Politecnico di Milano – Sede di Cremona
Anno Accademico 2018/2019

Architettura dei Calcolatori e Sistemi Operativi

Esame – 24.1.2019

Prof. Carlo Brandolese

Cognome _____ **Nome** _____

Matricola _____ **Firma** _____

Istruzioni

1. Scrivere con cura, negli spazi sopra segnati, il proprio cognome, nome, numero di matricola e apporre la firma.
2. È vietato consultare libri, eserciziari, appunti ed utilizzare la calcolatrice e qualunque strumento elettronico (inclusi i cellulari), pena l'invalidazione del compito.
3. Il testo, debitamente compilato, deve essere riconsegnato in ogni caso.
4. Il tempo della prova è di 3 ore

Valutazione

Domanda	Voto	Note
A		
B		
C		
D		
E		
F		

Domanda A

Si consideri la seguente per la ricerca binaria del valore **value** nell'array **data** di dimensione nota e pari a **size**:

```
int bsearch( int* data, int size, int value )
{
    int lo = 0;
    int hi = size-1;
    int md;
    while( lo < hi ) {
       md = (lo + hi) / 2;
       if( data[md] == value ) {
           return md;
       } else if( data[md] > value ) {
         hi = md - 1;
       } else {
         lo = md + 1;
    }
    return -1;
}
```

Si traduca la funzione in assembly MIPS

Domanda B

Sia dato un processore MIPS dotato di una pipeline standard a CINQUE STADI, senza propagazione né riconoscimento dei conflitti. Sia inoltre dato il seguente codice assembly.

```
1     ADDI  R1, R2, 8
2     SW    R1, 12(R2)
3     BEQ   R5, R4, ETICHETTA # Si supponga R5!=R4
4     ADDI  R4, R3, 1
5     SLT   R5, R6, R4
```

Inserire le istruzioni NOP necessarie per assicurare la corretta esecuzione del codice.

Risolvere il problema utilizzando le istruzioni NOP necessarie, ma tenendo presente che al processore è stata aggiunta l'ottimizzazione dei percorsi di propagazione.

Risolvere infine il problema supponendo che il processore sia dotato di PERCORSI DI PROPAGAZIONE e PREDIZIONE STATICA BRANCH NOT TAKEN. Indicare inoltre il numero di cicli di clock totali e il CPI.

N° Istruzione	**Cicli di Clock**														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Numero cicli di clock:

CPI:

Domanda C

Si consideri un sistema con uno spazio di indirizzamento di 8GByte ed due memorie cache (una cache dati ed una cache istruzioni) con le caratteristiche seguenti:

	Cache Dati	Cache Istruzioni
Associatività	Set associativa a 4 vie	Diretta
Dimensione totale	128 KB	32 KB
Dimensione linea	128 B (per ogni set)	128 B
Tempo di accesso	2 ns	1 ns

Si indichi la struttura dell'indirizzo visto dalle cache, descrivendo i vari campi e il loro significato.

Struttura dell'indirizzo cache Dati

Struttura dell'indirizzo cache Istruzioni

Sapendo che:

- L'accesso alla memoria RAM avviene a quad-words di 128 bit
- Il tempo di accesso alla RAM in modalità normale è di 40 ns
- Il tempo di accesso alla RAM in modalità burst è
    - 100 ns per la prima parola
    - 10 ns per le parole successive
- L'hit rate della memoria dati è pari al 98%
- L'hit rate della memoria istruzioni e pari al 99.5%

Per ognuna delle due cache, si calcoli il tempo di accesso medio alla memoria

Tempo medio di accesso D-Cache

Tempo medio di accesso I-Cache

Infine, sapendo che in un dato programma il 40% delle istruzioni accedono alla memoria, si calcoli il tempo medio di esecuzione di una istruzione per il programma in esame.

Tempo di esecuzione:

Domanda D

Si consideri un calcolatore dotato di sistema operativo Linux dotato di un filesystems con le seguenti caratteristiche:

- Le dimensioni dei blocchi sono di 4096 byte
- Per l'apertura dei file è sempre necessario accedere a:
  - I-node di ogni cartella o file presente nel percorso
  - Blocco per il contenuto di ogni cartella presente nel percorso
  - Primo blocco dati del file

Dato il contenuto del seguente volume:

I-lista: <0,dir,0> <1,dir,1> <2,dir,2> <3,dir,3> <4,dir,4> <5,norm,{100,…,132}> <6,norm,{200,…,208}> <7,norm,{300,…,353}> <8,dir,8> <9,dir,9> <10,dir,10> <20,norm,{400,401,402}> <21,norm,{500}> …
Blocco 0: … <1,bin> <2,home> <3,usr> …
Blocco 1: … <6, cat> <7, grep> …
Blocco 2: … <8, pippo> <9,pluto> …
Blocco 3: … <4, bin> …
Blocco 4: … <5,ls> …
Blocco 8: … <11,.bashrc> …
Blocco 9: … <10,data> …
Blocco 10: … <20, exam.docx> <21, solutions.docx> …

Per ciascuna delle chiamate di sistema sotto riportate, si indichi la sequenza di accessi agli I-Node e ai blocchi (del tipo: I-Node X oppure Blocco Y).

Chiamata di sistema	**Sequenza di accessi**

`fd = open(`
`  "/home/pluto/exam.docx",`
`  O_RDWR`
`);`	

Chiamata di sistema	Sequenza di accessi
`fd2 = open( "/home/pluto/solutions.docx", O_RDWR | O_CREAT, S_IRUSR );`	
`n = read( fd, buffer, 555 )`	

Assumiamo ora di creare un nuovo file di dimensione 11KB con la seguente istruzione:

```
fd3 = open(
  "/home/pippo/results.xlsx",
  O_RDWR | O_CREAT, S_IRUSR | S_IRGRP | S_IROTH
);
```

Riportare qui sotto le modifiche effettuate al contenuto del volume dopo la creazione del file stesso.

Domanda E

Si scriva un programma cha accetta due parametri da linea di comando:

```
$> squares <N> <S>
```

Il programma:

- Calcola in parallelo su N thread quali numeri da 1 a S sono quadrati perfetti
- Stampa a video in ordine crescente i numeri primi

Alcuni controlli opportuni sugli ingressi sono:

- $N \geq 1$
- $1 \leq S \leq 10000$

Secondo le specifiche sopra riportate la seguente invocazione:

```
$> squares 3 50
```

istanzia 3 thread concorrenti per identificare i quadrati perfetti da 1 a 50 e produce come output a video:

```
1 4 9 16 25 36 49
```

Domanda F

Si disegni e si descriva accuratamente il diagramma di transizione degli stati di un processo senza considerare il meccanismo dello swapping.