

Sistemi di Calcolo (A.A. 2017-2018)

Corso di Laurea in Ingegneria Informatica e Automatica
Sapienza Università di Roma

Esempio di compito per esonerati – Durata 1h 30’

Inserire nome, cognome e matricola nel file `studente.txt`.

Parte 1 (paginazione e memoria virtuale)

1. Si supponga di avere uno spazio logico di 1 GB suddiviso in pagine di 256 KB ed uno spazio fisico da 4 GB. Quanti bit servono per rappresentare il numero di frame in una entry della tabella delle pagine? E quante entry conterrà la tabella?
2. Si supponga di avere una tabella delle pagine con celle a 32 bit che occupa 64 MB. Se il sistema di memoria usa pagine da 8 KB, quanto è grande lo spazio virtuale?
3. Si consideri il seguente programma C eseguito su una piattaforma con memoria virtuale gestita con pagine da 4 KB:

```
double x[1500];
int main() {
    short y[10000];
    double* z = malloc(400*sizeof(double));
    return 0;
}
```

Quante pagine occupano, approssimativamente, le sezioni DATA, HEAP e STACK del programma subito prima dell’istruzione `return`? Si assuma che `short` occupi 2 byte, `int` 4 byte e `double` 8 byte, e che la variabile `z` sia mantenuta in un registro.

Inserire le risposte nel file `es1.txt` mostrando di volta in volta i calcoli effettuati.

Parte 2 (programmazione POSIX)

Si scriva un programma `chain` che prende da riga di comando un numero `k` e il percorso di un file eseguibile seguito dai suoi argomenti, lancia `k` istanze di quell’eseguibile e fa in modo che, prima di lanciare l’eseguibile, ogni processo uccida il processo creato immediatamente prima. Il programma deve restituire la somma degli stati di terminazione dei processi eseguiti. Se il programma viene lanciato con meno di 3 argomenti, stampare su `stderr` il testo: “usage: chain <num-instances> <exec-pathname> ...”. Esempio di uso:

```
$ ./chain 3 ls /pippo
ls: ls: cannot access /pippo: No such file or directorycannot access
/pippo
: No such file or directory
$ echo $?
4
$
```

Si noti come nell’esempio sopra i messaggi stampati dai vari processi si mescolino per via della schedulazione time-sharing dei processi.

Suggerimento: si può fare ricorso alla funzione `atoi()` per estrarre un `int` da una stringa. Usare il comando `man` per consultare la documentazione delle system call POSIX.

Parte 3 (analisi località)

Sia dato il seguente frammento di codice:

```
void blur(const int* in, int* out, int n, int d) {
    int i, j;
    for (i=0; i<d; ++i)
        out[i] = in[i];
    for (i=d; i<n-d; ++i) {
        int sum = 0;
        for (j=i-d; j<i+1+d; ++j) sum += in[j];
        out[i] = sum/(d+1+d);
    }
    for (i=n-d; i<n; ++i)
        out[i] = in[i];
}
```

Si assuma di eseguire `blur` in un sistema di calcolo con una cache completamente associativa con almeno tre linee da 64 byte ciascuna. Si assuma inoltre che tutte le variabili siano mantenute nei registri, che `in` e `out` siano allineati a 64 byte e che `n=1024` e `d=3`.

- Quante richieste di accesso a memoria vengono effettuate dal programma?
- Quanti cache miss si hanno?
- Cambierebbe qualcosa se le linee di cache fossero più grandi?

Inserire le risposte all'interno del file `es3A.txt`.

Parte 4 (quiz)

Si risponda ai seguenti quiz, inserendo le risposte (A, B, C, D o E per ogni domanda) nel file `es4-risposte.txt`. **Una sola risposta è quella giusta**. Rispondere E equivale a non rispondere (0 punti).

Domanda 1 (memory footprint)

Quanti byte in meno occuperebbe la seguente struttura se la ottimizzassimo? Assumere una piattaforma a 32 bit.

```
struct S {
    char a;
    void* b;
    short c;
    int d;
    char e;
};
```

A	0	B	4
C	8	D	12

Domanda 2 (esecuzione concorrente)

Quali dei seguenti stati non viene mai attraversato durante un context switch tra processi?

A	Blocked	B	Running user
C	Running kernel	D	Ready

Domanda 3 (flusso di controllo eccezionale)

Una delle seguenti affermazioni è **falsa**, quale?

A	L'esecuzione di un interrupt handler richiede che il PCB del processo corrente sia salvato dal sistema operativo e poi ripristinato	B	I timer di sistema permettono di generare interrupt periodici, ad esempio per consentire a un sistema operativo di effettuare un context switch tra processi per realizzare uno scheduling time-sharing
C	Quando si preme CTRL-C sulla tastiera viene generato un interrupt	D	Trap e interrupt consentono di entrare in modalità di esecuzione kernel

Domanda 4 (architetture di calcolo)

Quale dei seguenti fattori ostacola il raggiungimento di un instruction-level parallelism elevato?

A	Numero limitato di hazard dati	B	Branch predictor efficiente
C	Uso di istruzioni branchless	D	Creazione di "bolle" nella pipeline