Centro Universitário de Araraquara – UNIARA
Departamento de Ciências da Administração e Tecnologia

Curso de Engenharia de Computação

Estrutura de Dados
Pilha

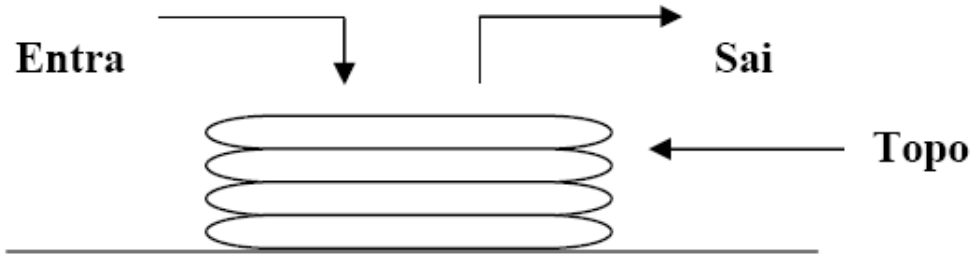
Prof. Msc. José Eduardo Ribeiro

Introdução

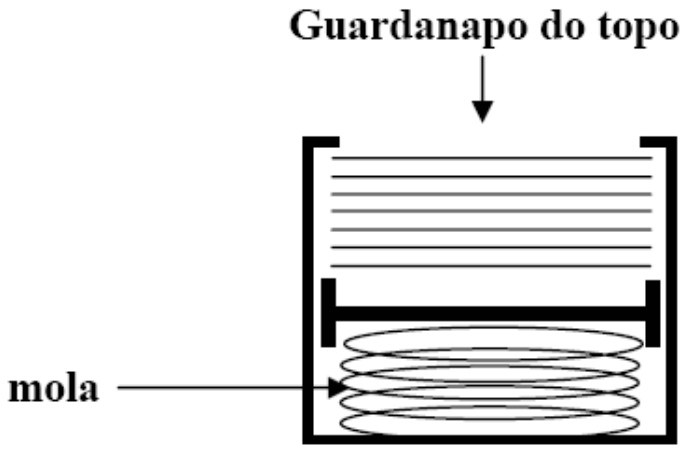
- Uma das estruturas de dados mais simples é a pilha.
- Possivelmente por essa razão, é a estrutura de dados mais utilizada em programação, sendo inclusive implementada diretamente pelo hardware da maioria das máquinas modernas.

Pilha

- Analogia
  - Pilha de Pratos
    - Se quisermos adicionar um prato na pilha, o colocamos no topo. Para pegar um prato da pilha, retiramos o do topo. Assim, temos que retirar o prato do topo para ter acesso ao próximo prato.
    - Cada novo elemento é inserido no topo e só temos acesso ao elemento do topo da pilha.

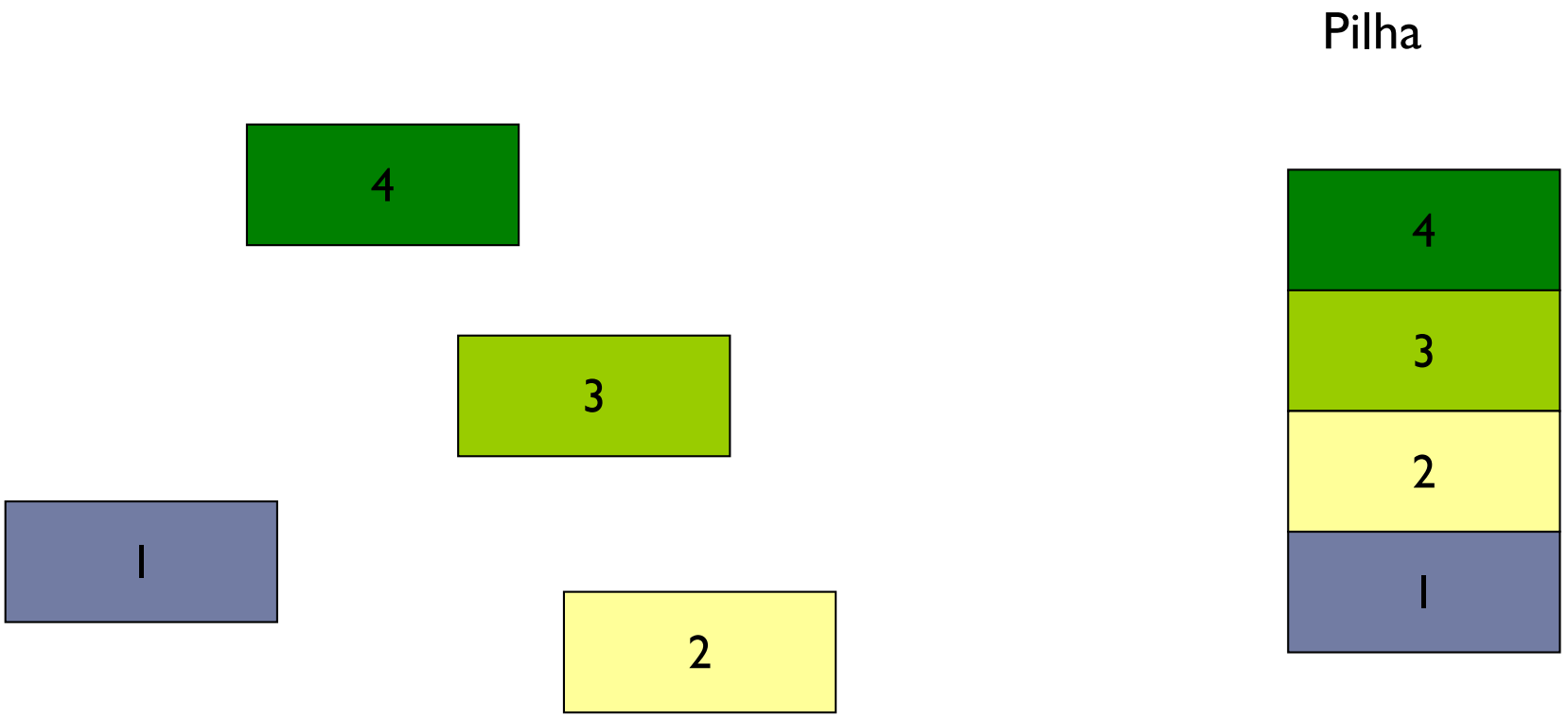


Uma pilha de pratos

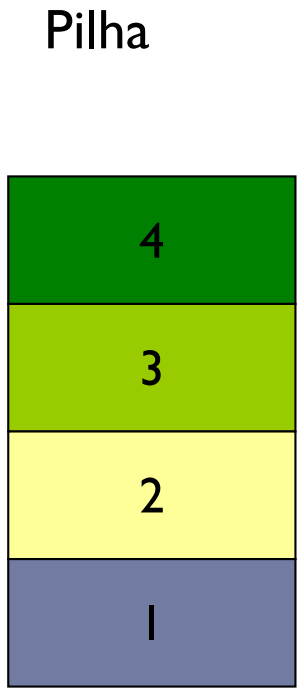


Um porta-guardanapos

O que é uma pilha?



O que é uma pilha?



Pilhas

- Uma pilha é uma **estrutura de dados** que pode ser acessada somente por uma de suas extremidades para **armazenar** e **recuperar** dados.
- A idéia fundamental da pilha é que todo o acesso a seus elementos é feito através do seu ***topo***.
  - Assim, quando um elemento novo é introduzido na pilha, passa a ser o elemento do topo, e o único elemento que pode ser removido da pilha é o do topo.
  - *Isto faz com que os elementos da pilha sejam retirados na ordem inversa à ordem em que foram introduzidos.*
- A sigla LIFO – last in, first out – é usada para descrever esta estratégia
  - O primeiro que sai é o último que entrou.

Pilha

- Usos:
  - Chamada de subprogramas.
    - Uso em funções em C.
  - Avaliação de expressões aritméticas, etc.

Pilha

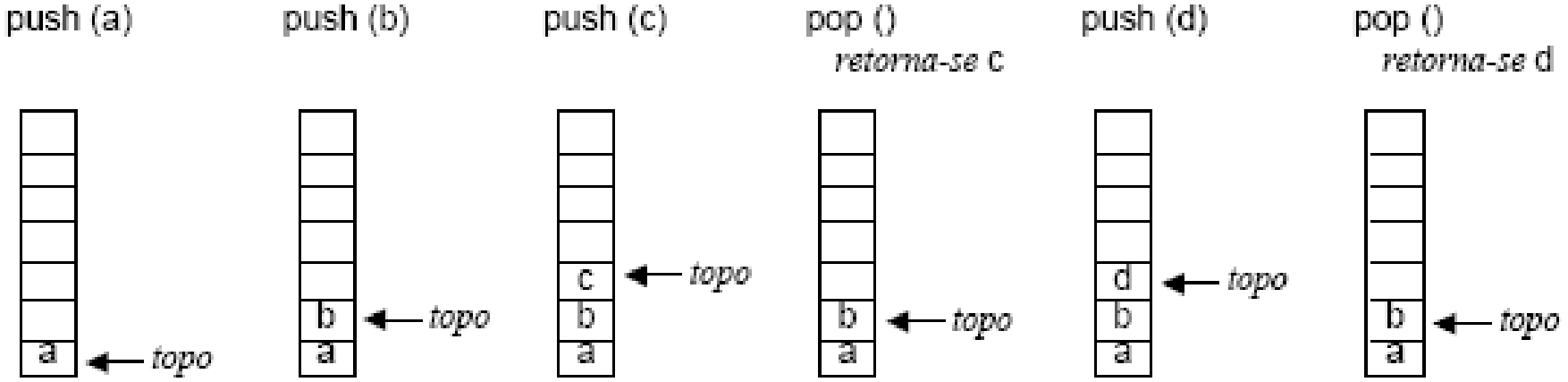
- Suporta três operações básicas:
  - Top (topo) – acessa o elemento do topo.
  - Push (empurre) – insere um elemento no topo.
  - Pop (salte) – remove um elemento do topo.

Pilha

- Limites da Pilha:
  - No mundo real uma pilha é limitada pelo chão e teto.
    - Não é possível inserir elementos infinitamente.
    - Não é possível remover elementos infinitamente.
  - No computador os limites da pilha se refere a quantidade de memória usada para representá-la.

Pilha

- Operações básicas:
  - Empilhar um novo elemento, inserindo-o no topo.
    - Push(empilhar)
  - Desempilhar um elemento, removendo-o do topo.
    - Pop(desempilhar)



Pilha

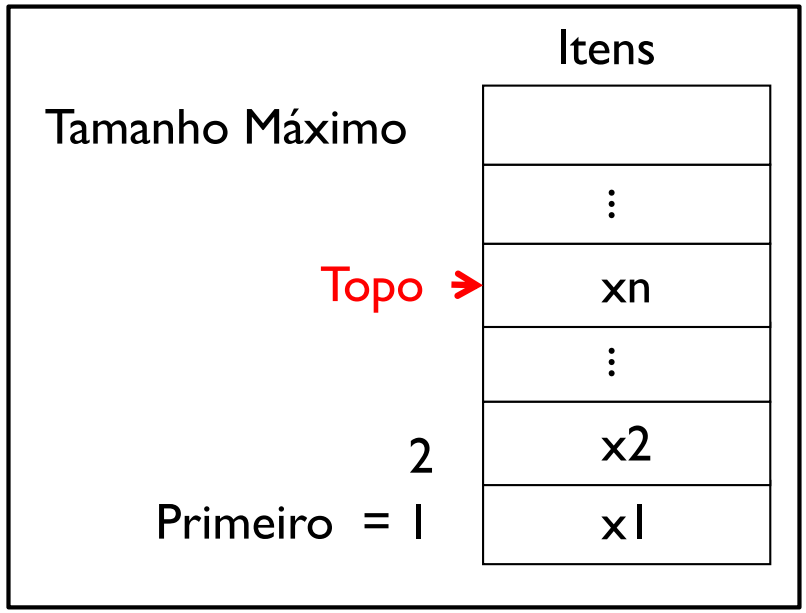
- O exemplo de utilização de pilha mais próximo é a própria pilha de execução da linguagem C.
- As variáveis locais das funções são dispostas numa pilha e uma função só tem acesso às variáveis que estão no topo (não é possível acessar as variáveis da função local às outras funções).
- Existem várias opções de estruturas de dados que podem ser usadas para representar pilhas.
  - As duas representações mais utilizadas são as implementações por meio de **arranjos** e de **apontadores**.

Implementação de pilha com vetor

- Em aplicações computacionais que precisam de uma estrutura de pilha, é comum sabermos de antemão o número máximo de elementos que podem estar armazenados simultaneamente na pilha, isto é, a estrutura da pilha tem um limite conhecido.
- Nesses casos, a implementação da pilha pode ser feita usando um vetor.
- Devemos ter um vetor para armazenar os elementos da pilha. Os elementos inseridos ocupam as primeiras posições do vetor. Desta forma, se temos n elementos armazenados na pilha, o elemento vet[n-1] representa o elemento do topo.

Implementação de pilha com vetor

- Os itens da pilha são armazenados em posições contíguas de memória.
- Como as inserções e as retiradas ocorrem no topo da pilha, uma variável chamado Topo é utilizado para controlar a posição do item no topo da pilha.



- Exemplo: Projeto(PilhaEstaticaSimplesV1).c

Alocação de Memória Dinâmica

Pilha com Alocação Dinâmica de Memória

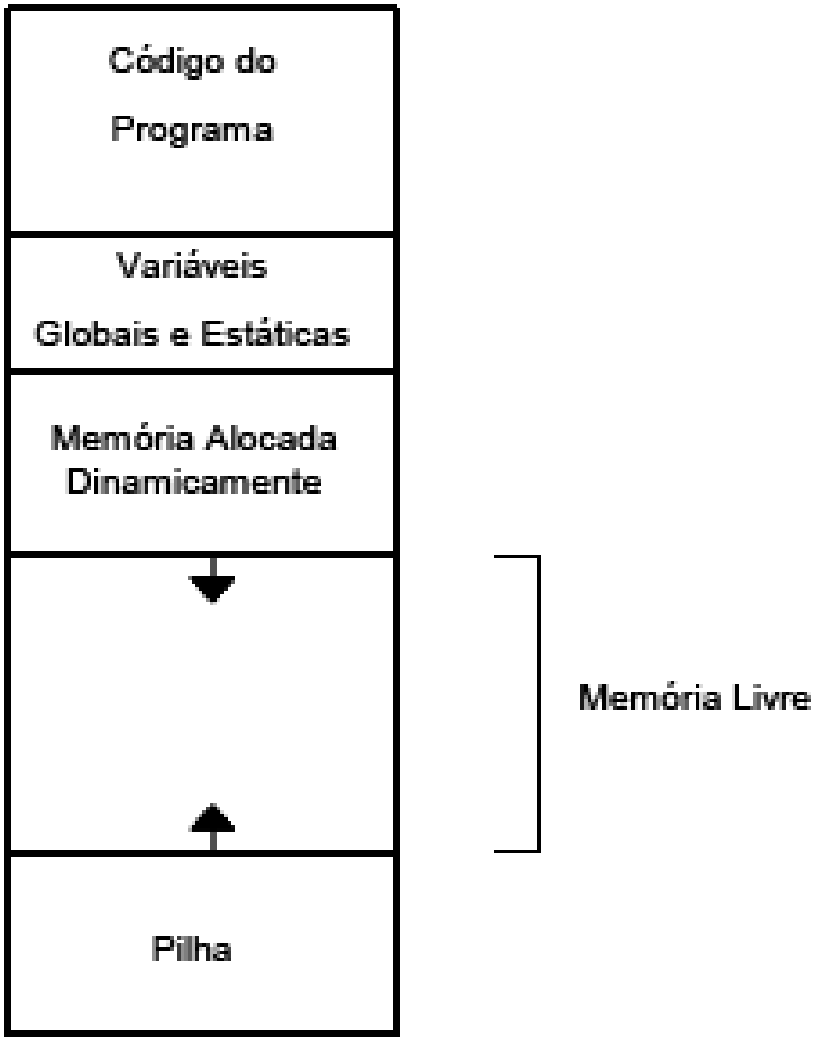
Pilha com Alocação Dinâmica de Memória

- Informalmente, podemos dizer que existem três maneiras de reservarmos espaço de memória para o armazenamento de dados.
  - Variáveis globais e estáticas
    - O espaço reservado para uma variável global existe enquanto o programa estiver sendo executado.
  - Variáveis locais
    - O espaço reservado existe apenas enquanto a função que declarou a variável está sendo executada, sendo liberado para outros usos quando a execução da função termina. Por este motivo, a função que chama não pode fazer referência ao espaço local da função chamada.
  - Requisitar ao sistema, em tempo de execução, a alocação de memória dinamicamente.

Pilha com Alocação Dinâmica de Memória

- Alocação de memória estática x Alocação de memória dinâmica.
  - Como podemos reservar espaço de memória para armazenar informações em tempo de execução?
    - Manipulações
      - 1° - Alocando memória de uma única vez e desalojando de uma única vez.
        - Necessita de uma função para inicializar a estrutura de dados.
      - 2° - Alocando memória de acordo com a necessidade de armazenamento e manipulação de dados e desalojando de acordo com a necessidade.
        - Não necessita de uma função para inicializar a estrutura de dados.
        - A alocação é feita antes de inserir o dado.

Exemplo ilustra a pilha de memória



Alocação de Memória Dinâmica

- Biblioteca padrão <stdlib.h>
  - malloc()
    - Função que recebe como parâmetro o número de bytes que se deseja alocar e retorna o endereço inicial da área de memória alocada dinamicamente.
  - Vamos considerar a alocação dinâmica de um vetor de inteiros com 10 elementos. Como a função malloc retorna o endereço da área alocada e, neste exemplo, desejamos armazenar valores inteiros nessa área, devemos declarar um ponteiro de inteiro para receber o endereço inicial do espaço alocado. O trecho de código então seria:
    - int *v;
    - v = malloc(10*4);

Alocação de Memória Dinâmica

- Podemos, então, tratar v como tratamos um vetor declarado estaticamente, pois, se v aponta para o inicio da área alocada, podemos dizer que v[0] acessa o espaço para o primeiro elemento que armazenaremos, v[1] acessa o segundo, e assim por diante (até v[9]).
- Para ficarmos independentes de compiladores e arquiteturas, usamos o operador sizeof( ).
  - v = malloc(10*sizeof(int));

Alocação de Memória Dinâmica

- A função malloc é usada para alocar espaço de memória para armazenarmos valores de ***qualquer tipo***.
  - A função malloc retorna um ponteiro genérico, para um tipo qualquer representado por void, então, fazemos a conversão explicitamente, utilizando o operador de molde de tipo (cast).
  - O comando para a alocação do vetor de inteiros fica então:
    - v = (int *) malloc(10*sizeof(int));

Alocação de Memória Dinâmica

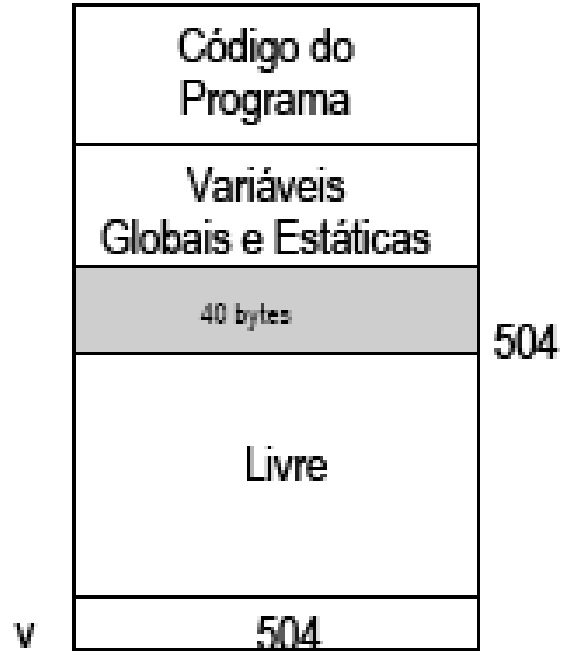
v = (int *) malloc(10*sizeof(int));

1 - Declaração: int *v

Abre-se espaço na pilha para o ponteiro (variável local)

2 - Comando: v = (int *) malloc (10*sizeof(int))

Reserva espaço de memória da área livre e atribui endereço à variável



Alocação de Memória Dinâmica

- Se, porventura, não houver espaço livre suficiente para realizar a alocação, a função retorna um endereço nulo (representado pelo símbolo NULL, definido em <stdlib.h>).
- Podemos cercar o erro na alocação do programa verificando o valor de retorno da função malloc. Por exemplo, podemos imprimir uma mensagem e abortar o programa com a função exit, também definida na <stdlib.h>.
  - …
  - v = (int*) malloc(10*sizeof(int));
  - if (v==NULL) {
    - printf("Memoria insuficiente.\n");
    - exit(1); /* aborta o programa */
  - }
  - …

Pilha (Alocação de Memória Dinâmica)

- Liberação do espaço de memória alocado dinamicamente.
  - free();

- Esta função recebe como parâmetro o ponteiro que corresponde a área de memória a ser liberada.
- No exemplo anterior:
  - free (v);

- Só podemos passar para a função *free()* um endereço de memória que tenha sido alocado dinamicamente. Devemos lembrar ainda que não podemos acessar o espaço de memória depois que foi liberada com a função *free()*.

Pilha com Alocação Dinâmica de Memória

- Alocação de memória estática x Alocação de memória dinâmica
  - Como podemos reservar espaço de memória para armazenar informações em tempo de execução?
    - Manipulações
      - 2° - Alocando memória de acordo com a necessidade de armazenamento e manipulação de dados e desalojando de acordo com a necessidade
        - Não necessita de uma função para inicializar a estrutura de dados
        - A alocação é feita antes de inserir o dado

Interface do tipo pilha com struct

- Alocação de memória dinâmica (função push)

- A função cria referencia um ponto de acesso da pilha na memória do computador
- As funções push e pop inserem e retiram, respectivamente, um valor real na pilha
  - A função push trata também a alocação dinâmica de memória
  - A função pop trata também o desalojamento de memória dinâmica
- A função vazia informa se a pilha está ou não vazia

- Exemplo: Projeto(PilhaDinamicaVersao2).dev