Informática PUC-Rio

Escopo, Ligação e Ambiente de Execução

Informática PUC-Rio

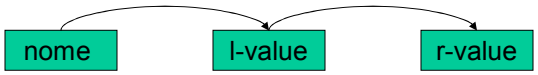
O que é uma variável?

- Uma variável pode ser definida como uma tupla <nome, escopo, tipo, l-value, r-value>
  - nome – é um string usado para identificar a variável no programa fonte;
  - escopo - região onde a variável é visível;
  - tipo – é o tipo de dado armazedo pela variável;
  - l-value – é a localização da variável na memória;
  - r-value – é o valor contido na memória.

Informática PUC-Rio

O que é uma variável?

- Antes que uma variável possa ser usada, é necessário alocar uma área de memória compatível com seu tipo.
- Esta área deverá permanecer reservada apenas enquanto a variável puder ser acessada pelo programa.



Informática PUC-Rio

Conceito de Ligação (Binding)

- Ligação é uma associação entre um objeto (variável, tipo, módulo, função, etc) e os seus atributos (nome, endereço, tipo, etc).
- Exemplos de ligação:
  - a associação entre um nome de uma variável e um endereço de memória;
  - a associação entre um nome e um tipo;
  - a associação entre um tipo e a sua representação;
  - a associação entre um nome e um escopo.

Informática PUC-Rio

Conceito de Ligação (Binding)

- A ligação pode ser estabelecida em diferentes momentos:
  - Especificação da LP;
  - Implementação da LP;
  - Compilação;
  - Linking;
  - Em tempo de execução.

Informática PUC-Rio

Conceito de Ligação (Binding)

- O momento em que ocorre a ligação pode ser classificado como cedo ou tardio (*early* or *late binding*).
- Usualmente quanto mais cedo ocorre a ligação, maior a eficiência de execução do programa mas menor a flexibilidade das estruturas disponibilizadas.
- LPs compiladas tendem a fazer maior uso de ligações que ocorrem relativamente cedo.
- LPs interpretadas tendem a fazer maior uso de ligações tardias.

Informática PUC-Rio

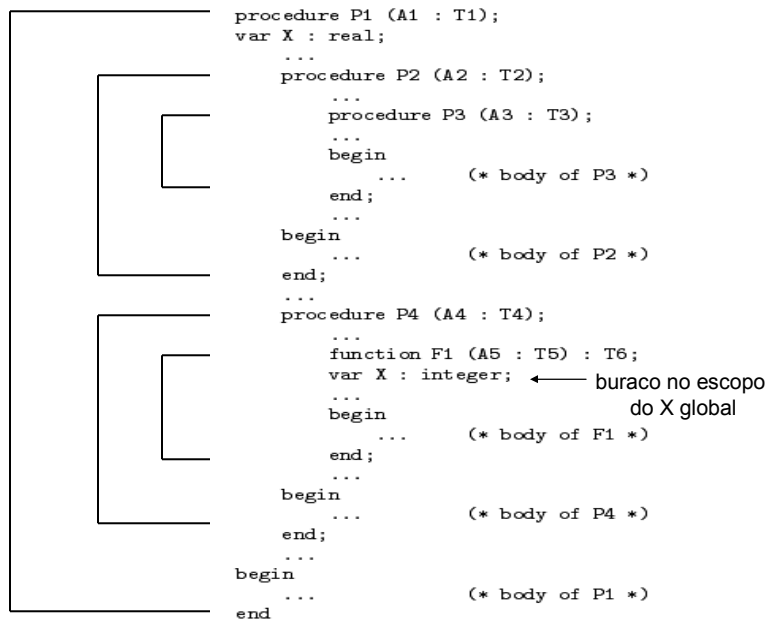
Ligação - Exemplos

- Nas primeiras versões de Fortran o tipo de uma variável era definido implicitamente pela sua sintaxe. A ligacão entre nome e tipo era estabelecida portanto na implementação.
- Em Pascal todas as variáveis devem ser declaradas no início de cada bloco. A ligação ocorre em tempo de compilação.
- Em Lua a atribuição de um valor a uma variável funciona como uma declaração. O tipo da variável é determinado pelo valor que lhe foi atribuído, e pode ser modificado durante a execução do programa. A ligação ocorre portanto em tempo de execução, e a "tipagem" é dinâmica.

Informática PUC-Rio

Conceito de Escopo

- O escopo é a região textual de um programa em que um objeto possui uma ligação ativa.
- Um escopo usualmente é aberto através da declaração de uma rotina ou bloco, e é fechado ao final dos mesmos.
- Podemos nos referir tanto ao escopo de um programa quanto ao escopo associado a um objeto.
- A redeclaração de uma variável em uma rotina cria um *buraco* (hole) no escopo desta variável.



Escopo Estático X Dinâmico

- Em LPs com escopo estático (ou léxico), o escopo é determinado através da estrutura textual do programa.
- Em LPs com escopo dinâmico, o escopo é determinado através da linha de execução do programa, sendo dependente portanto da ordem de execução das rotinas.

Informática PUC-Rio

Escopo Dinâmico - Exemplo

```
x: integer
procedure print_x()
begin
  print(x);
end
procedure p2
x: integer;
begin
  x= 4;
  print_x();
end
begin
  x = 3;
  p2();
end
```

- Se o escopo for dinâmico o programa imprime 4
- Se o escopo for estático, o programa imprime 3

Informática PUC-Rio

Módulos

- Diversas LPs permitem que o código de uma aplicação seja dividido em estruturas denominadas módulos (C++, Modula 2, Clu, Ada, etc).
- Módulos permitem encapsular as construções do programa, definindo *espaços de nome* (namespaces) independentes.
- A visibilidade entre módulos geralmente é controlada explicitamente através de declarações específicas.

```
CONST stack_size = ...
TYPE element = ...
...
MODULE stack;
IMPORT element, stack_size;
EXPORT push, pop;
TYPE
    stack_index = [1..stack_size];
VAR
    s   : ARRAY stack_index OF element;
    top : stack_index;        (* first unused slot *)

PROCEDURE error; ...

PROCEDURE push (elem : element);
BEGIN
    IF top = stack_size THEN
        error;
    ELSE
        s[top] := elem;
        top := top + 1;
    END;
END push;

PROCEDURE pop () : element;   (* A Modula-2 function is just a *)
BEGIN                         (* procedure with a return type. *)
    IF top = 1 THEN
        error;
    ELSE
        top := top - 1;
        RETURN s[top];
    END;
END pop;

BEGIN
    top := 1;
END stack;
```

```
VAR x, y : element;
...
push (x);
...
y := pop;
```

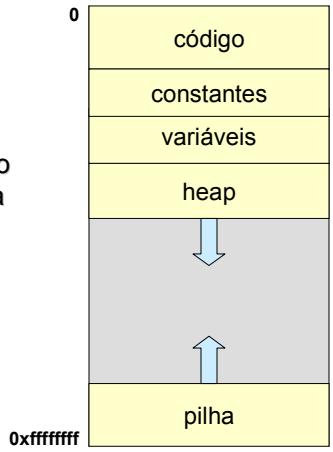

Ambiente de Execução

- A organização do ambiente de execução de um programa depende das regras de escopo e ligação da LP, sobretudo das seguintes características:
  - Suporte a recursividade;
  - Suporte a alocação dinâmica de memória;
  - Escopo dinâmico ou estático.

Informática PUC-Rio

Ambiente de Execução

Layout Típico da Memória



Informática PUC-Rio

Alocação de Objetos na Memória

- A alocação de objetos na memória segue diferentes estratégias:
  - Estática – o objeto possui um endereço absoluto que é mantido durante toda a execução do programa. A alocação de variáveis globais sempre é feita de forma estática.
  - Pilha – os objetos são alocados seguindo uma estratégia LIFO de acordo com a invocação de subrotinas. Em LPs com suporte a recursividade as variáveis locais são alocadas na pilha.
  - Heap – os objetos são alocados na memória de forma dinâmica, conforme demandas específicas do programa. LPs com suporte a alocação dinâmica utilizam sempre a heap para este fim.

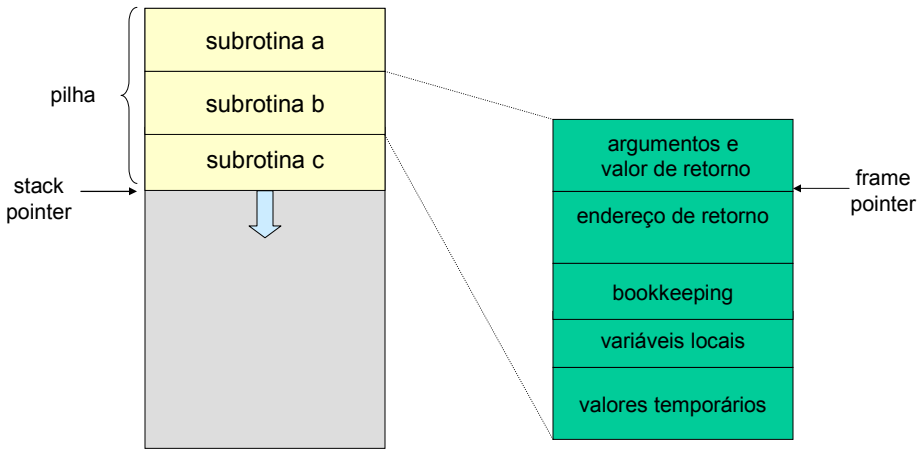
Informática PUC-Rio

Rotinas

- Rotinas são representadas em tempo de execução por um trecho de código e por uma estrutura de dados denominada registro de ativação.
- O registro de ativação contém todas as informações dinâmicas necessárias para a execução da rotina, incluíndo os argumentos, variáveis locais e endereço de retorno.
- O contexto de uma rotina (referencing environment) consiste das ligações ativas correspondentes a todas as variáveis locais e não locais.

Informática PUC-Rio

Registro de Ativação



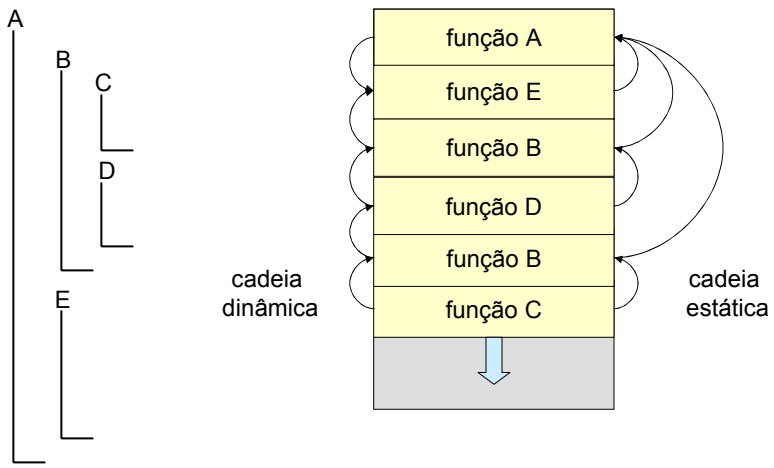
Informática PUC-Rio

Cadeias Dinâmicas e Estáticas

- Os registros de ativação contidos na pilha de um programa formam duas cadeias de referências.
- A cadeia dinâmica representa a sequência de chamadas das subrotinas em tempo de execução, indicando de forma ordenada o endereço do código de retorno de cada subrotina que fez uma invocação ainda ativa.
- A cadeia estática indica o aninhamento de escopos associados às diferentes subrotinas.

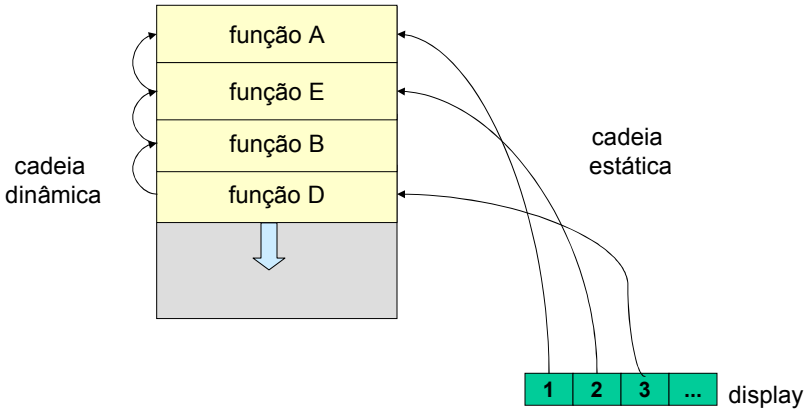
Informática PUC-Rio

Cadeias Dinâmicas e Estáticas



Informática PUC-Rio

Cadeias Dinâmicas e Estáticas



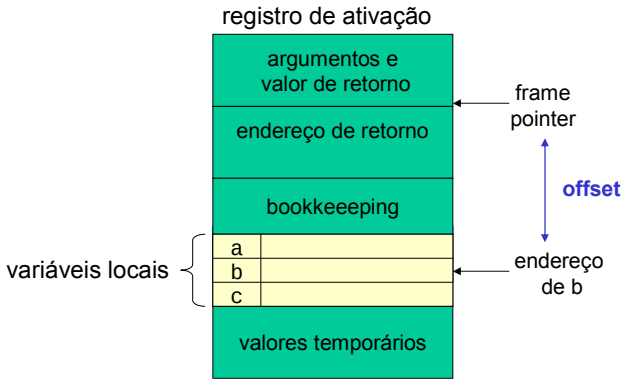
Informática PUC-Rio

Como encontrar o endereço de uma variável estática?

- Variáveis globais com tamanho definido possuem endereço fixo.
- Variáveis locais com tamanho definido possuem um offset fixo em relação ao frame pointer.
- Para variáveis globais com tamanho variável, o endereço estático contém uma referência para outra área da heap.
- Para variáveis locais com tamanho variável, o offset contém uma referência para outra área do registro de ativação ou da heap.

Informática PUC-Rio

Como encontrar o endereço de uma variável estática?



Informática PUC-Rio

Como encontrar o endereço de uma variável estática?

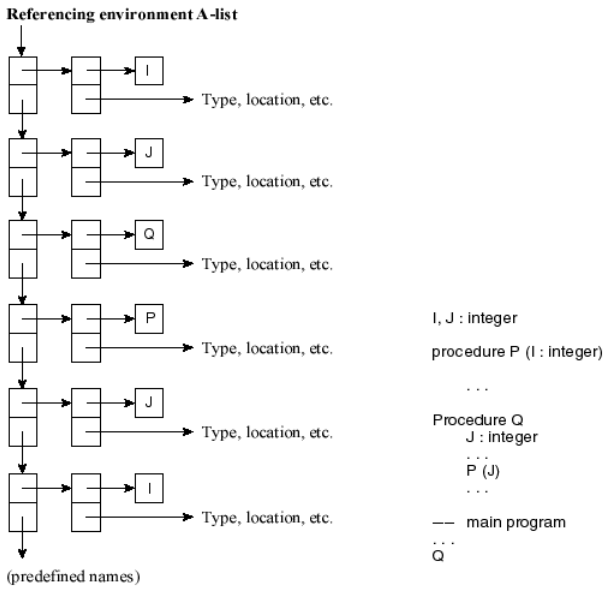
- Em LPs com escopo estático o valor de variáveis não locais pode ser encontrado através da cadeia estática e do respectivo offset.
- Em LPs com escopo dinâmico é necessário a utilização de uma estrutura de dados especial para encontrar o valor da variável, como por exemplo uma lista de associação (a-list).

Informática PUC-Rio

A-List

- Uma a-list é uma pilha que contém entradas para todos os nomes válidos em um escopo, indicando o respectivo tipo, valor, etc.
- Cada vez que um escopo é aberto, o compilador insere entradas para todas os nomes locais.
- Ao fechar um escopo, todas as entradas são retiradas da pilha.
- Para encontrar uma variável basta percorrer a pilha até encontrar a primeira entrada válida, que pode estar portanto em qualquer escopo.

Informática PUC-Rio



Informática PUC-Rio

Sobrecarga

- Quando um nome é usado para referir-se a mais de um objeto dentro de um mesmo escopo dizemos que este nome é sobrecarregado.
- Em algumas LPs o símbolo **+** é usado para representar a adição de inteiros, adição de reais e concatenação de strings.
- A sobrecarga é bastante empregada na definição de funções que utilizam parâmetros diferentes.

Informática PUC-Rio

Exemplo – C++

```
struct complex {
    double real, imaginary;
};
enum base {dec, bin, oct, hex};

int i;
complex x;

void print_num (int n) ...
void print_num (int n, base b) ...
void print_num (complex c) ...

print_num (i);      // uses the first function above
print_num (i, hex); // uses the second function above
print_num (x);      // uses the third function above
```

Informática PUC-Rio

Passagem de Parâmetros

- Por valor – a unidade invocado recebe apenas o valor do parâmetro que é armazenado como se fosse uma variável local.
- Por referência – a unidade invocada recebe o endereço no qual está armazenado o valor a ser usado como parâmetro. A variável torna-se "compartilhada".
- Por nome – o parâmetro é avaliado no contexto da unidade que fez a invocação toda fez que é usado pela unidade invocada.

Informática PUC-Rio

Passagem Por Referência e Por Valor

- Em Pascal os valores são passados por referência ou valor dependendo da definição da rotina. A utilização de *var* indica que o argumento deve ser passado por referência.
- Em Java todos os objetos são sempre passados por referência.
- Em C os parâmetros são sempre passados por valor. Pode-se passar um endereço através do operador &.

Informática PUC-Rio

Passagem Por Nome Exemplo

```
void swap(int a, b){
  int temp;
  temp = a;
  a = b;
  b = temp;
}
swap (i, a[i])
temp = i
i = a[i]
a[i] = temp
```


Informática PUC-Rio

Valores de Primeira Ordem

- Um valor de primeira ordem é aquele que pode ser passado como parâmetro, como valor de retorno de uma subrotina ou atribuído a uma variável.
- Um valor de segunda ordem pode apenas ser passado como parâmetro de uma subrotina.
- Nos demais casos o valor é classificado como de terceira ordem.

Informática PUC-Rio

Funções como Valores de Primeira Ordem

- Várias LPs suportam funções como valores de primeira ordem (Pascal, Fortran, Lua, etc)
- Normalmente, apenas os endereços dos procedimentos são armazenados em variáveis de tipo procedimento, e isso é suficiente para sua ativação através da chamada da variável.
- Em algumas LPs, como C, o fato de a variável conter um ponteiro para um procedimento é tornado explícito pela sintaxe.

Informática PUC-Rio

Exemplo - Pascal

```
procedure repita(n:integer; procedure q);
begin
  for i:=1 to n do
  q;
end;
....
procedure p;
begin
...
end;
....
repita(5, p); -- repita 5 vezes a ação p
....
```

Informática PUC-Rio

Exemplo - C

```
int (*v) (int); -- v declarada como
                   ponteiro para função
...
int f (int x) {
...
} -- função f
int i;
...
v=f; -- atribuição
...
i=(*v)(5); -- chamada (do conteúdo) de v
```


Informática PUC-Rio

Deep Binding X Shallow Binding

- O contexto é determinado pelas regras de escopo.
- Em LPs em que rotinas podem ser passadas como parâmetros é importante definir em que contexto estas rotinas irão ser executadas.
- Existem duas opções:
  - Quando a rotina é invocada (shallow binding ou ligação superficial);
  - Quando a referência para a rotina é criada (deep binding ou ligação profunda).

```
type person = record
        . . .
        age : integer
        . . .
threshold : integer
people : database

function older_than (p : person) : boolean
        return p.age ≥ threshold

procedure print_person (p : person)
        -- Call appropriate I/O routines to print record on standard
        -- output. Make use of non-local variable line_length to format
        -- data in columns.
        . . .

procedure print_selected_records (
                db : database
                predicate, print_routine : procedure)
        line_length : integer

        if device_type (stdout) = terminal
                line_length := 80
        else        -- Standard output is a file or printer.
                line_length := 132
        foreach record r in db
                -- Iterating over these may actually be
                -- a lot more complicated than a 'for' loop.
                if predicate (r)
                        print_routine (r)

-- main program
. . .
threshold := 35
print_selected_records (people, older_than, print_person)
```


Closure

- A representação explícita do contexto, que pode ser repassada juntamente com uma referência para a rotina, é denominado *closure (fechamento)*.
- Closures só fazem sentido em LPs que implementam deep binding.
- Em LPs com escopo dinâmica e que usam uma A-list, o contexto pode ser representado através de um ponteiro para o topo da pilha.