SISTEMI OPERATIVI E LABORATORIO
(Indirizzo Sistemi e Reti) - 7 luglio 2009

Cognome:_____Nome:_____
Matricola:_____

1. Ricordate che non potete usare calcolatrici o materiale didattico, e che se consegnate annullate automaticamente qualsiasi voto conseguito in una prova precedente.
2. Gli **studenti in corso che decidono di sostenere solo la parte di teoria o solo la parte di laboratorio** devono consegnare i loro elaborati al termine della prima ora dello scritto.
3. Gli **studenti in corso che decidono di sostenere l'intero scritto (teoria + laboratorio)** ricordino che se prendono in una delle due parti un voto X insufficiente e:
    a. $X < 10$: non possono presentarsi al successivo scritto di luglio
    b. $10 \leq X \leq 14$: devono risostenere l'intero scritto, anche se nell'altra parte hanno preso un voto maggiore o uguale a 18
    c. $14 < X$: possono riprovare la sola parte insufficiente nel secondo scritto di luglio (ovviamente se nell'altra parte hanno raggiunto la sufficienza)
4. Gli **studenti degli anni precedenti** devono sostenere l'intero scritto.
5. Si ricorda che a partire dallo scritto di settembre non è più possibile sostenere separatamente l'esame delle due parti del corso. Chi non ha conseguito la sufficienza in *entrambe* le parti entro il secondo scritto di luglio, da settembre deve comunque ridare l'intero scritto.

ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

(il punteggio conseguito farà media con quello ottenuto nella parte di laboratorio. E' comunque necessario prendere almeno 18 punti per considerare passata la parte di teoria.)

ESERCIZIO 1 (9 punti)

All'atto dell'installazione di un SO su una macchina, è possibile scegliere se usare pagine grandi 2^{10} byte oppure grandi 2^{20} byte. La dimensione dello spazio di indirizzamento logico rimane comunque la stessa. La macchina su cui viene installato il SO usa 32 bit per scrivere un indirizzo fisico, e lo spazio di indirizzamento fisico è la metà dello spazio di indirizzamento logico (nel seguito, motivate tutte le risposte che date, esplicitando tutti i calcoli ed eventuali assunzioni fatte).

a) Nel caso di pagine da 2^{20} byte, quanto può essere grossa, al massimo, la page table di un processo?

Si osservi innanzi tutto che lo spazio di indirizzamento logico è pari a 2^{33} byte.

Una page table può avere al massimo $2^{33} / 2^{20}$ entry, mentre il numero massimo di frame del sistema è pari a $2^{32} / 2^{20} = 2^{12}$. Quindi, abbiamo bisogno di 12 bit per scrivere il numero di un frame. Se per semplicità assumiamo che si usino due byte per ciascuna entry di una page table, la sua dimensione massima sarà di di $2 * 2^{33} / 2^{20} = 2^{14}$ byte = 16 Kbyte

b) Se le pagine sono grandi 2^{20}, il sistema potrebbe dover usare una paginazione a più livelli? E se invece sono grandi 2^{10} byte?

pagine da 2^{20} byte: dalla domanda precedente sappiamo che la PT di un processo può essere grande al massimo 16 kbyte, e può quindi sicuramente essere contenuta in un'unica pagina. Non è necessaria la paginazione su più livelli.

pagine da 2^{10} byte: una page table può avere fino a $2^{33} / 2^{10} = 2^{23}$ entry. Lo spazio di indirizzamento fisico è suddiviso in $2^{32} / 2^{10} = 2^{22}$ frame, e 3 byte sono sufficienti per scrivere il numero di un qualsiasi frame. Al massimo una page table occupa quindi $3 * 2^{23}$ byte = pari a circa 24 Mbyte. Poiché un frame è molto più piccolo, la paginazione su più livelli potrebbe essere necessaria.

c) E' corretto dire che "il progettista di questo sistema operativo deve scegliere se adottare una allocazione locale o globale dei frame"? (motivate la vostra risposta)

Si. Poiché lo spazio di indirizzamento logico è maggiore di quello fisico, il SO deve implementare la memoria virtuale, e quindi è necessario un algoritmo di rimpiazzamento che sceglierà la pagina vittima tra i frame assegnati al processo che ha generato il page fault (in caso di allocazione locale) o tra tutti i frame assegnati ad un qualsiasi processo (in caso di allocazione globale)

ESERCIZIO 2 (9 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 operazioni di wait e signal mancanti necessarie per il corretto funzionamento del sistema, indicando anche il semaforo mancante ed il suo 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) Indicate una semplice situazione in cui un processo può abbandonare *volonariamente* la CPU mentre è all'interno di una propria sezione critica.

Ad esempio se all'interno della sua sezione critica il processo esegue una system call wait su un semaforo con valore minore o uguale a zero.

c) Quali sono le proprietà che deve garantire una corretta soluzione al problema della sezione critica?

Progresso, attesa limitata e mutua esclusione.

d) quali problemi si possono verificare se le tre proprietà non sono rispettate?

Deadlock; starvation; due o più processi contemporaneamente in sezione critica.

e) Una soluzione al problema della sezione critica basata sul busy waiting è adatta per un sistema che usa un algoritmo di scheduling round robin (motivate la vostra risposta)?

No, un processo che cerca di entrare in una sezione critica occupata spreca inutilmente tutto il suo quanto di tempo senza riuscire ad entrare nella sezione critica (che verrà liberata solo quando il processo che la occupa riottiene l'uso della CPU).

ESERCIZIO 3 (9 punti)

a) In quale caso l'uso dell'allocazione indicizzata dello spazio su disco risulta particolarmente svantaggiosa in termini di spazio su disco sprecato?

Nel caso di file piccoli, poiché quasi tutto il blocco indice viene sprecato.

b) Utilizzando opportuni disegni descrivete le diverse forme di allocazione indicizzata viste a lezione: indicizzata *a schema concatenato*, indicizzata *a più livelli*, *variante adottata nei sistemi unix*.

Si vedano i lucidi della sezione 11.4.3

c) Se in un sistema unix i blocchi sono da 512 byte e il numero di un blocco è scritto su 4 byte, qual è la dimensione massima di un file memorizzabile nel sistema (è sufficiente riportare l'espressione da usare per il calcolo)

Un blocco da 512 byte può contenere 512/4 = 128 puntatori a blocco. Si ha quindi:
$(10 * 512) + (128 * 512) + (128^2 * 512) + (128^3 * 512)$ byte