

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

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

# A

**Esame del 19/02/2015 (non esonerati dalla prima parte) – Durata 1h 30'**

Inserire nome, cognome e matricola nel file `studente.txt` e le risposte (A, B, C, D o E per ogni domanda) nel file `risposte.txt`. Rispondere E equivale a non rispondere (0 punti).

### Domanda 1 (compilazione IA32 di costrutti C)

Si consideri il seguente frammento C, dove `a` è di tipo `int` ed è tenuta in `eax`:

```
while (a>0) a--;
```

Quale delle seguenti funzioni IA32 è equivalente al codice C dato?

<b>A</b>	<pre>testl %eax, %eax L: jg E decl %eax E:</pre>	<b>B</b>	<pre>testl %eax, %eax L: jle E decl %eax E:</pre>
<b>C</b>	<pre>testl %eax, %eax L: jg E decl %eax jmp L E:</pre>	<b>D</b>	<pre>testl %eax, %eax L: jle E decl %eax jmp L E:</pre>

Motivare la risposta nel file `M1.txt`. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** D. Infatti, si esce dal ciclo se la guardia `a>0` diventa falsa, quindi se `a<=0` (`jle E`). A e B sono ovviamente errate poiché realizzano un `if` e non un `while`. C effettua il test opposto.

### Domanda 2 (comprensione programmi assembly)

Si consideri il seguente programma formato da un modulo C e da uno assembly IA32:

<pre>#include &lt;stdio.h&gt;  int f(int* v, int n);  int main() {     int v[] = { 1, 4, 0, 2 };     printf("%d\n", f(v, 4));     return 0; }</pre>	<pre>f:  pushl %esi     movl 12(%esp), %ecx     movl \$-1, %eax     testl %ecx, %ecx     jle L5     movl 8(%esp), %esi     xorl %edx, %edx L3:  cmpl \$0, (%esi,%edx,4)     je L4     incl %edx     cmpl %ecx, %edx     jl L3     jmp L5 L4:  movl %edx, %eax L5:  popl %esi     ret</pre>
---	--

Cosa stampa il programma?

<b>A</b>	-1	<b>B</b>	1	<b>C</b>	2	<b>D</b>	7
----------	----	----------	---	----------	---	----------	---

Motivare la risposta nel file `M2.txt`. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** C. La funzione restituisce il primo indice `i` tale che `v[i]` è zero, e -1 altrimenti.

---

**Domanda 3 (convenzioni System V ABI)**

Si consideri il seguente frammento C:

```
void (int a, int b) {  
    g(a, b);  
}
```

Quale delle seguenti funzioni IA32 è equivalente al codice C dato?

<b>A</b>	f: movl 8(%esp), %ecx movl 4(%esp), %edx subl \$12, %esp movl %ecx, (%esp) movl %edx, 4(%esp) call g addl \$12, %esp ret	<b>B</b>	f: movl (%esp), %ecx movl 4(%esp), %edx subl \$12, %esp movl %ecx, (%esp) movl %edx, 4(%esp) call g addl \$12, %esp ret
<b>C</b>	f: movl 4(%esp), %ecx movl 8(%esp), %edx subl \$12, %esp movl %ecx, (%esp) movl %edx, 4(%esp) call g addl \$12, %esp ret	<b>D</b>	f: movl 4(%esp), %ecx movl 8(%esp), %edx subl \$12, %esp movl %ecx, 4(%esp) movl %edx, 8(%esp) call g addl \$12, %esp ret

Motivare la risposta nel file M3.txt. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** C. La A passa correttamente a g i parametri, ma in ordine inverso. La B accede in modo errato ai parametri passati a f, mentre la D passa in modo errato i parametri a g.

---

**Domanda 4 (modi di indirizzamento e calcolo di espressioni aritmetiche)**

La seguente istruzione:

```
leal -4(%eax,%ecx,4), %edx
```

Equivale a:

<b>A</b>	movl %ecx, %edx imull \$4, %edx addl %eax, %edx subl \$4, %edx movl (%edx), %edx	<b>B</b>	movl %eax, %edx imull \$4, %edx addl %ecx, %edx addl \$-4, %edx
<b>C</b>	movl %ecx, %edx imull \$4, %edx addl %eax, %edx subl \$4, %edx	<b>D</b>	movl %edx, %ecx imull \$4, %ecx addl %eax, %ecx subl \$4, %edx

Motivare la risposta nel file M4.txt. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** C. L'istruzione lea assegna infatti  $edx = eax + ecx * 4 - 4$ , che è equivalente alle seguenti operazioni:

```
edx = ecx;  
edx *= 4; // ora edx == ecx*4  
edx += eax; // ora edx == eax + ecx*4  
edx -= 4; // ora edx == eax + ecx*4 - 4
```

---

**Domanda 5 (istruzioni conversione formato numerico)**

Se  $R[\%eax]=0xDEADFA7E$ , quanto vale  $R[\%eax]$  dopo l'istruzione `movsbl %a1, %eax`?

<b>A</b>	0xDEAD007E	<b>B</b>	0xFFFFFFFF7E	<b>C</b>	0x0000007E	<b>D</b>	0xDEADFF7E
----------	------------	----------	--------------	----------	------------	----------	------------

Motivare la risposta nel file `M5.txt`. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** C. Infatti, il bit più significativo di `a1` è 0 ( $7_{16}=0111_2$ ) e l'estensione del segno azzerà tutti i bit rimanenti di `eax`.

---

**Domanda 6 (analisi delle prestazioni del software)**

Qual è lo speedup ottenibile per un programma se riduciamo del 10% una sua porzione che richiede il 20% del tempo di esecuzione?

<b>A</b>	1.20x	<b>B</b>	2.00x
<b>C</b>	1.02x	<b>D</b>	1.95x

Motivare la risposta nel file `M6.txt`. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** C. Ridurre del 10% il tempo di esecuzione di una porzione di codice vuol dire ottenere uno speedup  $k=1/0.9$ . Sia infatti  $T_A$  il tempo di esecuzione di quella porzione prima della riduzione e  $T'_A$  il tempo dopo la riduzione. Si ha:  $T'_A = T_A - 0.1 * T_A = 0.9 * T_A$ , da cui  $k = T_A / T'_A = 1/0.9$ . Per la legge di Amdahl, posto  $a=0.2$  e  $k=1/0.9$ , lo speedup complessivo del programma è  $1/(a/k+1-a) = 1/(0.2*0.9+1-0.2) = 1.02x$ .

---

**Domanda 7 (tecniche di ottimizzazione di programmi)**

Si considerino i seguenti frammenti di programma C:

Codice originario	Codice ottimizzato
<pre>void c(int* a, int* b) {     for (i=0; i&lt;l(a); i++)         b[i] = g(a[i]); }</pre>	<pre>void c(int* a, int* b) {     int j = l(a);     for (i=0; i&lt;j; i++)         b[i] = g(a[i]); }</pre>

Quale delle seguenti affermazioni è falsa?

<b>A</b>	Per passare dal codice originario a quello ottimizzato è stata applicata la tecnica del loop-invariant code motion	<b>B</b>	Per passare dal codice originario a quello ottimizzato è stata applicata la tecnica della strength reduction
<b>C</b>	L'ottimizzazione è corretta se la funzione <code>l</code> non effettua effetti collaterali	<b>D</b>	In generale non possiamo aspettarci che l'ottimizzazione applicata possa essere effettuata da un compilatore

Motivare la risposta nel file `M7.txt`. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** B. La strength reduction consiste nel rimpiazzare una o più istruzioni con istruzioni meno onerose computazionalmente. Qui si ha spostamento di codice da un punto all'altro per evitare che venga rieseguito inutilmente.

---

**Domanda 8 (sistemi di memoria)**

Si consideri un sistema con una piccola cache associativa a due vie contenente 4 linee da 8 byte ciascuna. Quanti cache miss vengono generati dal seguente frammento di programma? Assumere che le variabili `v` e `i` siano tenute in dei registri, che l'array `v` sia allineato a un indirizzo multiplo di 8 byte e che la cache inizialmente non contenga alcun blocco di

memoria in uso al processo.

```
short v[20];  
for (i=0; i<20; i++) v[i] = 0;
```

<b>A</b>	4	<b>B</b>	5
<b>C</b>	8	<b>D</b>	20

Motivare la risposta nel file M8 .txt. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** B. Infatti, poiché ogni linea di cache contiene 4 celle dell'array, viene fatto un cache miss ogni 4 accessi.

---

#### Domanda 9 (memoria virtuale)

Si consideri un sistema di memoria virtuale con uno spazio logico di 4 GB, uno spazio fisico di 2 GB e pagine di dimensione 4 KB. Quanti byte occupa una tabella delle pagine per mantenere la corrispondenza tra pagine e frame? Ignorare i flag associati alle pagine come i bit di validità e assumere di usare 32 bit per rappresentare gli indici dei frame.

<b>A</b>	4 MB	<b>B</b>	1 MB
<b>C</b>	8 KB	<b>D</b>	256 KB

Motivare la risposta nel file M9 .txt. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** A. 4 GB sono suddivisi in  $2^{32-12}=2^{20}$  pagine da 4 KB. Per ogni pagina la tabella delle pagine mantiene un indice di frame di 4 byte, quindi la dimensione della tabella delle pagine è  $4*2^{20}$  byte = 4 MB.

---

#### Domanda 10 (tempi di latenza tipici di un sistema di calcolo)

Quali delle seguenti affermazioni è falsa?

<b>A</b>	Un accesso a registro è circa cento milioni di volte più veloce di un accesso a un disco magnetico	<b>B</b>	Un accesso a una memoria cache è circa cento volte più veloce di un accesso a un disco magnetico
<b>C</b>	Un accesso a una memoria DRAM è circa 10 volte più lento di un accesso a una memoria cache	<b>D</b>	Un accesso a una memoria cache può arrivare ad essere circa dieci volte più lento di un accesso a registro

Motivare la risposta nel file M10 .txt. **Risposte non motivate saranno considerate nulle.**

**Soluzione:** B. Un accesso a disco richiede circa 10 millisecondi ( $10^{-2}$  secondi) mentre una cache ha tempi di accesso dell'ordine dei nanosecondi ( $10^{-9}$  secondi). La differenza prestazionale non è quindi di 2 ordini di grandezza (100x), ma di almeno 6 (1000000x).

---

#### Domanda 11 (allocazione dinamica della memoria)

In quale caso si ha frammentazione interna?

<b>A</b>	Quando vi sarebbe sufficiente spazio libero per accomodare una richiesta di allocazione, ma lo spazio non è contiguo	<b>B</b>	Quando viene allocato più spazio di quello effettivamente richiesto e quindi parte dello spazio allocato non è realmente utilizzato
----------	--	----------	---

Motivare la risposta nel file M11 .txt (ad esempio spiegando perché si scarta una possibile risposta). **Risposte non motivate saranno considerate nulle.**

**Soluzione:** B. La risposta A si riferisce infatti alla frammentazione esterna.

---

**Domanda 12 (flusso del controllo eccezionale)**

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

<b>A</b>	Quando si preme un tasto sulla tastiera viene generato un interrupt	<b>B</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
<b>C</b>	Quando si effettua un click del mouse viene generata una trap	<b>D</b>	Trap e interrupt consentono di entrare in modalità di esecuzione supervisore

Motivare la risposta nel file `M12.txt` (ad esempio spiegando perché si scarta una possibile risposta). **Risposte non motivate saranno considerate nulle.**

**Soluzione:** C. Una trap è un'istruzione macchina che genera un interrupt software di tipo sincrono. Il click del mouse è invece un evento asincrono che genera un interrupt hardware.