



**Politecnico di Milano – Sede di Cremona**  
**Anno Accademico 2017/2018**

**Architettura dei Calcolatori e Sistemi Operativi**

**Esame – 19.02.2018**

**Prof. Carlo Brandoles**

---

**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**

<b>Domanda</b>	<b>Voto</b>	<b>Note</b>
A		
B		
C		
D		
E		
F		

## Domanda A

---

Si consideri la funzione:

```
int strrchr( char* str, char c )
```

che restituisce la posizione dell'ultima occorrenza del carattere **c** nella stringa **str**. Qualora il carattere non fosse presente nella stringa, la funzione restituisce il valore -1. Il codice seguente mostra un utilizzo della funzione da sviluppare:

```
.data
str: .asciiz "Prova di stringa di esempio"
chr: .ascii "r"

.text
main:
    la    $a0,str
    lb    $a1,chr
    jal   strrchr
    # Print
    move  $a0, $v0
    li    $v0,1
    syscall
```

Si traduca la funzione **strrchr()** in assembly.

## Domanda B

Si consideri un processore MIPS dotato di pipeline standard a cinque stadi, **senza alcuna ottimizzazione**. Sia dato il seguente codice assembly:

```
1 SUB R1, R2, R3
2 LW  R2, 200(R1)
3 SUB R3, R4, R5
4 ADD R4, R3, R4
```

Si indichino le dipendenze dati e di controllo che possono causare conflitti.

Si esegua il codice inserendo stalli ove necessario, e si calcoli il numero di cicli di clock necessari per eseguire il codice, e il CPI.

Istruzione	1	2	3	4	5	6	7	8	9	0	1	1	1	1	1	1	1	1	2

Numero cicli di clock	
CPI	

Si riordino le istruzioni in modo da minimizzare il numero di stalli. Quindi si mostri l'esecuzione del codice modificato e si calcoli il numero di cicli di clock necessari per l'esecuzione e il CPI.

Istruzione	1	2	3	4	5	6	7	8	9	0	1	1	1	1	1	1	1	1	2

Numero cicli di clock	
CPI	

## Domanda C

Si consideri un sistema con uno spazio di indirizzamento di 1 GByte, una cache dati ed una cache istruzioni. Le caratteristiche delle due cache sono le seguenti:

	<b>Cache Istruzioni</b>	<b>Cache Dati</b>
<b>Associatività</b>	Mapping diretto	Set associativa a 4 vie
<b>Dimensione totale</b>	64 KB	128 KB
<b>Dimensione linea</b>	256 B	1 KB (per ogni set)
<b>Tempo di accesso</b>	1 ns	2 ns
<b>Hit rate</b>	99%	99%

Sulla base di queste informazioni si indichi la struttura dell'indirizzo visto dalle cache, descrivendo i vari campi e il loro significato.

Cache Istruzioni

Cache Dati

Sapendo che:

- L'accesso alla memoria RAM avviene a parole di 32 bit
- Il tempo di accesso alla RAM in modalità normale è di 50 ns
- Il tempo di accesso alla RAM in modalità burst è
  - 100 ns per la prima parola
  - 10 ns per le parole successive

Si calcoli il tempo di accesso medio alla memoria per le due cache

Cache Istruzioni

Cache Dati

Considerando un programma in cui il 50% delle istruzioni sono di tipo load/store, si calcoli il tempo medio di esecuzione di una istruzione con e senza cache ed il miglioramento delle prestazioni ottenuto introducendo le due cache.

Tempo medio per istruzione senza cache

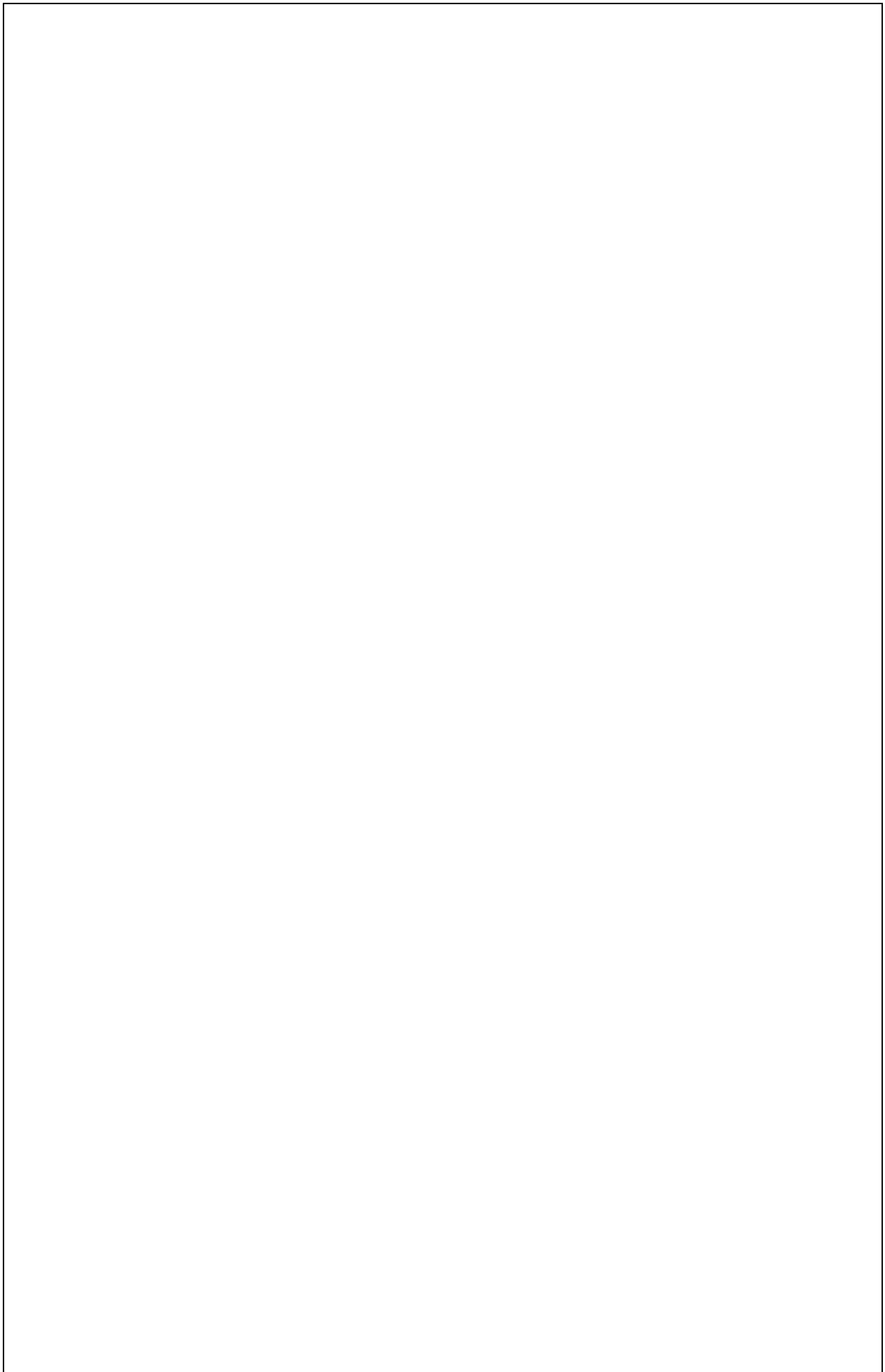
Tempo medio per istruzione con cache

Miglioramento delle prestazioni

## **Domanda D**

---

Un programma deve cercare uno specifico valore in un array di  $N$  interi, con  $N$  molto grande, dell'ordine del milione e restituire 1 se il valore è stato trovato, 0 in caso contrario. Il programma dovrà essere eseguito su una macchina dotata di  $K$  microprocessori e dovrà essere strutturato in modo tale da sfruttare il più possibile il parallelismo disponibile. Si considerino  $N$  e  $K$  costanti (definite mediante macro), l'array di interi già disponibile in memoria (come variabile globale) e si realizzi un programma efficiente sfruttando la programmazione multiprocesso.



## Domanda E

Si consideri un sistema dotato di memoria virtuale con paginazione e segmentazione di tipo UNIX caratterizzato dai parametri seguenti:

Memoria logica: 64 Kbyte  
Memoria fisica: 32 Kbyte  
Dimensione pagina: 4 Kbyte

Si definisca la struttura degli indirizzi fisico e logico indicando la lunghezza dei campi NPF, Spiazzamento fisico, NPL, Spiazzamento logico.

Indirizzo fisico:

Indirizzo logico:

Nel sistema vengono creati alcuni processi, indicati nel seguito con P, Q, R. I programmi eseguiti da tali processi sono due: X e Y. La dimensione iniziale dei segmenti dei programmi è la seguente:

CX: 12K                  DX: 12K                  PX: 4K  
CY: 16K                  DY: 4K                  PY: 4K

Completare la seguente tabella, riportando la struttura in pagine della memoria virtuale dei due programmi X e Y sapendo che le pagine vengono allocate sequenzialmente.

Pagina Virtuale	Programma X	Programma Y
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
A		
B		
C		
D		
E		
F		

Inizia quindi l'esecuzione e ad un certo istante  $T$  le seguenti operazioni sono state completate:

- Il processo P viene creato ed esegue il programma Y
  - Il processo P legge un dato all'indirizzo 0x4E16
  - Il processo Q viene creato ed esegue il programma X
  - Lo stack del processo Q cresce e richiede una nuova pagina
  - Il processo Q salta all'indirizzo 0x16BA
  - Il processo P genera un figlio R
  - Il processo R scrive un dato all'indirizzo 0x4AAC
  - Il processo P termina

Sapendo che:

1. L'esecuzione di un programma avviene caricando inizialmente e in quest'ordine:
    - a. La pagina di codice con l'istruzione di partenza
    - b. Una pagina di pila
  2. Il caricamento di ulteriori pagine in memoria avviene su richiesta ( on demand )
  3. Il numero di pagine residenti in memoria fisica per ogni processo ( R ) è pari a 3
  4. L'indirizzo esadecimale di partenza di X è 0x0A35
  5. L'indirizzo esadecimale di partenza di Y è 0x25E1
  6. Per la sostituzione delle pagine di memoria si utilizza una politica FIFO. Almeno una pagina di pila deve sempre rimanere in memoria. La politica si usa sia al raggiungimento del massimo numero di pagine residenti per processo, che nel caso di riempimento della memoria fisica.
  7. L'allocazione delle pagine virtuali nelle pagine fisiche avviene sempre in sequenza.
  8. All'inizio della sequenza di eventi la MMU è vuota

Si riporti nelle tabelle seguenti una descrizione dello stato della memoria fisica e della MMU al termine della sequenza di operazioni di cui sopra.

Memoria Fisica	
Indirizzo Fisico	Pagine Allocate
0	
1	
2	
3	
4	
5	
6	
7	

## **Domanda F**

---

Facendo riferimento al sistema operativo Linux, si descrivano i concetti di libreria statica e di libreria dinamica, indicando in particolare le differenze tra le modalità del loro impiego in un programma.