

Corso di Laurea in Informatica — Esame di Sistemi Operativi

29 novembre 2004

1. Si descrivano, brevemente, vantaggi e svantaggi della strutturazione a microkernel riguardo
 - a) progettazione e implementazione del sistema
 - b) manutenzione (aggiornamento) del sistema
 - c) prestazioni del sistema
3. Per ogni delle seguenti situazioni, indiate il meccanismo di comunicazione più adeguato per implementare un produttore-consumatore:
 - a) Thread
 - b) Processi locali “imparentati”
 - c) Processi *non* in esecuzione contemporanea

Risposta:

- a) (2pt) Progettazione e implementazione vengono facilitate, perché il sistema è più modulare, ed è più facile debuggare un kernel piccolo di uno pesante.
 - b) (2pt) Anche l’aggiornamento è facilitato dalla maggiore modularità, in quanto spesso si tratta di cambiare processi in user space, senza dover toccare il kernel.
 - c) (1pt) Tendenzialmente, un sistema a microkernel ha un overhead maggiore dei sistemi monolitici o stratificati.
2. Si consideri un sistema time-sharing con scheduling Shortest Remaining Time First, dove vengono sottoposti tre processi con i seguenti tempi di arrivo T , burst previsto τ , e burst reale t :
 4. Nei sistemi di calcolo odierni è normale che le risorse installate cambino nel tempo, e che i processi (o meglio, i loro programmatori) non sappiano specificare bene le risorse di cui avranno bisogno. In tale situazione, si dica quali delle seguenti strategie sono idonee a risolvere il problema dei deadlock, e perché:
 - a) Algoritmo del banchiere
 - b) Allocazione delle risorse secondo ordinamento prefissato
 - c) Quando l’uso di CPU scende sotto un certo livello, identificare i processi in stallo e terminarli automaticamente.

	T	τ	t
A	0	15	10
B	5	8	12
C	10	4	4

Trascurando la latenza del kernel si determini

- a) l’ordine di terminazione dei tre processi;
- b) il tempo di attesa medio;
- c) le nuove stime dei prossimi burst dei tre processi, con un fattore di media $\alpha = 0.5$.

Risposta: Si disegna il diagramma Gantt, facendo attenzione a lasciare B in esecuzione quando sfiora la sua previsione, e si vede che

- a) (2pt) l’ordine di terminazione è B C A;
- b) (3pt) il tempo di attesa di A, B, C è rispettivamente 7ms, 0ms, 16ms; la media è 7.66ms;

Risposta:

- a) (2pt) Memoria condivisa (globale) del processo, con semafori o monitor.
- b) (1pt) Pipe (o socket)
- c) (2pt) Code di messaggi (mailbox), o file.

Risposta:

- a) (2pt) No, perché diminuire le risorse disponibili al runtime non è safe.
 - b) (2pt) Sì: funziona indipendentemente dalle risorse e numero di processi.
 - c) (2pt) Sì, ma costoso per sistemi interattivi poco carichi. Richiede il mantenimento di un diagramma di allocazione risorse.
5. In un dato sistema con paginazione a un livello con memoria virtuale, la tabella delle pagine viene mantenuta in memoria principale, il cui tempo di accesso è $t = 5ns$. In media, si verifica un

page fault ogni milione di accessi alla memoria principale (per qualunque motivo); in tal caso, la pagina mancante viene caricata da disco su un frame prelevato dalla free list (che supponiamo non essere mai vuota) in $t_f = 10ms$.

- a) Si calcoli il tempo effettivo di accesso (comprensivo della traduzione) EAT_1 .
- b) Si decide di aggiungere un TLB, il cui hit rate è del 97% e con tempo di risposta $\epsilon = 1nsec$. Si calcoli il tempo effettivo di accesso EAT_2 .

Risposta:

- a) (3pt) Per ogni indirizzo, dobbiamo accedere due volte alla memoria (uno per la tabella delle pagine, uno per l'indirizzo fisico). Detto $\alpha = 10^{-6}$ la possibilità di page fault, il costo effettivo di ognuno di questi accessi è

$$EAT_0 = (1 - \alpha)t + \alpha(t_f + t) \\ = t + \alpha t_d = 5 + 10 = 15ns$$

quindi complessivamente abbiamo

$$EAT_1 = 2EAT_0 = 30ns.$$

- b) (3pt) Quando la TLB ha successo ossia nel 97% dei casi, si risparmia l'accesso in memoria per la tabella delle pagine. Quindi

$$EAT_2 = (1 + 0.03)EAT_0 + \epsilon \\ = 1.03 * 15 + 1 = 16.75ns$$

6. a) Come sono implementate le directory, in un usuale file system Unix come UFS o EXT2?
- b) Che conseguenze ha questo fatto sul costo di lookup e di creazione di un file nella dir?
- c) È possibile recuperare le due entry “.” e “..” se, accidentalmente, vengono cancellate?

Risposta:

- a) (2pt) Come lista di coppie nome-inode.
- b) (2pt) Sono operazioni lineari nella lunghezza della directory.
- c) (2pt) Le due entry rappresentano la directory stessa e la directory padre (che è unica). Se si cancellano, la directory è ancora raggiungibile dall'esterno, e quindi si possono ricostruire “da sopra”, ma non da dentro.

7. Nei sistemi Unix-like, i dispositivi sono accessibili come (speciali) file nel file system; ad esempio, il disco principale può essere `/dev/disk0`.

- a) Dal punto di vista della protezione, è questa una scelta conveniente?
- b) Come deve essere impostato il modo, l'utente e il gruppo di `/dev/disk0`?

Risposta:

- a) (2pt) Sì, perché così si possono riapplicare le politiche e i meccanismi standard dei file, senza dover inventare soluzioni ad hoc.
- b) (2pt) Ad esempio come segue

```
brw-r----- 1 root operator ... /dev/disk0
```

8. a) Si indichino almeno due differenze tra un sistema SMP e un sistema multicomputer.
- b) Quando conviene passare ad una architettura multicomputer invece di un SMP?
- c) Quale modello di IPC è più indicato per la programmazione di un multicomputer?

Risposta:

- a) (2pt) Le due architetture differiscono almeno in: configurazione di ogni nodo (sola CPU vs. nodo intero), locazione fisica (stesso rack/cabinet vs. stanza), sistema operativo (singolo vs. multiplo), struttura di connessione (memoria condivisa vs. rete).
- b) (2pt) Quando il carico di lavoro richiede una decina di CPU, essendo l'architettura SMP applicabile in pratica fino a 4-8 CPU.
- c) (2pt) Data l'assenza di memoria condivisa, conviene il passaggio di messaggi, possibilmente asincrono. Vanno bene le socket a datagrammi, per esempio.