

SISTEMI OPERATIVI – 8 giugno 2017
corso A nuovo ordinamento
e parte di teoria del vecchio ordinamento

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

1. Ricordate che non potete usare calcolatrici o materiale didattico, e che potete consegnare al massimo tre prove scritte per anno accademico.
2. Gli studenti a cui sono stati riconosciuti i 3 cfu di “linguaggio C” devono rispondere solo alle domande delle parti di teoria e di laboratorio Unix, e consegnare entro 1 ora e quindici minuti.
3. Gli studenti del vecchio ordinamento e di “*Istituzioni di Sistemi Operativi*” devono rispondere solo alle domande della parte di teoria, e devono consegnare entro 1 ora.

ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

ESERCIZIO 1 (5 punti)

- a) Si consideri il problema dei produttori e consumatori, con buffer limitato ad m elementi, dove i codici del generico produttore e del generico consumatore sono riportati qui di seguito. Inserite le opportune operazioni di wait e signal necessarie per il corretto funzionamento del sistema, indicando anche i semafori necessari ed il loro valore di inizializzazione (in grassetto sono indicate le parti mancanti da inserire).

semafori necessari con relativo valore di inizializzazione:

semaphore mutex = **1**;
semaphore full = **0**;
semaphore empty = **m**;

“consumatore”

repeat

wait(full)

wait(mutex)

<preleva dato dal buffer>

signal(mutex)

signal(empty)

<consumo dato>

forever

“produttore”

repeat

<produci dato>

wait(empty)

wait(mutex)

<inserisci dato nel buffer>

signal(mutex)

signal(full)

forever

- b) si supponga la presenza di $m+1$ produttori che cercano di inserire “contemporaneamente” nel buffer un elemento, mentre nessun consumatore ha ancora iniziato le sue operazioni. Cosa succede a ciascuno degli $m+1$ produttori?

L' $m+1$ -esimo produttore si addormenta sul semaforo *empty*, mentre i precedenti m produttori riescono a superarlo. Di questi, $m-1$ si addormentano su *mutex*, mentre un produttore lo supera e inserisce un elemento nel buffer.

c) (1 p.) Riportate, a vostra scelta, il diagramma di stato o il diagramma di accodamento di un sistema time sharing che implementa la memoria virtuale;

Per i diagrammi si vedano i lucidi della sezione 3.1.2 (p. 6), 3.2.1 (p. 18) e il lucido a p. 9 del cap. 9.

d) (1 p.) facendo riferimento al diagramma della domanda precedente, elencate le ragioni per cui un processo può passare dallo stato *waiting* allo stato *ready to run*

è stata completata una operazione di I/O; signal sul semaforo su cui il processo era addormentato; è arrivata in ram la pagina mancante che aveva causato un page fault del processo.

e) Perché il context switch tra peer thread è più veloce del context switch tra processi?

I peer thread condividono lo spazio di indirizzamento, che quindi non deve essere cambiato (ad esempio si continua ad usare la stessa page table) al context switch.

ESERCIZIO 2 (5 punti)

In un sistema che implementa la paginazione della memoria ma non la memoria virtuale, un indirizzo fisico è scritto su 40 bit, l'offset più grande all'interno di una pagina è pari a 3FF, e la tabella delle pagine più grande del sistema occupa 32 megabyte.

a) (2 p.) Quanto è grande lo spazio di indirizzamento logico del sistema (esplicitate i calcoli che fate)?

Lo spazio di indirizzamento fisico del sistema è diviso in $2^{40}/2^{10} = 2^{30}$ frame, Se decidiamo di usare 32 bit, ossia 4 byte per scrivere il numero di un frame, questa è la dimensione di una entry della tabella delle pagine del sistema. La tabella delle pagine più grande del sistema è grande 32 megabyte, ossia 2^{25} byte e quindi ha $2^{25}/2^2 = 2^{23}$ entry. Lo spazio di indirizzamento logico quindi è grande $2^{23} * 2^{10} = 2^{33}$ byte.

b) (2 p.) Se il sistema ha un tempo di accesso in RAM di 180 ns e adotta un TLB con un tempo di accesso di 60 ns, quale hit rate minimo deve garantire il TLB perché il sistema abbia un medium access time (mat) massimo di 280 ns? (esplicitate i calcoli che fate, e assumete per semplicità che l'accesso al TLB e alla tabella delle pagine in RAM avvengano in parallelo)

a partire dalla formula: $\text{mat} = \text{hit-rate} (240\text{ns}) + (1 - \text{hit-rate}) * (180\text{ns} + 180\text{ns}) \leq 280\text{ns}$ ricaviamo:

$240\text{hit-rate} + 360 - 360\text{hit-rate} \leq 280$; $80 \leq 120\text{hit-rate}$; $80/120 = 2/3$, da cui si ricava un hit-rate $\geq 66,6\%$

c) (1 p.) Descrivete (eventualmente ricorrendo ad opportuni schemi) il funzionamento di una Inverted Page Table.

Si vedano i lucidi della sezione 8.5.3

ESERCIZIO 3 (5 punti)

Un hard disk ha la capienza di 2^{36} byte, ed è formattato in blocchi da 4096 byte.

a) (1 p.) Quanti accessi al disco sono necessari per leggere l'ultimo blocco di un file A della dimensione di 2^{14} byte, assumendo che sia già in RAM il numero del primo blocco del file stesso e che venga adottata una allocazione concatenata dello spazio su disco? (motivate la vostra risposta)

5. Ogni blocco infatti memorizza 4093 byte di dati più 3 byte di puntatore al blocco successivo (infatti, $2^{36}/2^{12} = 2^{24}$), per cui sono necessari 5 blocchi per memorizzare l'intero file.

b) (1 p.) Se ora dovessimo inserire sull'hard disk un nuovo blocco di dati tra il secondo e il terzo blocco del file A, quante operazioni su disco dovremmo compiere? (assumete che siano già in RAM il numero del primo blocco del file A e il nuovo blocco da aggiungere).

4: lettura del primo blocco del file A per sapere il numero del secondo. Lettura del secondo blocco per sapere il numero del terzo blocco. Scrittura del nuovo blocco di dati sull'hard disk, riscrittura del secondo blocco che ora deve puntare al nuovo blocco appena inserito.

c) (1 p.) Quanto sarebbe grande, in megabyte, la FAT di questo sistema? (motivate numericamente la vostra risposta)

La FAT è un array con una entry per ciascun blocco dell'hard disk e che contiene il numero di un blocco, per cui: $2^{24} \times 3 \text{ byte} = 48 \text{ megabyte}$

d) (1 p.) Quali sono gli svantaggi nell'uso della FAT?

Per garantire un accesso efficiente ai file deve essere sempre tenuta in RAM occupando spazio, se viene persa si perdono tutte le informazioni sul file system, e deve quindi essere periodicamente salvata su disco.

e) (1 p.) Disegnate la FAT di un hard disk formato da 2^4 blocchi e contenente un unico file memorizzato, nell'ordine, nei blocchi 8, 4, 12, 2. Indicate anche dove è memorizzato il numero del primo blocco del file.

Si veda la figura 11.7 della sezione 11.4.2 dei lucidi.