

**Sistemi Operativi 1 — A.A. 2004-2005, prova scritta del 20 settembre 2005**

Usa questa pagina per la brutta, staccala.

**Sistemi Operativi 1 — A.A. 2004-2005, prova scritta del 20 settembre 2005**

Usa questa pagina per la brutta, staccala.

Cognome \_\_\_\_\_ Nome \_\_\_\_\_ Matricola \_\_\_\_\_

Libri e appunti chiusi. Vietato comunicare con chiunque. Vietato l'uso di cellulari, calcolatrici, palmari e affini. Tempo a disposizione: 60 minuti.

Le domande sono etichettate con 1,2 o 3 asterischi:

- \* = domanda semplice, valutazione alta, **rispondi a queste prima delle altre**
- \*\* = domanda di media difficoltà
- \*\*\* = domanda difficile, valutazione bassa, **rispondi dopo aver risposto alle altre**

## Domanda 1

1. \* Cosa contiene l'inode nel filesystem di un sistema operativo UNIX? come è correlato l'inode al concetto di nome di file e di directory?

2. \* Mettiti nei panni del progettista di un software per gestire un filesystem. Supponi di voler utilizzare l'algoritmo di I/O scheduling C-SCAN e di voler ottimizzare la lettura di file molto lunghi. Quale è il modo migliore di sistemare i blocchi del file sul disco? in particolare quale è il modo migliore di sistemare i blocchi dei file che occupano più di una traccia?

## Domanda 2

1. \* Un processo accede alle pagine in memoria con la seguente sequenza: 1234521121332213. Mostra il working set per  $\Delta=4$  dopo ciascun accesso.

pagina acceduta	1	2	3	4	5	2	1	1	2	1	3	3	2	2	1	3
working set	1	2	3	4												
(l'ordine in cui disponi le pagine nelle caselle è irrilevante, scegli quello che più ti fa comodo)		1	2	3												
			1	2												
				1												

2. \*\* Descrivi l'approccio Page Fault Frequency e spiega perché tale approccio può essere considerato una approssimazione della tecnica del working set. Elenca pregi e difetti di tale approccio.

## Domanda 3

1. \* Supponi, in una architettura con 4 gigabyte di indirizzamento virtuale, che un processo occupi per lo stack gli indirizzi più alti e per il codice e i dati gli indirizzi più bassi. Quante page table entry servono se la pagina è di 4 kilobyte? Fai una assunzione per la grandezza della page table entry (giustificala) e calcola la grandezza della page table del processo.

2. \*\* Confronta l'architettura "tabella delle pagine a più livelli" con l'architettura "inverted page table" e mostra quali sono i pro e i contro dei due approcci.

## Domanda 4

1. \* Devi progettare lo scheduling di un sistema in cui sono presenti processi che interagiscono con l'utente e decidi per un approccio round robin. In base a quali considerazioni decidi la durata del quanto di tempo?

2. \*\* Supponi che la durata del quanto di tempo di un certo sistema round robin sia di 0.5 secondi. Sono presenti due processi, uno interattivo e l'altro cpu bound. Se il processo interattivo ha durata del cpu burst distribuita uniformemente (densità di probabilità uniforme) tra 0.1 e 0.6 secondi. Quale sarà la distribuzione (densità di probabilità) del tempo di risposta percepito dall'utente?

## Sistemi Operativi 1 — A.A. 2004-2005, prova scritta del 20 settembre 2005

3. \*\*\* Se il secondo processo invece di essere cpu bound avesse un cpu burst di 0,1sec con probabilità  $\frac{1}{2}$  e di 0,5sec con probabilità  $\frac{1}{2}$ , quale sarebbe la distribuzione del tempo di risposta percepito dall'utente del processo interattivo supponendo che l'altro sia ready (supponi indipendente la durata dei cpu burst dei due processi)?

### Domanda 5

1. \* Descrivi l'architettura dei sistemi operativi "process-based" (anche detta microkernel), descrivendo vantaggi e svantaggi rispetto a quella "execution within process" (stile UNIX).

2. \*\*Considera un sistema operativo di tipo "execution within process" con scheduling non-preemptive. Supponi che ciascun *processo utente* possa richiedere un servizio di I/O S mediante system call bloccante. Conta il numero di process switch e di mode switch complessivo nel caso in cui si abbiano  $p$  processi che fanno ciascuno solo  $r$  richieste al servizio S, che non ci siano altri processi nel sistema e ci sia sempre almeno un processo pronto quando viene effettuata la richiesta S. Giustifica la risposta.

3. \*\*\* Stessa situazione di prima ma con architettura a microkernel. Supponi che per eseguire S il sistema operativo debba attivare un *processo di sistema* il quale esegue altre  $n$  system call bloccanti che non richiedono l'attivazione di ulteriori processi di sistema. Il processo di sistema ha priorità maggiore, e interrompe i processi utente quando diviene *ready*. Quando ottiene la cpu serve tutte le richieste in coda. Calcola il numero massimo di process switch e di mode switch per l'esecuzione dei  $p$  processi. Supponi che ci sia sempre un processo pronto quando viene chiamata una system call bloccante. Giustifica la risposta.

