

Esercizi e quesiti - Parte 2

I seguenti esercizi e quesiti sono proposti per verificare la preparazione e comprensione della materia. La loro validità è tanto maggiore quanto più sono svolti di pari passo con il programma del corso e in modo critico. È fondamentale che le risposte siano date in forma chiara e rigorosa, usando i concetti e la terminologia del corso, evitando ambiguità e spiegazioni fuori tema.

Si raccomanda, agli studenti interessati, di chiarire ogni dubbio e di verificare le risposte con il docente.

1) Spiegare tutte le azioni svolte in fase di: 1) compilazione, 2) creazione di un processo. Cercare di usare un livello di dettaglio adeguato per chiarire completamente l'argomento (entità in gioco, algoritmi, strutture dati, valutazioni quantitative, e quant'altro). Si faccia riferimento a:

- un programma di prova consistente nella modifica di ogni elemento di un array di un milione di interi secondo una funzione disponibile come procedura di 1000 istruzioni;
- un calcolatore con macchina assembler D-RISC, con memoria principale indirizzabile alla parola e capacità massima 32G parole;
- una gerarchia di memoria memoria virtuale – memoria principale con paginazione e pagine di dimensione 1K parole;
- un supporto a tempo di esecuzione dei processi con PCB di 1K parole e un'unica lista pronti con disciplina FIFO;
- un numero massimo di processi contemporaneamente attivi (creati e non ancora terminati) uguale a 200;
- una implementazione delle strutture dati condivise riferite indirettamente con il metodo degli indirizzi logici distinti.

Introdurre altre ipotesi o dati, se ritenuti necessari.

2) Pronunciarsi sulla verità o meno delle seguenti affermazioni:

- a) dato un processo con n strutture dati private e m condivise, per quanto riguarda l'allocazione di tali strutture il suo spazio di indirizzamento consta di $n + m$ puntatori alle strutture stesse;
- b) ogni volta che un processo modifica una struttura dati S , la modifica deve essere riportata anche nella propria memoria virtuale;
- c) ogni volta che un processo modifica una struttura dati X condivisa, la modifica deve essere riportata anche nelle memorie virtuali di tutti i processi che condividono X ;
- d) l'unità MMU deve necessariamente essere distinta dal processore;
- e) l'unità MMU implementa a livello firmware una funzionalità di sistema operativo;
- f) la Tabella di Rilocazione di un processo viene trasferita interamente all'interno della MMU all'atto della commutazione di contesto;
- g) la Tabella di Rilocazione di un processo viene trasferita parzialmente all'interno della MMU all'atto della commutazione di contesto.

3) Rivedere completamente l'Esercizio 2 del Cap. V, sez. 6.2, utilizzando tutte le possibili modalità di passaggio dei parametri alla procedura, a seconda che esso avvenga

- per valore o per riferimento,
- via memoria o via registri generali.

Valutare l'impatto che tali modalità hanno sul tempo di completamento del programma.

4) Compilare in D-RISC, mostrare la memoria virtuale, e valutare il tempo di completamento per la CPU del Cap. VII, dei seguenti programmi:

- a) prodotto matrice-vettore,
- b) prodotto di matrici,
- c) ricerca in un vettore,
- d) ricerca in una lista privata,
- e) ricerca in una lista condivisa.

5) Si confrontino i tempi di completamento della seguente espressione aritmetica:

$$y = a * b + c / d - e \% f - g$$

con tutti operandi in memoria, per i seguenti calcolatori:

- C1: D-RISC con architettura del Cap. VII;
- C2: DRISC + istruzioni aritmetiche con operandi in memoria e indirizzamento indiretto via registro;
- C3: D-RISC + una istruzione assembler primitiva esattamente corrispondente al calcolo dell'espressione stessa; tale istruzione è codificata su due parole e prevede indirizzi indiretti via registro.

Effettuare il confronto anche nel caso che si possa assumere trascurabile il tempo di accesso in memoria per la sola chiamata dell'istruzione (questa ipotesi verrà verificata su architetture studiate successivamente nel corso).

Trarre conseguenze dai risultati dei confronti richiesti.

6) Valutare il tempo di completamento del programma:

$$y = a * b + c / d - e \% f - g$$

con tutti operandi fisicamente allocati nella memoria locale di una unità di I/O operante con modalità Memory Mapped I/O. La macchina è D-RISC con architettura del Cap. VII.

7) Una unità di I/O, operante in Memory Mapped I/O e in DMA, implementa un co-processore matematico per l'esecuzione dell'operazione di radice quadrata. Scrivere un programma D-RISC che calcola la radice quadrata di tutti gli elementi di un array di interi e che sia in grado di sfruttare il co-processore.

8) Una unità di I/O, operante in Memory Mapped I/O, su richiesta del processore esegue l'incremento del contenuto di una locazione di memoria locale, avente l'indirizzo comunicato dal processore stesso. Dare un'implementazione dell'unità in modo tale che questa funzionalità sia eseguita in un tempo pari a non più di due accessi in memoria locale. Spiegare anche come il processore comunica la richiesta.

9) Modificare, rispetto a quanto visto nel Cap. IX, sez. 2, il trattamento delle interruzioni nella seguente ipotesi: le unità di I/O, tutte operanti in Memory Mapped I/O, hanno un funzionamento tale da non essere in grado di inviare il messaggio di interruzione dopo aver ricevuto l'ack dell'interruzione stessa. Rimane invece l'ipotesi che il segnale di interruzione non contiene informazioni aggiuntive che identifichino l'unità di I/O.