al nome usato per il link simbolico. All'interno del nuovo index-node viene scritto il pathname di A specificato nel comando.

c) perché sono necessari i link simbolici?

Per permettere di costruire collegamenti fra le directory, tra le quali non sono ammessi i link fisici (in Unix, i link simbolici permettono anche link fra file che stanno su partizioni diverse degli hard disk).

d) In quale caso, e perché, l'allocazione indicizzata dello spazio in memoria secondaria è particolarmente **svantaggiosa**?

Nel caso di file molto piccoli, che ad esempio occupano solo un blocco, perché per tenere traccia di quel blocco si preca quasi completamente il blocco indice, che va comunque allocato.

e) In quali casi, rispettivamente, è meglio usare un sistema RAID nella configurazione 0, 01 o 5?

Usiamo il RAID 0 se abbiamo bisogno di massimizzare lo spazio di memorizzazione disponibile e la velocità di accesso ai dati, mentre l'affidabilità non è un requisito fondamentale.
Usiamo il RAID 01 se abbiamo bisogno della massima affidabilità e velocità di accesso ai dati.
Usiamo il RAID 5 se vogliamo un ragionevole compromesso tra affidabilità, velocità di accesso ai dati, e spazio di memorizzazione disponibile.

f) 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?

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, poiché in questo caso, se il file è memorizzato su almeno due strip, si può sfruttare il disco di mirroring)