

Linguaggi di Programmazione

Francesca Del Bonifro¹

¹Dipartimento di Informatica - Scienza e Ingegneria - UniBO

29/10/2019

Esercitazione

- Gestione della memoria
- Implementazione della regola di scope statico: catena statica e display
- Implementazione della regola di scope dinamico: A-List e CRT

Esercizio 1

È dato il seguente frammento di codice in uno pseudolinguaggio con scope statico

```
void P1 {  
    void P2 {  
        corpo di P2  
    }  
    void P3 {  
        void P4 {  
            corpo di P4  
        }  
    }  
    corpo di P1  
}
```

Si descriva graficamente la pila dei record di attivazione, limitatamente ai puntatori di catena statica e dinamica, dopo la successione di chiamate (tutte attive) P1, P3, P4, P2, P3. Si descriva inoltre brevemente come viene determinato il puntatore di catena statica di P2 quando questa procedura è chiamata da P4.

Esercizio 1

E utilizzando il display?

È dato il seguente frammento di codice in un pseudolinguaggio con scope statico

```
void P1 {  
    void P2 {  
        corpo di P2  
    }  
    void P3 {  
        void P4 {  
            corpo di P4  
        }  
    }  
    corpo di P1  
}
```

Si descriva graficamente la pila dei record di attivazione, limitatamente ai puntatori di catena statica e dinamica, dopo la successione di chiamate (tutte attive) P1, P3, P4, P2, P3. Si descriva inoltre brevemente come viene determinato il puntatore di catena statica di P2 quando questa procedura è chiamata da P4.

Esercizio 2

Supponendo di essere in un linguaggio con scoping statico, implementato mediante display, si dia graficamente la situazione del display e della pila dei record di attivazione al momento in cui il controllo entra per la seconda volta nella funzione A.

```
{int x = 0;
  int A(reference int y) {
    int x =2;
    y=y+1;
    return B(y)+x;
  }
  int B(reference int y){
    int C(reference int y){
      int x = 3;
      return A(y)+x+y;
    }
    if (y==1) return C(x)+y;
    else return x+y;
  }
  write (A(x));
}
```

Esercizio 3

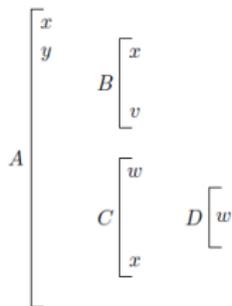
È dato il seguente frammento di codice in uno pseudolinguaggio con `goto`, scope statico e blocchi annidati etichettati (indicati con `A :{...}`):

```
A: { int x = 1;
    int y = 2;
    goto C;
    B: {int x = 3;
        int z = 4;
        int y = 5;
        E: {int x = 6; // (**)
            }
        C: {int x = 7;
            D: {int x = 8;
                }
            goto B;
        }
    }
```

Lo scope statico è gestito mediante display. Si illustri graficamente la situazione del display (e della pila dei record di attivazione) nel momento in cui l'esecuzione raggiunge il punto segnato con il commento `(**)`.

Esercizio 4

Si consideri la struttura di blocchi schematizzata nella figura seguente; i nomi all'interno di un blocco indicano una dichiarazione di quel nome.

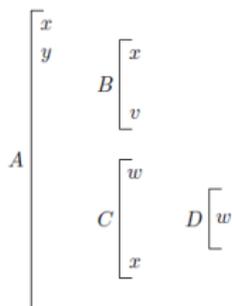


Si rappresenti graficamente l'ambiente per il blocco D di tale figura, dopo la sequenza di chiamate A,B,C,D, con scope dinamico realizzato mediante tabella centrale dell'ambiente (CRT).

Esercizio 4

Con A-list?

Si consideri la struttura di blocchi schematizzata nella figura seguente; i nomi all'interno di un blocco indicano una dichiarazione di quel nome.



Si rappresenti graficamente l'ambiente per il blocco D di tale figura, dopo la sequenza di chiamate A,B,C,D, con scope dinamico realizzato mediante tabella centrale dell'ambiente (CRT).

Esercizio 5

Si consideri il seguente frammento di programma scritto in uno pseudo-linguaggio che usa scope statico.

```
{
void f() {
    void g() {
        corpo_di_g;
    }

    void h() {
        void l(){
            corpo_di_l;
        }
        corpo_di_h;
    }
    corpo_di_f;
}
```

Si descriva graficamente l'evoluzione del display nella sequenza di chiamate f, h, l, g, h supponendo che tutte le chiamate rimangano attive (ossia nessuna funzione ha restituito il controllo).

Esercizio 6

Si consideri la seguente definizione di funzione

```
int f(int n, int m){
    if (n==0) return 1;
    else {
        m = m+1;
        return f(n-1, m*2)
    }
}
```

Qual è il numero minimo di RdA che una macchina astratta deve usare nel corso della valutazione di $f(5,0)$? Perché?

Esercizio 7

Si consideri l'implementazione dello scope statico mediante display. Si dica, motivando la risposta, se dimensione massima del display può essere determinata durante la compilazione.

Esercizio 8

Dato il seguente frammento di codice in uno pseudolinguaggio con goto, scope dinamico e blocchi annidati etichettati (indicati con A :{...}):

```
A: { int x = 5;
    int y = 4;
    goto C;
    B: {int x = 4;
        int z = 3;
        goto E;
        }
    C: {int x = 3;
        D: {int x = 2;
            }
        goto B;
        E: {int x = 1; // (**)
            }
        }
    }
```

Lo scope dinamico è gestito mediante tabella centrale dell'ambiente (CRT). Si illustri graficamente la situazione della CRT (esplicita, senza pila nascosta) in (**).

Esercizio 9

Dato il seguente frammento di codice in uno pseudolinguaggio con scope statico gestito con catena statica:

```
int x = 5;
int y = 4;
void B(){
    int x = 4;
    int z = 3;
    C();
}
void C(){
    int x = 3;
    void D(){
        int x = 2;
    }
    D();
}
B();
```

Si rappresenti graficamente la pila degli RdA subito dopo che il controllo è entrato nella funzione D.

Esercizio 10

Dato il seguente frammento di codice in uno pseudolinguaggio con scope statico gestito con catena statica:

```
int x = 10;
void A(){
    x=x+1;}
void B(){
    int x = 5;
    void C (){
        x=x+2; A();
    }
    A(); C();
}
B();
```

Si rappresenti graficamente la pila degli RdA.

Esercizio 11

```
{
  void a () {
    void b () {
      corpo_di_b
    }
    void c () {
      void d () {
        corpo_di_d
      }

      corpo_di_c
    }
    corpo_di_a
  }
}
```

Fornire uno schema del display in seguito alla chiamata di procedure a, b, c, d, b, considerando che alla fine della successione di chiamate tutte le procedure siano ancora attive, ovvero nessuna di esse ha ancora ritornato il controllo.

Esercizio 12

Mostrare la pila dei RdA del seguente codice, scritto in linguaggio a scope statico, dopo avere eseguito l'istruzione `foo(x, y)`, mostrando per ogni RdA:

- variabili locali e valori;
- funzioni;
- puntatore di catena statica;
- puntatore di catena dinamica.

```
A:{ int x = 1;
  B:{ int y = 0;
    void foo(int n, int m){
      x = m + n;
      y = m - n;
    }
    C:{ int x = 2;
      int y = 3;
      foo(x, y);
    }
  }
}
```