

## Sistemi di Calcolo (A.A. 2014-2015)

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

### Soluzioni esercizi riepilogativi sulla seconda parte del Modulo I – Sistemi di memoria

---

#### Domanda 1

Si consideri un'operazione `movl x, %eax`. L'istruzione richiede una transazione di accesso a memoria che effettua una serie di passi che scambiano dati fra la CPU e la memoria utilizzando il bus della memoria. Quali?

#### Soluzione:

1. La CPU mette l'indirizzo `x` viene sul bus della memoria;
  2. La memoria centrale preleva `x` dal bus della memoria, recupera il dato memorizzato a quell'indirizzo e lo pone sul bus della memoria;
  3. La CPU legge il dato dal bus della memoria e lo copia nel registro `eax`.
- 

#### Domanda 2

Qual è il tempo tipico di accesso (latenza) a una memoria DRAM?

<b>A</b>	1-10 nanosecondi	<b>B</b>	10-100 nanosecondi
<b>C</b>	1-10 microsecondi	<b>D</b>	1-10 millisecondi

#### Soluzione:

B. 10-100 nanosecondi.

---

#### Domanda 3

Qual è il tempo tipico di accesso (latenza) a un disco (tradizionale, non a stato solido)?

<b>A</b>	1-10 nanosecondi	<b>B</b>	10-100 nanosecondi
<b>C</b>	1-10 microsecondi	<b>D</b>	1-10 millisecondi

#### Soluzione:

D. 1-10 millisecondi.

---

#### Domanda 4

Una memoria DRAM non conserva i dati in essa immagazzinati in assenza di alimentazione elettrica. E' richiesto un refresh elettrico:

<b>A</b>	Ogni 10-100 nanosecondi	<b>B</b>	Ogni 10-100 microsecondi
<b>C</b>	Ogni 10-100 millisecondi	<b>D</b>	Ogni secondo

#### Soluzione:

C. Ogni 10-100 millisecondi

---

#### Domanda 5

Grazie ai progressi tecnologici, la velocità di accesso ai dispositivi di memorizzazione basati su dischi magnetici si è costantemente ridotta durante l'ultimo decennio:

<b>A</b>	Vero: i tempi di accesso a disco si sono dimezzati ogni 18 mesi come predetto dalla Legge di Moore	<b>B</b>	Falso: i tempi di accesso a disco sono rimasti quasi invariati negli ultimi anni.
----------	--	----------	---

**Soluzione:**

B. Falso: i tempi di accesso a disco sono rimasti quasi invariati negli ultimi anni.

---

**Domanda 6\*<sup>1</sup>**

L'istruzione `movl (%eax), %ecx` richiede sempre lo stesso tempo per essere completata indipendentemente dal contesto in cui viene richiamata.

<b>A</b>	Vero	<b>B</b>	Falso
----------	------	----------	-------

**Soluzione:**

B. Falso! I tempi di accesso a una memoria gerarchica possono variare fino a 6 ordini di grandezza: da pochi **nanosecondi** ( $10^{-9}$  secondi) se il dato da accedere è nella cache L1 fino a diversi **millisecondi** ( $10^{-3}$  secondi) se l'accesso genera un page fault poiché il dato è su disco.

---

**Domanda 7**

Una memoria cache è una memoria:

<b>A</b>	Veloce e costosa	<b>B</b>	Veloce ed economica
<b>C</b>	Lenta e costosa	<b>D</b>	Lenta ed economica

**Soluzione:**

A. Veloce e costosa. La dimensione delle cache in una memoria gerarchica è inversamente proporzionale al loro costo: sono tanto più veloci, piccole e costose tanto più vicine sono alla CPU. Sono tanto più efficaci tanto maggiore è la località negli accessi a memoria di un programma.

---

**Domanda 8**

Lo scopo di una memoria cache è:

<b>A</b>	Garantire protezione della memoria in modo che un processo non possa accedere allo spazio logico di un altro processo	<b>B</b>	Fornire uno spazio di memoria condiviso tra più processi per consentire loro di comunicare
<b>C</b>	Sfruttare le proprietà di località dei programmi per velocizzare gli accessi a memoria	<b>D</b>	Permettere a più core di accedere in parallelo alla stessa zona di memoria fisica

**Soluzione:**

C. Sfruttare le proprietà di località dei programmi per velocizzare gli accessi a memoria.

---

**Domanda 9**

Un programma esibisce località spaziale nell'accesso alla memoria in uno di questi casi:

<b>A</b>	Se una cella di memoria viene acceduta, è probabile che altre celle di memoria nel suo vicinato vengano accedute a breve	<b>B</b>	Se una cella di memoria viene acceduta, è probabile che la stessa cella venga acceduta ancora a breve
----------	--	----------	---

---

<sup>1</sup> Le risposte alle domande marcate con \* devono essere motivate.

**Soluzione:**

A. Se una cella di memoria viene acceduta, è probabile che altre celle di memoria nel suo vicinato vengano accedute a breve.

**Domanda 10**

Le dimensioni attuali delle cache sono dell'ordine:

<b>A</b>	GB	<b>B</b>	KB-MB
----------	----	----------	-------

**Soluzione:**

B. KB-MB. Ad esempio, in un Intel Core i7 si hanno le seguenti dimensioni: 1) cache L1: 32KB dati + 32KB istruzioni; 2) cache L2: 256KB per istruzioni e dati; 3) cache L3: 8MB per istruzioni e dati.

**Domanda 11**

In una cache completamente associativa:

<b>A</b>	Ogni blocco di memoria può essere ospitato in qualsiasi blocco di cache	<b>B</b>	Ogni blocco di memoria ha un solo possibile blocco di cache che può ospitarlo
<b>C</b>	Ogni blocco di memoria ha diversi possibili blocchi di cache che possono ospitarlo, ma non tutti	<b>D</b>	Nessuna delle precedenti

**Soluzione:**

A. Ogni blocco di memoria può essere ospitato in qualsiasi blocco di cache.

**Domanda 12**

Si ha un cache hit quando:

<b>A</b>	Il processore accede a un indirizzo di memoria il cui contenuto non è stato precedentemente caricato in cache	<b>B</b>	Il processore accede a un indirizzo di memoria il cui contenuto è stato precedentemente caricato in cache
----------	---	----------	---

**Soluzione:**

B. Il processore accede a un indirizzo di memoria il cui contenuto è stato precedentemente caricato in cache.

**Domanda 13\***

Un cache miss di tipo "conflict" può aversi in una cache completamente associativa?

<b>A</b>	Sì	<b>B</b>	No
----------	----	----------	----

**Soluzione:**

B. No. In una cache completamente associativa non vi sono vincoli su quale blocco di cache può ospitare un qualunque blocco di memoria.

**Domanda 14**

Un cache miss di tipo "cold (compulsory)" si ha quando:

<b>A</b>	La cache avrebbe blocchi liberi per ospitare il blocco acceduto, ma per vincoli architetturali non sono utilizzabili per quel particolare blocco	<b>B</b>	Un determinato blocco di memoria viene caricato per la prima volta in cache
----------	--	----------	---

**Soluzione:**

B. Un determinato blocco di memoria viene caricato per la prima volta in cache.

---

**Domanda 15\***

Si consideri un sistema con una piccola cache completamente associativa contenente 2 sole linee da 16 byte ciascuna. Quanti cache miss vengono generati dal seguente frammento di programma? Assumere che l'array  $v$  sia allineato a un indirizzo multiplo di 16 byte e che la cache inizialmente non contenga alcun blocco di memoria in uso al processo.

```
int v[12];
v[0]=5;
v[1]=4;
v[10]=3;
v[8]=2;
v[2]=v[10];
v[9]=v[1];
```

<b>A</b>	1	<b>B</b>	2
<b>C</b>	5	<b>D</b>	8

**Soluzione:**

B. 2. Poiché l'array è allineato a un indirizzo multiplo di 16, può essere partizionato in tre blocchi da 16 byte, ciascuno dei quali può essere portato in cache come un unico pacco:

- blocco 0:  $v[0]$ ,  $v[1]$ ,  $v[2]$ ,  $v[3]$
- blocco 1:  $v[4]$ ,  $v[5]$ ,  $v[6]$ ,  $v[7]$
- blocco 2:  $v[8]$ ,  $v[9]$ ,  $v[10]$ ,  $v[11]$

Poiché la cache è completamente associativa, ogni blocco dell'array può essere portato in uno qualsiasi dei due blocchi di cache. La sequenza di accessi a memoria è:

- scrive  $v[0]$  (blocco 0): **miss** (cold) – blocco 0 portato nel primo blocco cache
- scrive  $v[1]$  (blocco 0): hit
- scrive  $v[10]$  (blocco 2): **miss** (cold) – blocco 2 portato nel secondo blocco cache
- scrive  $v[8]$  (blocco 2): hit
- legge  $v[10]$  (blocco 2): hit
- scrive  $v[2]$  (blocco 0): hit
- legge  $v[1]$  (blocco 0): hit
- scrive  $v[9]$  (blocco 2): hit

---

**Domanda 16\***

Con riferimento alla domanda 15, il programma esibisce qualche tipo di località?

<b>A</b>	No	<b>B</b>	Sì, solo spaziale
<b>C</b>	Sì, solo temporale	<b>D</b>	Sì, sia spaziale che temporale

**Soluzione:**

D. Sì, sia spaziale che temporale. Si ha località spaziale perché ad esempio dopo l'accesso a  $v[0]$  si accede a breve a  $v[1]$  e  $v[2]$ . Si ha località temporale perché ad esempio dopo l'accesso a  $v[10]$  lo si riaccede di nuovo a breve.

---

**Domanda 17\***

Con riferimento alla domanda 15, quanti cache miss si avrebbero se la cache fosse a indirizzamento diretto?

<b>A</b>	1	<b>B</b>	2
----------	---	----------	---

<b>C</b>	4	<b>D</b>	8
----------	---	----------	---

**Soluzione:**

C. 4. Poiché la cache è a indirizzamento diretto e ha due soli blocchi, si hanno questi vincoli:

- blocco 0 array può andare solo in blocco 0 cache
- blocco 1 array può andare solo in blocco 1 cache
- blocco 2 array può andare solo in blocco 0 cache

La sequenza di accessi a memoria è:

- scrive  $v[0]$  (blocco 0): **miss** (cold) – blocco 0 portato nel blocco 0 cache
- scrive  $v[1]$  (blocco 0): hit
- scrive  $v[10]$  (blocco 2): **miss** (cold) – blocco 2 portato nel blocco 0 cache
- scrive  $v[8]$  (blocco 2): hit
- legge  $v[10]$  (blocco 2): hit
- scrive  $v[2]$  (blocco 0): **miss** (conflict) – blocco 0 portato nel blocco 0 cache
- legge  $v[1]$  (blocco 0): hit
- scrive  $v[9]$  (blocco 2): **miss** (conflict) – blocco 2 portato nel blocco 0 cache

**Domanda 18\***

Con riferimento alla domanda 15, i cache miss sono tutti “cold”?

<b>A</b>	Sì	<b>B</b>	No
----------	----	----------	----

**Soluzione:**

A. Sì. Vedi soluzione della domanda 15.

**Domanda 19\***

Con riferimento alla domanda 17, vi sono cache miss di tipo “conflict”?

<b>A</b>	Sì	<b>B</b>	No
----------	----	----------	----

**Soluzione:**

A. Sì. Vedi soluzione della domanda 17.

**Domanda 20**

In una gerarchia di memoria, le velocità di accesso sono tali che:

<b>A</b>	L1 > L2 > L3 > registri < memoria centrale < disco	<b>B</b>	Registri < L3 < L2 < L1 < memoria centrale < disco
<b>C</b>	L3 < L2 < L1 < registri < memoria centrale < disco	<b>D</b>	Registri < L1 < L2 < L3 < memoria centrale < disco

**Soluzione:**

D. Registri < L1 < L2 < L3 < memoria centrale < disco.

**Domanda 21\***

Di quanto è più lenta una mov  $M,R$  (memoria-registro) che accede a un valore non in cache rispetto a una mov  $R,R$  (registro-registro)?

<b>A</b>	$\sim 2x$	<b>B</b>	$> 30x$
<b>C</b>	$\sim 5x$	<b>D</b>	$\sim 10x$

**Soluzione:**

B.  $> 30x$ . Il tempo tipico di una mov da registro a registro è inferiore a 1 ns mentre un accesso

a un dato di memoria non in cache richiede almeno 30 ns.

---

**Domanda 22\***

Quanti cache miss genera la scansione lineare di un array di  $n$  elementi assumendo una cache in cui ogni linea può contenere  $k$  elementi dell'array?

<b>A</b>	$k$	<b>B</b>	$n/k$
<b>C</b>	$n$	<b>D</b>	1

**Soluzione:**

B.  $n/k$ . Accedendo sequenzialmente, si ha località spaziale per cui dopo ogni cache miss si avranno  $k-1$  hit essendo i  $k-1$  elementi successivi parte dello stesso blocco di memoria appena caricato in cache. Si ha quindi un miss ogni  $k$  accessi agli elementi dell'array, quindi  $n/k$  miss totali.

---

**Domanda 23\***

Quanti cache miss può generare nel caso peggiore la scansione lineare di una lista collegata di  $n$  elementi assumendo una cache completamente associativa in cui ogni linea può contenere  $k$  nodi dell'array?

<b>A</b>	$k$	<b>B</b>	$n/k$
<b>C</b>	$n$	<b>D</b>	1

**Soluzione:**

C.  $n$ . Poiché ogni nodo della lista viene scandito una sola volta non si ha località temporale. Inoltre, nodi consecutivi potrebbero essere allocati in blocchi di memoria distinti. Pertanto nel caso più sfortunato si potrebbe avere un cache miss per ognuno degli  $n$  nodi della lista.

---

**Domanda 24\***

Si considerino gli accessi a memoria generati dalla scansione lineare di un array. Usando una cache completamente associativa gli accessi all'array generano meno cache miss rispetto a una cache ad indirizzamento diretto?

<b>A</b>	Vero	<b>B</b>	Falso
----------	------	----------	-------

**Soluzione:**

B. Falso. Scandire sequenzialmente un array porta a  $n/k$  cache miss indipendentemente dal tipo di cache. Infatti, una volta caricato un blocco i successivi  $k-1$  accessi ricadranno nello stesso blocco.