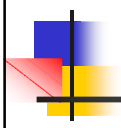
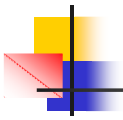


Pilhas

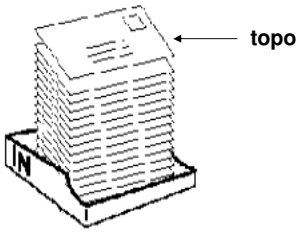
Prof. Rui Jorge Tramontin Jr.



Índice

Introdução

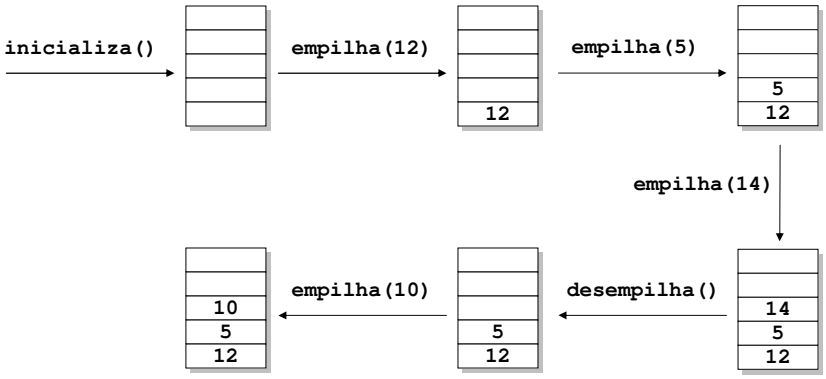
- **Pilha** é uma lista na qual todas as inserções e exclusões são feitas em um único lado, chamado **topo**.
- O último elemento inserido será o primeiro a ser removido.
  - Pilhas são também chamadas de listas **LIFO** (Last In First Out).



Operações de uma pilha

- **`inicializa_pilha(P)`** : criar uma pilha P vazia.
- **`empilha(item, P)`** : insere item no topo de P.
- **`desempilha(P)`** : elimina o elemento do topo de P.
- **`topo(P)`** : acessa o elemento do topo da pilha (sem eliminar).
- **`estah_vazia(P)`** : testa se P está vazia.
- **`esta_cheia(P)`** : testa se P está cheia.

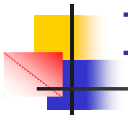
Exemplo



Modelagem da Pilha

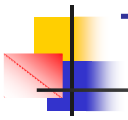
- Aspecto Estrutural:
  - Vetor para armazenar as informações.
  - Indicador da posição atual do topo da pilha.
  - Constantes que indiquem quando a pilha está cheia e para codificar situações de erros.
- Pseudo-código:

```
constantes MaxPilha        100;
           ErroPilhaVazia  -1;   // Underflow
           ErroPilhaCheia  -2;   // Overflow

tipo Pilha {
   inteiro   dados[MaxPilha];
   inteiro   topo;
};
```



Inicialização da Pilha

```
procedimento inicializa_pilha(var p: Pilha)

início
   p.topo <- -1;
fim
```

Testa se Pilha está vazia

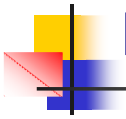
```
função estah_vazia(p: Pilha): booleano

início
   se p.topo = -1 então
      retorne VERDADEIRO
   senão
      retorne FALSO
   fimse
fim
```

Testa se Pilha está cheia

```
função estah_cheia(p: Pilha): booleano

início
   se p.topo = MaxPilha - 1 então
      retorne VERDADEIRO
   senão
      retorne FALSO
   fimse
fim
```

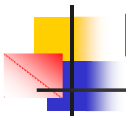

Empilha

```
função empilha(var p: Pilha; item: inteiro): inteiro

início
   se estah_cheia(p) então
      retorne ErroPilhaCheia
   senão
      p.topo <- p.topo + 1
      p.dados[p.topo] <- item
      retorne p.topo
   fimse
fim
```



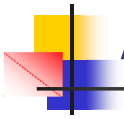
Desempilha

```
função desempilha(var p: Pilha): inteiro

início
   se estah_vazia(p) então
      retorne ErroPilhaVazia
   senão
      p.topo <- p.topo - 1
      retorne p.dados[p.topo + 1]
   fimse
fim
```

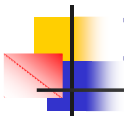
Retorna elemento no topo

```
função topo(p: Pilha): inteiro

início
   se estah_vazia(p) então
      retorne ErroPilhaVazia
   senão
      retorne p.dados[p.topo]
   fimse
fim
```



Aplicações

- Inversão de listas
- Chamada de funções
  - Pilha do Sistema Operacional
- Avaliação de expressões aritméticas
  - Notação Polonesa Reversa (*pós-fixada*)



Inversão de listas

- Pilhas podem ser usadas para inversão de listas
- Por exemplo, uma palavra (lista de caracteres) pode ser invertida da seguinte maneira:
  - Empilhe cada letra da palavra;
  - Desempilhe e adicione as letras numa outra palavra.

`palavra → arvalap`

Chamada de Funções

- Sistema operacional mantém uma pilha utilizada sempre que uma função é chamada.
- Serve para que o processador possa continuar o fluxo de execução:
  - Tal pilha armazena o endereço da próxima instrução após a chamada;
  - Os parâmetros da função e variáveis locais também são empilhados.

Chamada de Funções

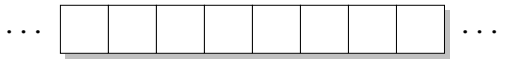
- Situação inicial: `a()` está executando.

```
função a()
início
    ...
    b();
e1: ...
fim
```

```
função b()
início
    ...
    c();
e2: ...
fim
```

```
função c()
início
    ...
    d();
e3: ...
fim
```

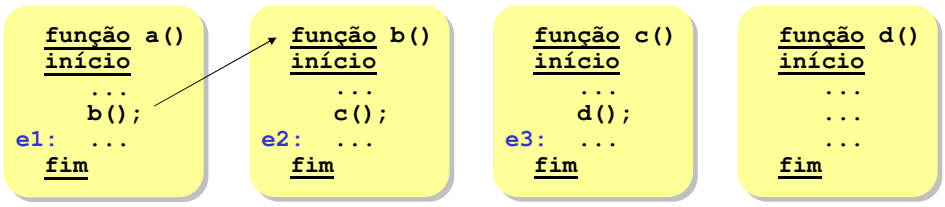
```
função d()
início
    ...
    ...
    ...
fim
```

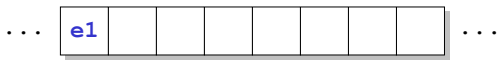


Pilha do Sistema

Chamada de Funções

- **a()** chama **b()**.

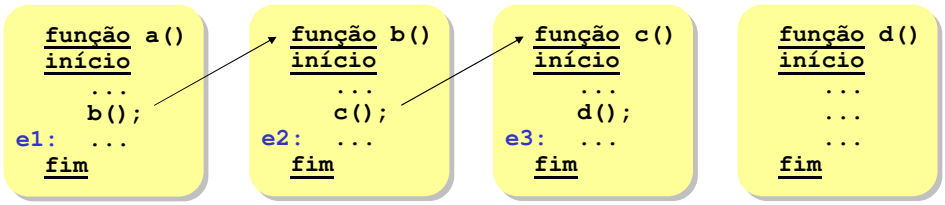


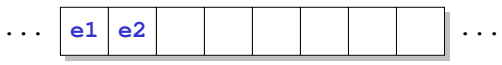


Pilha do Sistema

Chamada de Funções

- **b()** chama **c()**.

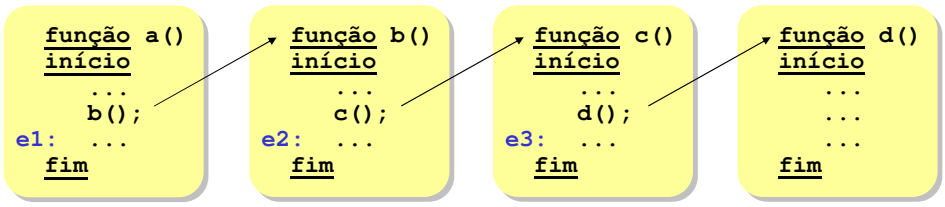


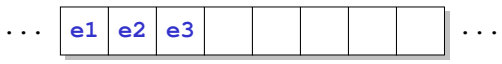


Pilha do Sistema

Chamada de Funções

- `c()` chama `d()`.

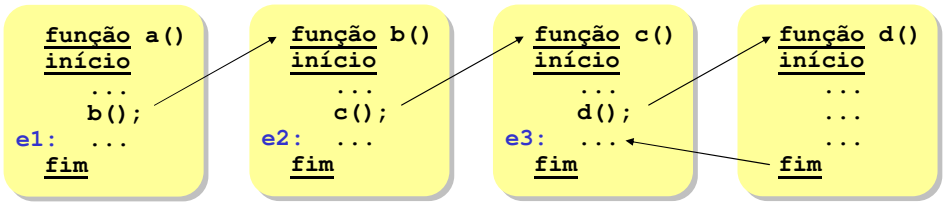


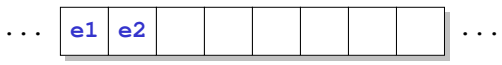


Pilha do Sistema

Chamada de Funções

- Retorno de `d()`.

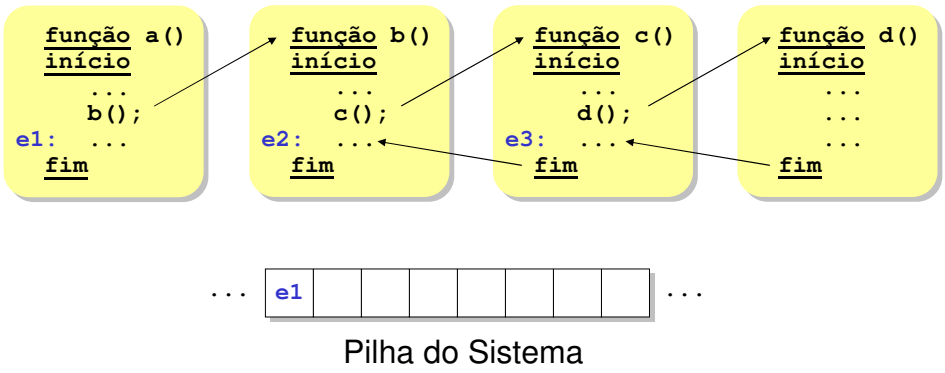




Pilha do Sistema

Chamada de Funções

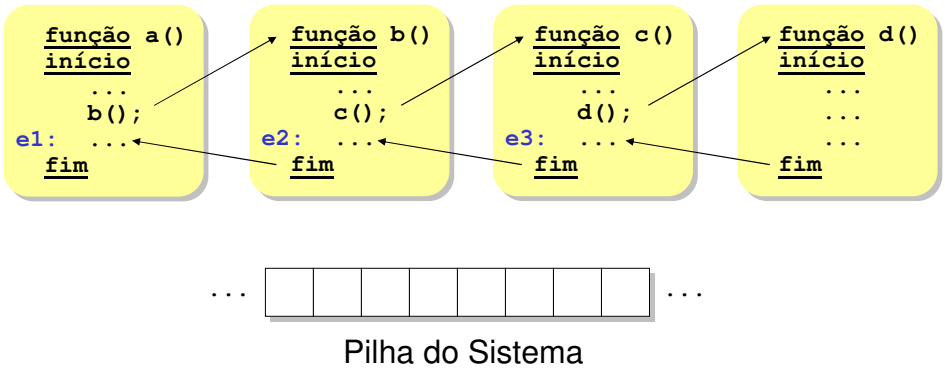
- Retorno de **c()**.



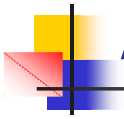
Pilha do Sistema

Chamada de Funções

- Retorno de **b()**.

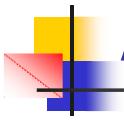


Pilha do Sistema



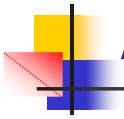
Avaliação de Expressões

- Expressões aritméticas são normalmente escritas em diferentas notações
  - *infixada*: A+B*C
  - *pré-fixada*: +A*BC
  - *pós-fixada*: ABC*+
- Pilhas podem ser usadas para a avaliação de expressões em notação *pós-fixada*.
  - A+B → AB+
  - A*C → AC*
  - A+B*C → ABC*+
  - (A+B)*C → AB+C*



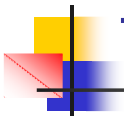
Avaliação de Expressões

- Dada uma expressão em notação pós-fixada e uma pilha, a avaliação pode ser feita da seguinte maneira:
  - Leia a expressão da esquerda para a direita;
  - Quando um valor (operando) é encontrado, empilhe-o;
  - Quando um operador é encontrado:
    - Desempilhe duas vezes (os operandos) e aplique o operador;
    - O resultado da operação é empilhado.
  - Ao final da leitura expressão, basta desempilhar o valor da pilha → resultado da expressão.



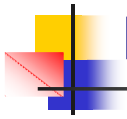
Avaliação de Expressões

- **Exemplo**:
  - Infixada: (1 + 3 - 7) * 4
  - Pós-fixada: 7 1 3 + - 4 *



TAD da Pilha em C (Pilha.h)

```
// definição de constantes
#define  MaxPilha          100
#define  ErroPilhaVazia    -1
#define  ErroPilhaCheia    -2

// define estrutura e tipo Pilha
typedef struct {
   int dados[MaxPilha];
   int topo;
} Pilha;

// cabeçalho das funções
```



Exercício

- Definir o TAD para pilhas (Pilha.h)
  - falta definir o protótipo das funções;
- Implementar as operações do TAD:
  - Pilha.c;
- Escrever um programa em C que utilize o TAD Pilha.