

Discussione della domanda 3 dell'esercitazione del 10 maggio 2019

Domanda 3 Data la seguente porzione di codice, quale delle seguenti affermazioni è *FALSA*?

```
#include<stdio.h>
#include<unistd.h>

int main(int argc, char *argv[]) {
    int s = getpagesize();
    int m = s - 1;
    int t = s, k = 0;
    while (t >>= 1) k++;

    int x = 100;
    void *p = &x;

    unsigned A = (unsigned)p / s;
    unsigned B = (unsigned)p % s;
    unsigned C = (unsigned)p >> k;
    unsigned D = (unsigned)p & m;
    unsigned E = (unsigned)p & ~m;
    unsigned F = (unsigned)p % (1 << k);

    return 0;
}
```

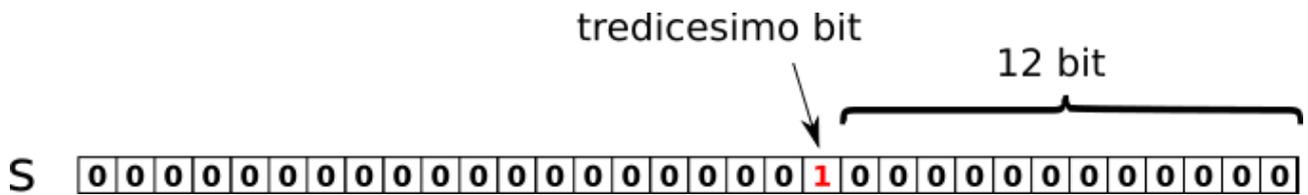
- **A.** A è il numero di pagina dell'indirizzo di x
- **B.** B è l'offset (all'interno della pagina) dell'indirizzo di x
- **C.** C è il numero di pagina dell'indirizzo di x
- **D.** D è l'offset (all'interno della pagina) dell'indirizzo di x
- **E.** E è il numero di pagina dell'indirizzo di x
- **F.** F è l'offset (all'interno della pagina) dell'indirizzo di x

Analizziamo il codice passo-passo.

La seguente istruzione assegna alla variabile s la dimensione delle pagine di memoria:

```
int s = getpagesize();
```

Assumendo di essere in un sistema con pagine di 4KB alla variabile s verrà assegnato il valore 4096. In binario:

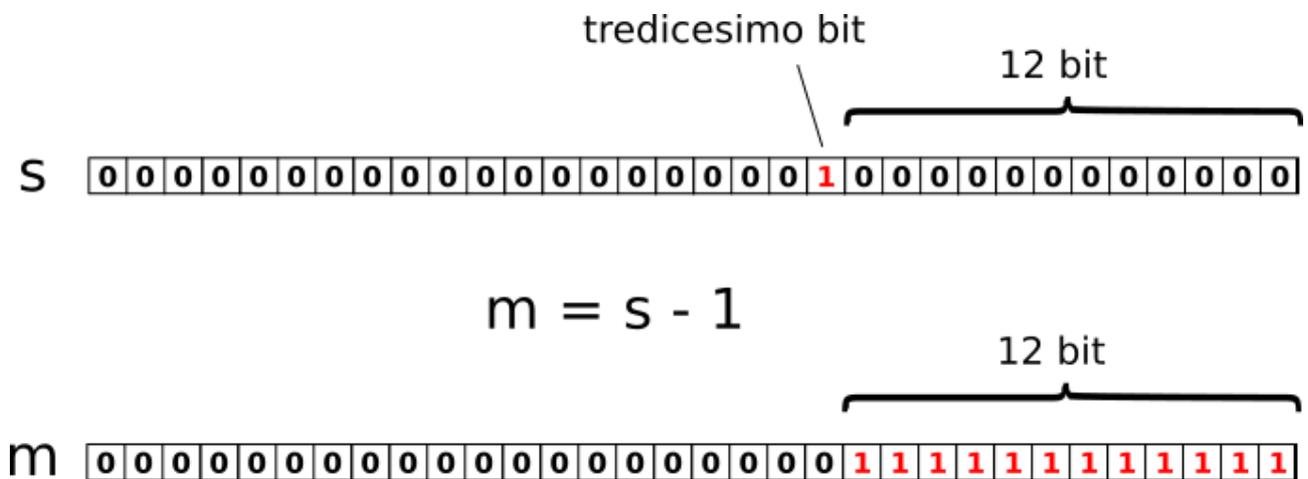


Si noti che essendo s una potenza di 2, ed in particolare 2^{12} , s ha i 12 bit meno significativi a 0 e il 13-esimo bit a 1.

Passiamo all'istruzione successiva:

```
int m = s - 1;
```

Poiché a m viene assegnato $s - 1$, m avrà i 12 bit meno significativi a 1 e tutti gli altri bit a 0, come mostrato in figura:



Veniamo ora al ciclo che calcola k :

```
int t = s, k = 0;
while (t >>= 1) k++;
```

Nella parte di inizializzazione viene assegnata s a t e 0 a k . Nel ciclo, t (che inizialmente è pari a s) viene ripetutamente shiftata a destra di 1 bit (si noti che $t >>= 1$ corrisponde a $t = t >> 1$) fino a che non raggiunge il valore 0. Poiché t ha solo il 13-esimo bit impostato a 1, tale condizione si verifica alla 13-esima esecuzione della condizione del ciclo, che quindi porta all'uscita dal ciclo. Di conseguenza il ciclo esegue 12 iterazioni e quindi 12 incrementi di k .

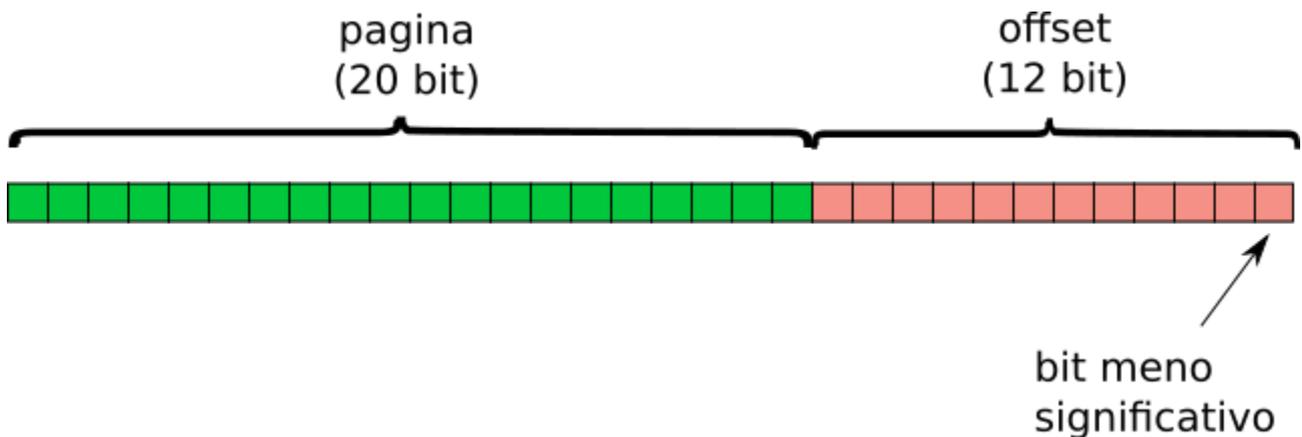
In generale se la dimensione della pagina è uguale a un numero 2^m , alla fine del ciclo k sarà pari all'esponente m (si noti quindi che il ciclo non fa altro che calcolare il logaritmo binario della variabile t).

Si veda la figura seguente per un maggiore chiarimento:

A questo punto abbiamo tutti gli elementi per poter rispondere alla domanda.

Prima di procedere all'analisi delle varie risposte, si ricorda che dato un indirizzo (logico) su una architettura a n bit con pagine di dimensione 2^k , i k bit meno significativi corrispondono all'offset all'interno della pagina, mentre i restanti $n-k$ bit corrispondono al numero di pagina.

Ad esempio, su un sistema a 32 bit con pagine di $2^{12} = 4096$ byte, i 12 bit meno significativi corrispondono all'offset, mentre i restanti al numero di pagina, come mostrato in figura:



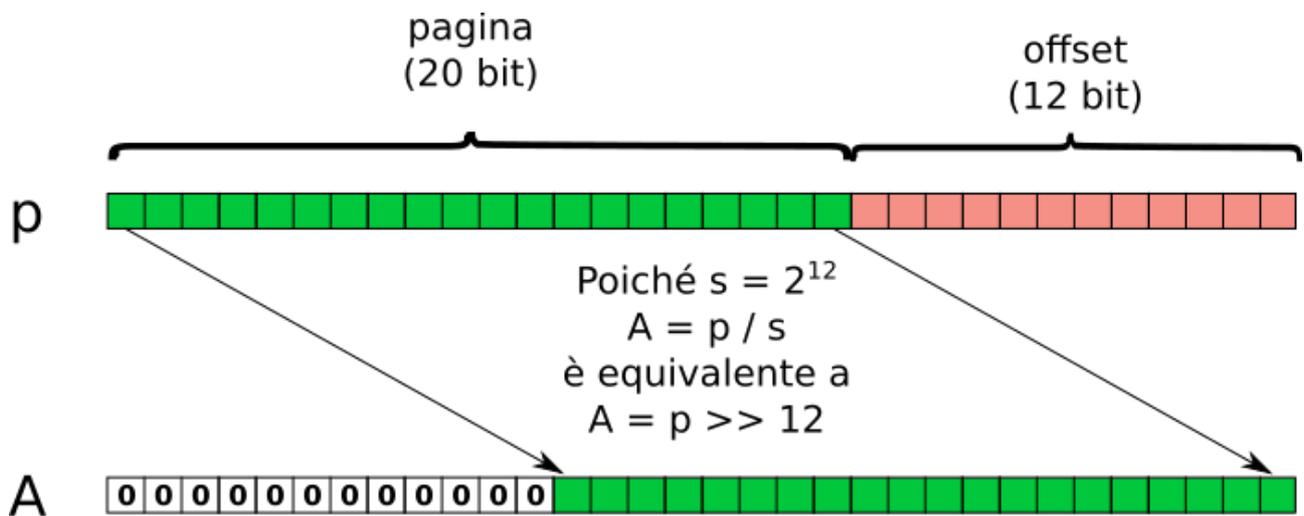
Risposta A: A è il numero di pagina dell'indirizzo di x

$$\text{unsigned } A = (\text{unsigned})p / s;$$

La variabile p contiene l'indirizzo di x (che può essere un numero qualsiasi). La variabile s è 2^{12} . Si ricorda che dividere un numero per una potenza di 2, ad esempio 2^k , è equivalente a shiftarlo a destra di k bit.

Perciò assegnare ad A il valore p / s , è equivalente ad assegnargli il valore di $p \gg 12$.

Shiftando p di 12 bit a destra vengono eliminati tutti i bit dell'offset e rimangono solo quelli del numero di pagina, come mostrato nella seguente figura:



A risulta quindi uguale al numero di pagina dell'indirizzo in p

Perciò A contiene il numero di pagina dell'indirizzo contenuto in p (cioè l'indirizzo di x).

Quindi l'affermazione della risposta A è vera (e di conseguenza non è la risposta corretta alla domanda 3).

Si noti che poiché $k = 12$, lo stesso ragionamento si applica alla risposta C.

Risposta B: B è l'offset (all'interno della pagina) dell'indirizzo di x

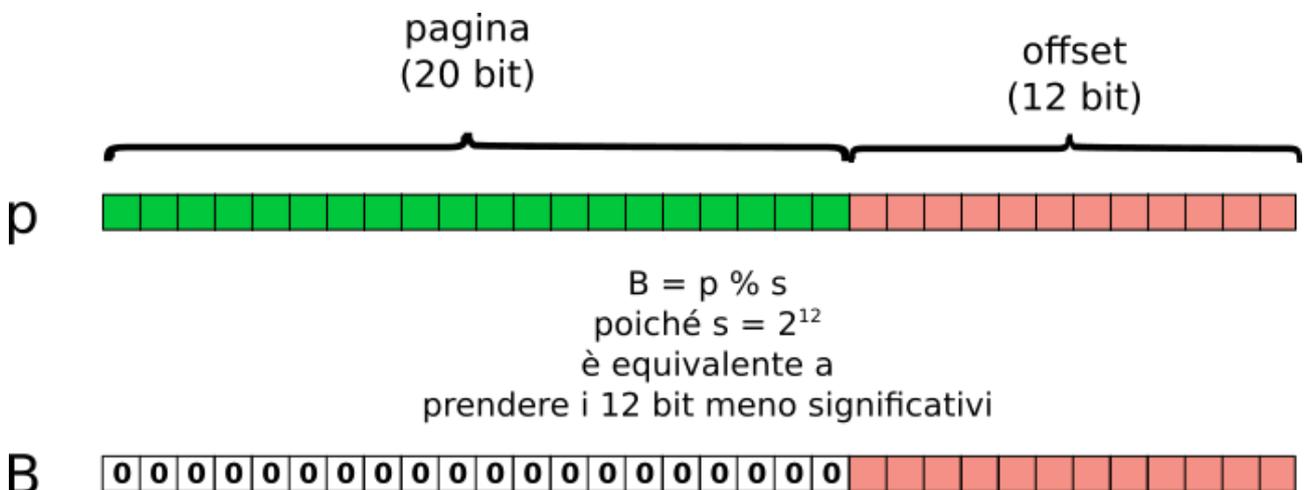
Alla variabile B viene assegnato il resto della divisione per s:

```
unsigned B = (unsigned)p % s;
```

Il resto della divisione di un numero non-negativo qualsiasi per s è un numero compreso nell'intervallo $[0, s - 1]$.

Poiché s è 2^{12} , l'intervallo $[0, s - 1]$ è rappresentato proprio dai 12 bit meno significativi. Perciò $p \% s$ è uguale ai 12 bit meno significativi di p, corrispondenti proprio all'offset dell'indirizzo di x.

Si veda la figura seguente come ulteriore chiarimento:



L'affermazione della risposta B è quindi vera (e di conseguenza non è la risposta corretta alla domanda 3).

Risposta C: C è il numero di pagina dell'indirizzo di x

Si veda il ragionamento relativo alla risposta A.

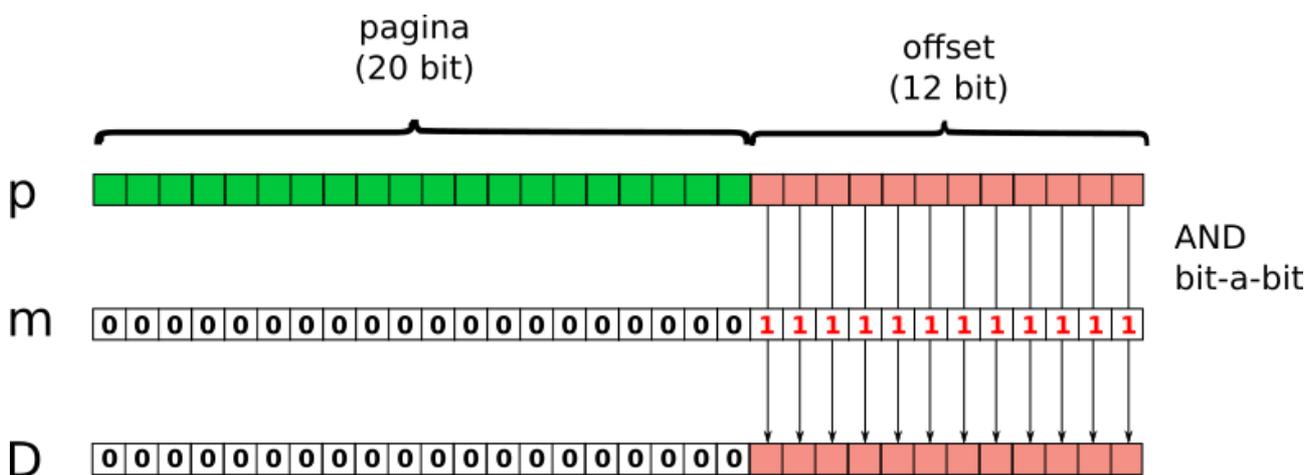
Risposta D: D è l'offset (all'interno della pagina) dell'indirizzo di x

Alla variabile D viene assegnato il risultato del AND bit-a-bit tra l'indirizzo p e la variabile m.

$$\text{unsigned } D = (\text{unsigned})p \ \& \ m;$$

Come già discusso in precedenza, la variabile m ha i 12 bit meno significativi a 1 e i restanti bit a 0.

Perciò l'espressione $p \ \& \ m$ vale i 12 bit meno significativi di p. Si veda la figura seguente:



si noti che per ogni bit b_i di p si ha:

$$b_i \text{ AND } 0 = 0$$

$$b_i \text{ AND } 1 = b_i$$

Perciò, D avrà i 20 bit più significativi a 0 e i 12 bit meno significativi pari ai 12 bit meno significativi di p

Perciò a D viene effettivamente assegnato l'offset dell'indirizzo contenuto in p (cioè l'indirizzo di x) e l'affermazione D risulta vera (e di conseguenza non è la risposta corretta alla domanda 3).

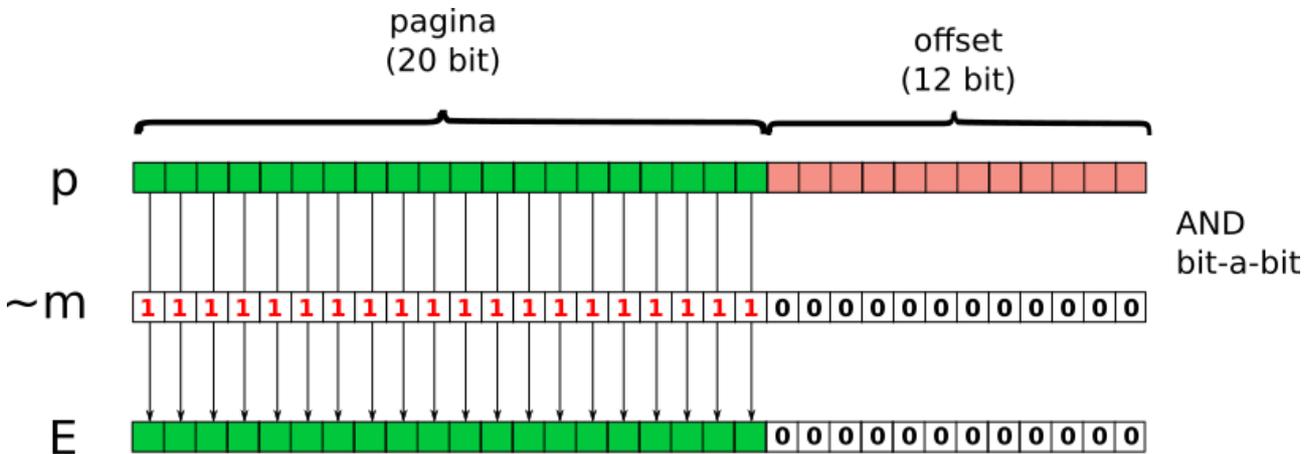
Risposta E: E è il numero di pagina dell'indirizzo di x

Alla variabile E viene assegnato il risultato del AND bit-a-bit tra l'indirizzo p e la variabile $\sim m$.

$$\text{unsigned } E = (\text{unsigned})p \& \sim m;$$

L'operatore \sim è la negazione bit-a-bit (o complemento). Ha l'effetto di invertire tutti i bit. Perciò se m ha i 12 bit meno significativi a 1 e i restanti 20 a 0, $\sim m$ ha i 12 bit meno significativi a 0 e i restanti a 1.

Perciò ad E vengono assegnati i 20 bit più significativi di p , come mostrato in figura:



si noti che per ogni bit b_i di p si ha:

$$b_i \text{ AND } 0 = 0$$

$$b_i \text{ AND } 1 = b_i$$

Perciò, E avrà i 12 bit meno significativi a 0 e i 20 bit più significativi pari ai 20 bit più significativi di p

I 20 bit più significativi rappresentano effettivamente il numero di pagina dell'indirizzo contenuto in p (cioè dell'indirizzo di x).

Tuttavia come è evidente dalla figura risultano shiftati di 12 bit a sinistra. Perciò E contiene di fatto il numero di pagina dell'indirizzo di x moltiplicato per 2^{12} (4096).

Ne segue che l'affermazione della risposta E è falsa, e quindi è la risposta corretta alla domanda 3.

Risposta F: F è l'offset (all'interno della pagina) dell'indirizzo di x

$$\text{unsigned } F = (\text{unsigned})p \% (1 \ll k);$$

1 shiftato di $k=12$ bit è esattamente uguale a $s = 2^k=2^{12}$. Perciò $p \% (1 \ll k)$ è equivalente a $p \% s$.

Ne segue che F è uguale a B , e quindi l'affermazione F è vera (e di conseguenza non è la risposta corretta alla domanda 3).