

Contents

Ottimizzazione della chiamata di coda

1

Ottimizzazione della chiamata di coda

Una funzione f_1 , nel suo corpo, chiama f_2 , si dice chiamata di coda se dopo questa chiamata non c'è altro codice.

In questo caso, il compilatore può ottimizzare la funzione f_1 , riutilizzando lo stack di f_1 per utilizzarlo con f_2 così da non creare un nuovo frame della funzione nello stack.

Questa ottimizzazione viene fatta in caso di **sibling code**, cioè quando f_2 :

- Ha un valore di ritorno che occupa lo stesso spazio di f_1
- Una sequenza di parametri che occupa lo stesso spazio di f_1

Esempio di chiamate di coda:

```
// Non è una chiamata di coda perchè c'è 1+g
int f(int x,...,int n) {
    ...
    return 1 + g(x,...,n);
}
```

```
// è una chiamata di coda
int f(int x,...,int n) {
    ...
    return g(x,...,n);
}
```

Una funzione ricorsiva di coda contiene solo chiamate di coda

Questa ottimizzazione, mantiene la dimensione dello stack costante.

Per rendere una funzione ricorsiva, una tail call recursion, usiamo l'accumulatore come parametro della funzione. Senza esplicitarlo nel return.