SISTEMI OPERATIVI E LABORATORIO
Indirizzo Sistemi e Reti – 2 marzo 2010

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. Ricordate che se prendete un voto inferiore a 10 in una qualsiasi delle due parti dello scritto, dovete saltare il primo scritto della sessione di giugno/luglio.
3. 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.
4. Gli **studenti degli anni precedenti** devono sostenere l'intero scritto.
5. Si ricorda che a partire dallo sessione di giugno/luglio 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 della sessione di febbraio/marzo, dalla sessione di giugno/luglio deve comunque ridare l'intero scritto.

Ho svolto (solo per chi consegna alla fine della prima ora):
[ ] la parte relativa alla teoria. **[ ] la parte relativa al laboratorio UNIX**

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)

a) Riportate lo pseudocodice che descrive l'implementazione delle operazioni di Wait e Signal.

Si vedano i lucidi della sezione 6.5.2

b) Descrivete brevemente la differenza tra un algoritmo di scheduling preemptive e un algoritmo non-preemptive

Si vedano i lucidi della sezione 5.1.3

c) Descrivete brevemente un algoritmo di scheduling a code multiple con retroazione

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.

d) Riportate il diagramma di stato della vita di un processo.

Vedere il lucido della sezione 3.1.2.

e) rispetto al diagramma di stato della vita di un processo, e considerando un moderno sistema time sharing con memoria paginata che implementa la memoria virtuale, elencate brevemente le diverse ragioni per cui i processo può transire nella condizione di wait.

1. quando deve compiere una operazione di I/O
2. quando esegue una operazione di wait su un semaforo
3. quando il codice eseguito dal processo genera un page fault

ESERCIZIO 2 (9 punti)

In un sistema la memoria fisica è divisa in 2^{20} frame, un indirizzo logico è scritto su 31 bit, e all'interno di una pagina, l'offset massimo è 1FF.

a) Quanti frame occupa la la page table più grande del sistema? (motivate numericamente la vostra risposta)

Un frame/pagina è grande 2^9 = 512 byte, e quindi la page table più grande può avere $2^{(31-9)} = 2^{22}$ entry. Nel sistema vi sono 2^{20} frame, per cui sono necessari tre byte per scrivere il numero di un frame, e quindi la page table più grande occupa $(2^{22} \cdot 3) / 2^9$ frame = $2^{13} \cdot 3$ frame = 24K frame

b) Il sistema deve adottare una paginazione a più livelli (motivate la vostra risposta)?

Si perché la page table più grande non può essere memorizzata in un unico frame, e neanche in un numero ragionevolmente piccolo di frame adiacenti.

c) Quale dimensione minima dovrebbero avere le pagine di questo sistema per essere certi di non dover ricorrere ad una paginazione a più livelli? (motivate la vostra risposta, e per questa domanda assumete di usare 8 byte per scrivere il numero di un frame all'interno di una page table)

Poniamo 31 = m + n (m = bit usati per scrivere un numero di pagina, n = bit usati per scrivere l'offset). Allora il numero di entry della PT più grande, moltiplicato per la dimensione di una entry deve poter essere contenuto in una pagina/frame, ossia: $2^m \cdot 2^3 \leq 2^n$
Da cui: $m + 3 \leq n$. Poiché $m = 31 - n$; risolvendo il semplice sistema si ha $n = 17$, ossia le pagine devono almeno essere grandi 2^{17} = 128 Kbyte.

d) Se in un sistema non si vuole usare una paginazione a più livelli, quale soluzione alternativa si può adottare?

Una inverted page table

ESERCIZIO 3 (9 punti)

a) In un sistema i blocchi su disco occupano 1024 byte, e un puntatore a blocco è scritto su 4 byte. Il sistema operativo adotta una forma di allocazione indicizzata. Quanti accessi al disco sono necessari per leggere il byte numero 300.000, assumendo che gli attributi del file in questione siano già in memoria primaria? (motivate la vostra risposta, specificando le eventuali assunzioni che fate).

3. In un blocco indice possiamo scrivere 1024/4 = 256 puntatori, indirizzando così 256 Kbyte di dati, per cui un solo blocco indice non è sufficiente ad indirizzare il blocco contenente il byte specificato. Sia usando l'allocazione indicizzata gerarchica che l'allocazione indicizzata concatenata, un secondo blocco indice è sufficiente per indirizzare il blocco contenente il byte specificato. Sono quindi necessari due accessi al disco per leggere i due blocchi indice + un accesso per leggere il blocco contenente il byte numero 300.000.

b) Dite a cosa serve e descrivete brevemente la FAT.
(Se lo preferite, potete rispondere alla domanda con un esempio, disegnando la FAT relativa ad un hard disk da 16 blocchi che contiene un unico file memorizzato nei blocchi 9, 3, 5, 13)

E' un array con un numero di entry pari al numero di blocchi dell'hard disk, e permette un'implementazione efficiente dell'allocazione concatenata dei file: se X e Y sono due blocchi appartenenti al file F, con il blocco X che punta al blocco Y, allora nella entry X della FAT c'è scritto il numero Y. Se Z è l'ultimo blocco del file, nella entry Z della FAT viene scritto un marker di fine file. Se K è un blocco libero dell'hard disk, nella entry K della FAT viene scritto un marker di "blocco libero"

c) In un dato momento, lo spazio occupato sul disco da una cartella MYDIR è di X byte. Dopo un po' di tempo, lo spazio occupato dalla cartella MYDIR aumenta, ma la quantità totale di spazio occupato sul disco è diminuita. Come è possibile spiegare questa situazione? (si assuma per semplicità che vi sia un solo utente nel sistema, e che l'utente stia operando solo nella cartella MYDIR)

L'utente ha cancellato pochi file molto grossi da MYDIR, e ha inserito in MYDIR molti file piccoli la cui dimensione totale è inferiore alla dimensione dei file cancellati.