

Modulo: Approfondimenti sui Sistemi Aritmetici di un computer: tipo reale [P2_03]

Unità didattica: Struttura di campi di bit [approfondimento C] [2-AC]

Titolo: Struttura di campi di bit

Argomenti trattati:

- ✓ Come accedere ai campi (s,e,m) di un numero floating-point
- ✓ Cos'è la struttura di campi di bit in C
- ✓ Esempi d'uso

Prerequisiti richiesti: `struct{}` e `union{}` del C

Per estrarre i campi (s,e,m) da un numero floating-point è utile usare una struttura di campi di bit in C

Esempio

```
struct F_fields /** STRUTTURA DI CAMPI DI BIT **/  
{  
    unsigned int m: 23; // dai bit meno significativi  
    unsigned int e:  8;  
    unsigned int s:  1; // a quelli più significativi  
};
```

numero di bit per il campo della struttura

Struttura per data (gg,mm,aa)

Esempio 0

```
int main()
{
    /* data: {giorno=xx, mese=yy, anno=zz}, anno dal 2000 */
    typedef struct{
        unsigned int gg; /* gg in {1,2,...,31} */
        unsigned int mm; /* aa in {1,2,...,12} */
        unsigned int aa; /* aa in {0,1,...,50} */
    } DATA;
    DATA oggi;
    oggi.gg = 31;
    oggi.mm = 03;
    oggi.aa = 14; /* sta per 2014 */
    printf("Data di oggi: gg = %u\n", (unsigned int)oggi.gg);
    printf("sizeof(oggi) = %u byte\n",
(unsigned int)sizeof(oggi));
    return 0;
}
```

Data di oggi: gg = 31
sizeof(oggi) = 12 byte

Campi di bit in una struttura

Esempio 1

```
int main()
{
    /* data: {giorno=xx, mese=yy, anno=zz}, anno dal 2000 */
    typedef struct{
        unsigned int gg: 5 ; // 1...31 su 5 bit
        unsigned int mm: 4 ; // 1...12 su 4 bit
        unsigned int aa: 6 ; // 0...50 su 6 bit
    } DATA;

    DATA oggi;
    oggi.gg = 31;
    oggi.mm = 03;
    oggi.aa = 20; /* sta per 2020 */
    printf("Data di oggi: gg = %u\n", oggi.gg);
    printf(" sizeof(oggi)  = %u byte\n",
        (unsigned int)sizeof(oggi));

    return 0;
}
```

numero di bit per il campo della struttura

Data di oggi: gg = 31
sizeof(oggi) = 4 byte

Campi di bit in una struttura

Esempio 2

```
int main()
{
    /* data: {giorno=xx, mese=yy, anno=zz}, anno dal 2000 */
    typedef struct{
        unsigned short gg: 5; // 1...31 su 5 bit
        unsigned short mm: 4; // 1...12 su 4 bit
        unsigned short aa: 6; // 0...50 su 6 bit
        // totale = 15 bit
    } DATA;
    DATA oggi;
    oggi.gg = 31;
    oggi.mm = 03;
    oggi.aa = 20; /* sta per 2020 */
    printf("Data di oggi: gg = %u\n", (unsigned int)oggi.gg);
    printf(" sizeof(oggi)  = %u byte\n",
        (unsigned int)sizeof(oggi));

    return 0;
}
```

Data di oggi: gg = 31
sizeof(oggi) = 2 byte

Campi di bit in una struttura

Esempio 3

```
int main()
{
    /* data: {giorno=xx, mese=yy, anno=zz}, anno dal 2000 */
    typedef struct{
        unsigned char gg: 5; // 1...31 su 5 bit
        unsigned char mm: 4; // 1...12 su 4 bit
        unsigned char aa: 6; // 0...50 su 6 bit
        // totale = 15 bit 2 byte
    } DATA;
    DATA oggi;
    oggi.gg = 31;
    oggi.mm = 03;
    oggi.aa = 20; /* sta per 2020 */
    printf("Data di oggi: gg = %u\n", (unsigned int)oggi.gg);
    printf("sizeof(oggi) = %u byte\n",
        (unsigned int)sizeof(oggi));

    return 0;
}
```

```
Data di oggi: gg = 31
sizeof(oggi) = 2 byte
```

In funzione del numero totale di bit alloca un multiplo del numero di byte per il tipo

struttura di campi di bit per i campi (s,e,m) di un fl.-point

Esempio 1

P2_03_02.7

Tipo Reale Floating-Point

(prof. M. Rizzardi)

```
typedef struct {
    unsigned int m: 23; // dai bit meno significativi
    unsigned int e:  8; //                                formato little endian
    unsigned int s:  1; // a quelli più significativi
} FloatFields;

typedef union {
    float      F; // per il numero float
    FloatFields B; // per i campi di bit
} BasicSingle;

int main(void) /** (1) ASSEGNA UN VALORE AL FLOAT E VISUALIZZA IL VALORE DEI CAMPI **/
{
    BasicSingle x;
    x.F = -1.625f;
    int Bias = (1<<7) - 1; // per float n=8: 2^(n-1)-1
    printf("\nfloat x = %g      [float ==> Bias = %d]\n", x.F, Bias);
    printf("\tValore decimale del campo segno      di x = %d\n", x.B.s);
    printf("\tValore decimale del campo esponente di x = %d (biased)  = %d"
           " (unbiased)\n", x.B.e, x.B.e-Bias);
    printf("\tValore decimale del campo mantissa  di x = %d (intero)  = %g"
           " (frazionario)\n", x.B.m, x.B.m/((float)(1<<23)));
    return 0;}
```

```
float x = -1.625      [float ==> Bias = 127]
Valore decimale del campo segno      di x = 1
Valore decimale del campo esponente di x = 127 (biased)  = 0 (unbiased)
Valore decimale del campo mantissa  di x = 5242880 (intero) = 0.625 (frazionario)
```

struttura di campi di bit per i campi (s,e,m) di un fl.-point

Esempio 2

```
#define Mbit 23 // num. bit per mantissa di float
#define Ebit 8  // num. bit per esponente di float
#define Sbit 1  // num. bit per segno di float
```

cambiando le costanti, si può fare lo stesso per i tipi double e long double

```
typedef struct { /** STRUTTURA DI CAMPI DI BIT **/
    unsigned int m: Mbit; // dai bit meno significativi
    unsigned int e: Ebit;
    unsigned int s: Sbit; // a quelli più significativi
} FloatFields;
```

```
typedef union {
    float      F; // per il numero float
    FloatFields B; // per i campi di bit
} BasicSingle;
```

```
int main(void) /** (2) ASSEGNA IL VALORE AI CAMPI DI UN FLOAT **/
{
```

```
    BasicSingle x;
```

```
    x.B.s = 0x1;
```

```
    x.B.e = 0x7F;
```

```
    // segno -
```

```
    // esponente Bias = 0111 1111
```

```
    x.B.m = 0x100000; // mantissa = 001 0000 0000 0000 0000 0000
```

```
    printf("\nRappresentazione floating-point (s,e,m) di x = %g\n", x.F);
```

```
    return 0;
```

```
}
```

```
x.B.e = (unsigned char)~(1<<(Ebit-1));
```

Rappresentazione floating-point (s,e,m) di x = **-1.125**

struttura di campi di bit per i campi (s,e,m) di un fl.-point

Esempio 3

```
#define Mbit, Ebit, Sbit ...
typedef struct { unsigned int m: Mbit, e: Ebit, s: Sbit; } FloatFields;
typedef union { float F; FloatFields B; } BasicSingle;
int main(void) /** (3) VISUALIZZA I BIT DEL CAMPO ESPONENTE DI x.F **/
{
    BasicSingle x;
    x.F = -1.625f;
    unsigned char k, b, mask;
    mask = 1 << (Ebit-1); // 1000 0000
    printf("\nCampo esponente di x = %g:\n\tdec = %d\n\tbin = ", x.F, x.B.e);
    for (k=0; k<Ebit; k++)
    {
        b = mask & x.B.e;    b = b >> (Ebit-1-k);
        printf("%d", b);
        mask = mask >> 1;
    }
    return 0;
}
```

Campo esponente di $x = -1.625$:
dec = 127
bin = 01111111

perché?

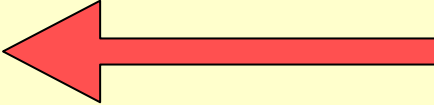
struttura di campi di bit per i campi (s,e,m) di un fl.-point

Esempio 4

```
#define Mbit, Ebit, Sbit ...
typedef struct { unsigned int m: Mbit, e: Ebit, s: Sbit; } FloatFields;

typedef union {
    float      F; // per il numero float
    unsigned int I; // per gli operatori bitwise
    FloatFields B; // per i campi di bit
} BasicSingle;

int main(void) /** (4) COSTRUISCE MASCHERA PER ESTRARRE IL CAMPO ESPONENTE **/
{
    BasicSingle x, ex, maskE;
    x.F = -1.625f;
    unsigned char k, b, mask;
    maskE.I = 0; // azzera la maschera
    maskE.B.e = 0xFF; // campo esponente di maschera: 1111 1111
    ex.I = x.I & maskE.I; // estrae il campo esponente di x
    printf("\nCampo esponente di x = %g: dec = %d\n", x.F, ex.B.e);
    return 0;
}
```



Campo esponente di x = -1.625: dec = 127