

Laboratorio su standard IEEE 754

Vitoantonio Bevilacqua
vitoantonio.bevilacqua@poliba.it

Richiamo: *Floating point*

Dato un numero reale, questo numero si può rappresentare nella macchina seguendo lo standard IEEE 754 a singola o a doppia precisione:

- Nello standard in singola precisione (quello che la maggior parte dei calcolatori utilizza quando si decide di dichiarare una variabile "float"), la gestione è quella di utilizzare un pacchetto di 32 bit (4 Bytes) organizzati nel seguente modo:

1 Bit (Segno)	8 bit (Esponente)	23 bit (Mantissa)
------------------	----------------------	----------------------

- Nello standard in doppia precisione (utilizzato quando si dichiara una variabile di tipo "double"), il "pacchetto" di bit utilizzati per rappresentare il numero è di 64 (8 Bytes):

1 Bit (Segno)	11 bit (Esponente)	52 bit (Mantissa)
------------------	-----------------------	----------------------

Dopo aver scelto la codifica da effettuare, il calcolatore inserisce, nel primo bit a sinistra, il valore, in binario, del segno dell'intero numero (0 = segno positivo, 1 = segno negativo).

Nella seconda parte si inserisce l'esponente in forma polarizzata, ovvero sommando a questo una costante chiamata "bias" (che è uguale a 127 nel caso della singola precisione, quindi si hanno 256 possibili configurazioni a ciascuna delle quali si dà un significato numerico); all'interno di questa parte si deve riuscire ad inserire anche gli infiniti, infinitesimi e i "Not a Number".

- In un calcolatore il concetto di numero infinito non è lo stesso concetto utilizzato in matematica. L'ordine di grandezza dell'infinito (singola precisione) è $2^{255-BIAS} = 2^{128}$, quello dello 0 o di un infinitesimo è 2^{-127} (numero infinitamente piccolo).
- Esiste anche la rappresentazione denormalizzata che, quando il numero si avvicina sempre più allo zero, secondo una certa logica, il calcolatore fa diventare i numeri ancora più piccoli spostando il bit più significativo di mantissa.

ESEMPIO:

```
float a = 3.5; /* Il compilatore trova nella ram 4 Bytes adiacenti organizzandoli secondo lo
               standard IEEE 754. */

float* p; /* Dichiaro una variabile di tipo puntatore a float */

p = &a; /* Assegno l'indirizzo della variabile "a" alla variabile "p" */
```

Se il valore della variabile "a" è stato memorizzato (ad esempio) a partire dal Byte 8 nella Ram, in "p" verrà memorizzato il numero 8.

Ora si vuole trovare un modo che mi permetta di stampare solo una sequenza di bit e quindi di saltare tra gli indirizzi della Ram secondo la logica Byte a Byte. Se scrivessi:

```
p = p+1; /* Incremento il valore di 4 Bytes */
```

Se si incrementa una variabile di tipo puntatore, allora, secondo l'aritmetica dei puntatori, la variabile verrà incrementata della dimensione dei Bytes della variabile a cui punta.

Nel caso la variabile fosse stata di tipo "double", l'incremento sarebbe stato di 8 Bytes.

Per poter incrementare la variabile di 1 solo Byte, si deve dichiarare una nuova variabile di tipo puntatore a "char" ed associarla alla variabile "p" con l'operazione di "Casting" che quindi forza la variabile "p" ad essere del tipo indicato. Ovvero:

```
char* c; // Dichiaro una variabile di tipo puntatore a char  
c = (char*)p; // Forzo la variabile "p" ad essere di tipo char *
```

A questo punto, la variabile "c" avrà valore 8, che è il valore memorizzato in "p", quindi, se si dovesse incrementare "c", l'incremento sarebbe di 1 (poiché una variabile di tipo "char" è composta da 8 bit e quindi 1 Byte). Se ora si volesse stampare il valore nel Byte che desideriamo mostrare, sarebbe sufficiente eseguire la funzione "printf" della variabile puntata da c (ovvero *c):

```
printf("%c", *(c + 2)); /* Eseguo la stampa a video del Byte 10 (8+2) */
```

Traccia di ESERCIZIO:

Sviluppare un programma che richieda un numero reale all'utente, convertirlo in Floating Point e mostrare a video la sequenza dei 32 bit relativa alla codifica di quel numero nello standard IEEE 754 in singola o doppia precisione (cercare dispense del corso di Fondamenti di Informatica).