Estrutura de Dados

Prof. Msc. Matheus R. D. Ullmann

matheusullmannifba@gmail.com
Instituto Federal da Bahia - IFBA
Campus Barreiras

February 22, 2024

Sumário

INSTITUTO FEDERAL
Bahia

Apresentação da Disciplina

O professor

- Professor Matheus Rudolfo Diedrich Ullmann
  - Bacharel em Ciência da Computação - UFG
  - Mestre em Ciência da Computação - INF/UFG
  - Doutorando em Engenharia Elétrica e de Computação - EMC/UFG

Bibliografia da Disciplina

- TENENBAUM, A.M.; LANGSAM, Y.; AUGENSTEIN, M.J. Estruturas de Dados Usando C. Porto Alegre: Editora Makron Books/Pearson Education, 2005.
- CELES, W.; CERQUEIRA, R. RANGEL, J. L. Introdução à estruturas de dados. São Paulo: Editora Campus Elsevier, 2004.
- SEDGEWICK, R. and WAYNE, K. Algorithms, 4th Edition, Editora Addison-Wesley, 2011.

INSTITUTO FEDERAL
Bahia

Bibliografia Complementar da Disciplina

- LOUDON, K. Dominando Algoritmos com C, Ed. Ciência Moderna, Rio de Janeiro, 2000.
- EDELWEISS, N.; GALANTE, R., Estrutura de Dados. Porto Alegre: Bookman, 2009. (Série Livros Didáticos Informática UFRGS).
- LAFORE, R. Aprenda em 24 horas: estrutura de dados e algoritmos. Rio de Janeiro: Editora Campus, 1999.

INSTITUTO FEDERAL
Bahia

Ementa

- Análise e projeto dos tipos de dados abstratos, estruturas de dados e suas aplicações: listas lineares, pilhas, filas.
- Identificar e implementar métodos e técnicas de classificação de dados.
- Conhecer as principais técnicas de programação envolvendo estruturas de dados em Linguagem de alto nível.
- Definir formalmente as estruturas de dados, manipular estas estruturas, selecioná-las para suas aplicações e analisar métodos de pesquisa, ordenação, representação de dados.

INSTITUTO FEDERAL
Bahia

Programa do Curso

- Conceitos Iniciais
  - Introdução à linguagem de programação C;
    - Tipos primitivos de dados;
    - Vetores;
    - Matrizes;
    - Estruturas (structs);
  - Tipos abstratos de dados (TADs);
  - Representação e implementação de TDA;
- Recursividade
  - Definição;
  - Exemplos;
  - Simulação;
  - Implementação de recursividade.

INSTITUTO FEDERAL
Bahia

Programa do Curso

- Visão geral de estruturas e listas lineares
  - Conceituar Estrutura de Dados;
  - Descrever os tipos de Estruturas de Dados;
  - Implementar operações básicas da Estrutura de Dados Lista;
    - Inserção;
    - Troca;
    - Seleção;
    - Distribuição e intercalação;
    - Comparação entre os métodos;
- Listas Lineares
  - Definição;
  - Estruturas estáticas e dinâmicas;
  - Operações básicas em listas de elementos.

INSTITUTO FEDERAL
Bahia

Programa do Curso

- Pilhas
  - Definição do tipo abstrato, aplicações e exemplos;
  - Operações básicas em uma pilha;
  - Implementações de pilhas;
- Filas
  - Definição do tipo abstrato, aplicações e exemplos;
  - Operações básicas em uma fila;
  - Filas circulares;
  - Implementações de filas.

INSTITUTO FEDERAL
Bahia

Programa do Curso

- Listas Ligadas
  - Pilhas ligadas;
  - Filas lidadas;
  - Listas ligadas;
  - Listas duplamente ligadas;
  - Implementação.

Método de Avaliação

- As notas podem variar de 0,0 a 10,0;
- A avaliação será feita por meio de:
  - Prova;
  - Trabalho e/ou seminário;
  - Exercícios e laboratório.

Informações Importantes

- Haverá chamada em todas as aulas e será realizada 10 minutos após o horário de início da aula. Após a chamada só será permitida a entrada em sala até 15 minutos do início da aula;
- Será permitido o uso de notebooks, tablets para fins de educacionais, mas o áudio destes dispositivos deverá estar desligado;
- Não será permitido atender ou efetuar ligações dentro da sala de aula, esta medida visa garantir o respeito aos ambiente de estudo;

INSTITUTO FEDERAL
Bahia

Informações Importantes

- Durante as provas não será permitido o uso de quaisquer dispositivos eletrônicos, ou outros materiais de consulta;
- As provas serão individuais;
- As aulas laboratoriais, provas e entrega de trabalhos e exercícios serão marcadas antecipadamente e divulgadas no Classroom;
- O atraso na entrega dos trabalhos e exercícios assim como a não participação nas aulas laboratoriais serão computados como desconto da nota de exercícios e participação.

INSTITUTO FEDERAL
Bahia

Linguagem de Programação C

Linguagem C - Introdução

- Desenvolvida inicialmente por Dennis M. Ritchie e Ken Thompson em 1972;
- Baseada na linguagem B criada por Thompson, esta linguagem evoluiu da linguagem BCPL, dando origem a duas linguagens anteriores.

INSTITUTO FEDERAL
Bahia

Linguagem C - Introdução

- Projetada inicialmente para ser utilizada no sistema operacional *Unix*;
- C é considerada uma linguagem procedural (forma sequencial de instruções);
- Características principais:
  - Robusta;
  - Multiplataforma;
  - Projetada para aplicações modulares de acesso rápido.

INSTITUTO FEDERAL
Bahia

Linguagem C - Introdução

- C é considerada uma linguagem de médio nível:
  - Possui instruções de alto nível e estruturada, como o *Pascal*;
  - Possui instruções muito próximas da linguagem de máquina, que só o *Assembler* possui;
- Com essa linguagem podemos construir programas organizados e concisos (como o *Pascal*) e ocupando pouco espaço de memória com alta velocidade de execução (como o *Assembler*).

INSTITUTO FEDERAL
Bahia

Linguagem C - Introdução - Características

- Portabilidade entre máquinas e sistemas operacionais;
- Dados compostos em forma estruturada;
- Programas Estruturados;
- Total interação com o Sistema Operacional;
- Código compacto e rápido, quando comparado ao código de outras linguagem de complexidade análoga.

Linguagem C - Introdução - Características

Programa estruturado?

- A programação estruturada (PE) é um paradigma de programação, uma forma de programação de computadores, com ênfase no uso de sub-rotinas, laços de repetição, condicionais e estruturas em bloco.

Linguagem C - Estrutura básica de um programa em C

- Um programa em C consiste em uma ou mais "funções";
- Menor programa em C:

```
main ( ) {

}
```

Linguagem C - Estrutura básica de um programa em C

- Adicionando uma instrução:

```
main ( ){

printf("olá"); /* mostra na tela a mensagem Olá*/

}
```

Linguagem C - Fundamentos em C

- Primeiramente iremos se ater mais na compreensão geral do programa do que na análise detalhada de cada comando ou função utilizada;
- Utilizaremos comandos fundamentais para a escrita de programas básicos e utilizaremos sua sintaxe elementar.

Linguagem C - Fundamentos em C

```
/* Exemplo Idade */

main ( ) {

int idade;

idade = 40;

printf("Sua idade e' %d anos. \n", idade);

}
```

Linguagem C - Fundamentos em C

- Diretiva *#include*
  - Inclui o conteúdo de outro arquivo dentro do programa atual, ou seja, a linha que contém a diretiva é substituída pelo conteúdo do arquivo especificado.

```
#include <nome do arquivo>

ou

#include "nome do arquivo"
```

Linguagem C - Fundamentos em C

- Primeiro modo (com $<$ *nome_do_arquivo* $>$)
  - Utilizado para incluir arquivos que contém declaração das funções na biblioteca padrão;
  - Geralmente possuem a extensão *.h*;
  - Arquivo e descrição:
    - *stdio.h* - Funções de entrada e saída (I/O);
    - *string.h* - Funções de tratamento de strings;
    - *math.h* - Funções matemáticas;
    - *ctype.h* - Funções de teste e tratamento de caracteres;
    - *stdlib.h* - Funções de uso genérico.

INSTITUTO FEDERAL
Bahia

Linguagem C - Fundamentos em C

- Segundo modo (com "*nome_do_arquivo*")
  - É usado normalmente para incluir algum arquivo que tenha sido criado pelo próprio programador ou por terceiros;
  - Tem que estar no mesmo diretório em que o programa está sendo compilado.

Linguagem C - Comandos Básicos

- As instruções de entrada e saída são os comandos mais básicos e obrigatórios em quase todos os programas em C;
- Objetivo dos programas?

Linguagem C - Comandos Básicos

- As instruções de entrada e saída são os comandos mais básicos e obrigatórios em quase todos os programas em C;
- Objetivo dos programas?
  - Fornecimento de um conjunto de dados (entradas);

Linguagem C - Comandos Básicos

- As instruções de entrada e saída são os comandos mais básicos e obrigatórios em quase todos os programas em C;
- Objetivo dos programas?
  - Fornecimento de um conjunto de dados (entradas);
  - Realização de cálculos ou pesquisas (processamento);

INSTITUTO FEDERAL
Bahia

Linguagem C - Comandos Básicos

- As instruções de entrada e saída são os comandos mais básicos e obrigatórios em quase todos os programas em C;
- Objetivo dos programas?
  - Fornecimento de um conjunto de dados (entradas);
  - Realização de cálculos ou pesquisas (processamento);
  - Obtenção de resultados (saídas).

Linguagem C - Comandos Básicos

- A função *printf*()
  - Servirá basicamente para apresentação dos dados no monitor;
  - Forma geral: *printf ("string de controle", lista de argumentos)*;

Linguagem C - Comandos Básicos

```
#include <stdio.h>
#include <conio.h>
main ( ) {
int numero;
numero=10;
printf("O %d elevado ao quadrado resulta em %d. \n",
numero,numero*numero);
        getch ( );
}
```

Linguagem C - Comandos Básicos

- Operadores Especiais suportados por *printf*()
  - $\backslash b$ Retrocesso (BackSpace);
  - $\backslash f$ Salto de página (Form Feed);
  - $\backslash n$ Nova linha (Line Feed);
  - $\backslash t$ Tabulação horizontal (TAB);
  - $\backslash x$ Representação de *byte* na base hexadecimal;
- Exemplo: *printf*("$\backslash x41$"), causa a impressão da letra A na tela.

INSTITUTO FEDERAL
Bahia

Linguagem C - Comandos Básicos

- A função *scanf*()
- Serve para fazer a leitura de dados tipados através do teclado;
- Forma geral: *scanf("string de controle", lista de argumentos)*;
- Esta é a sintaxe simples, posteriormente veremos a sintaxe completa.

Linguagem C - Comandos Básicos

- A função *scanf*()
- Sintaxe básica:
  - %c - leitura de caractere;
  - %d - leitura de números inteiros;
  - %f - leitura de números reais;
  - %s - leitura de caracteres.
- Importante! A lista de argumentos deve conter **exatamente** a mesma quantidade de códigos que estão sendo utilizados na $<$ *stringdecontrole* $>$.

INSTITUTO FEDERAL
Bahia

Linguagem C - Comandos Básicos

- A função *scanf*()
- Cada variável a ser lida, deverá ser precedida pelo caractere & ;
- Para sequência de caracteres (%s), o caractere & não deverá ser usado.

Linguagem C - Comandos Básicos

```
/* Exemplo Lê e Mostra Idade */

main ( ) {

int idade;

char nome[30];

printf("Digite sua Idade: ");

scanf("%d",&idade);

printf("Seu Nome: ");

scanf("%s",nome); /* Strings não utilizar '&' na leitura */

printf("%s sua idade e' %d anos. \n", nome, idade);

}
```

INSTITUTO FEDERAL
Bahia

Linguagem C - Tomada de Decisão

- *if*
- *if-else*
- *switch*

Linguagem C - Tomada de Decisão

- Fazer um programa que utilize *if-else* para dizer se uma pessoa é:
  - Idosa (> 70)
  - Adulta (> 21)
  - Jovem (caso contrário)

Linguagem C - Tomada de Decisão

- Fazer um programa que utilize *if-else* para dizer o maior valor entre três números.

Linguagem C - Tomada de Decisão

- Fazer um programa que utilize *switch* para dizer se uma pessoa é idosa (70 acima) ou adulta (70 abaixo).

Linguagem C - Laços

- Utilizar o laço *for* para imprimir a tabuada de um determinado número.

Linguagem C - Laços

- Elabore tabela de Conversão de temperaturas entre as escalas Celsius e Fahrnheit;
- Fahrnheit 0 - 300, incrementa em 20;
- $celsius = (5.0/9.0) * (fahr - 32)$;

Linguagem C - Laços

- Comando while
- Semelhante ao Java já aprendido, o laço while significa enquanto;
- É geralmente utilizado para se realizar repetições quando não se pode determinar a quantidade de vezes que será repetido o laço;
- *while (condição) { <instrução>; }*

Linguagem C - Laços

- Utilizando o comando while faça a contagem de 0 a 100 imprimindo na tela;

Linguagem C - Laços

- Utilizando o comando while faça a contagem de 0 a 100 imprimindo na tela;

```
main ( ) {

int cont=0;

while (cont = 100)

                    cont++;

          printf("%d",cont);

}
```

Linguagem C - Laços

- Comando do-while
- O laço do-while significa faça enquanto;
- O diferencial desse laço é que o código dentro dele é executado pelo menos uma vez, mesmo que a condição seja falsa;
- *do { <instrução>; } while (condição);*
- Como ficaria a contagem de 0 a 100?

INSTITUTO FEDERAL
Bahia

Linguagem C - Laços

```
#include <stdio.h>

#include <stdlib.h>
int main() {

int cont=0;

do {
                    cont++;

                    printf("%d ",cont);

} while (cont < 100);

system("pause");

}
```

Linguagem C - Laços

- Exercícios

1. Some, subtraia, multiplique e divida 10 com 15 e imprima na tela a seguinte frase: "O resultado da XX é: " mostrando o resultado.

2. Leia o nome e as duas notas de um aluno e apresente ambos na tela, juntamente com a média.

Linguagem C - Funções

- Conceitualmente, C é baseada em blocos de construção;
- Assim sendo, um programa em C nada mais é que um conjunto de funções básicas ordenadas pelo programador;
- As instruções printf( ) e scanf( ), vistas anteriormente, não fazem parte do conjunto de palavras padrões da linguagem (instruções), pois não passam elas mesmas de funções escritas em C!

INSTITUTO FEDERAL
Bahia

Linguagem C - Funções

- Esta abordagem permite a portabilidade da linguagem, pois seus comandos de entrada e saída, não são parte do conjunto básico da linguagem;
- Isso a livra dos problemas de suporte aos diversos padrões de vídeos, teclados e sistemas operacionais existentes;
- Cada função em C é uma sub-rotina, contendo um ou mais comandos em C ou executa uma ou mais tarefas.

INSTITUTO FEDERAL
Bahia

Linguagem C - Funções

- Uma função em C deve possuir um **nome** e uma **lista de argumentos**

```
main ( ) {
                    alo ( );
}
alo ( ) {
                    printf (“Alô!\n\n”);
}
```

Linguagem C - Funções

- Exercício: Faça o quadrado de um número utilizando uma função.

Linguagem C - Funções

- Exercício: Faça o quadrado de um número utilizando uma função.

```
main ( ) {
int num;
printf("Digite um numero: ");
scanf("%d",&num);
sqr(num); /* sqr recebe "num" do programa principal */
}
sqr ( ) {
                int x; /* x é um "parâmetro" recebido do programa principal
                no caso x "vale" o conteúdo de num */
                printf("%d ao quadrado e' %d ",x,x*x);
```

Linguagem C - Funções

- O código do exercício anterior possui um erro, lembre-se de passar os parâmetros nas funções!

Linguagem C - Funções

- Argumento: se refere ao valor que é usado para chamar uma função;
- Parâmetro: se refere à variável em uma função que recebe o valor dos argumentos usados na função;
- A distinção que deve ser compreendida é que a variável usada como argumento na chamada de uma função não tem nenhuma relação com o parâmetro formal que recebe o valor dessa variável.

Linguagem C - Funções

- A declaração de uma função quando feita no inicio de um programa em C é dita protótipo da função;
- Esta declaração deve ser feita sempre antes da função *main*, definindo-se o tipo, o nome e os argumentos desta mesma função. Exemplo:
- *float soma (float, float);*
- O protótipo indica ao compilador C que a função está definida em outro local do código.

Linguagem C - Funções

- VAMOS CONCERTAR A FUNÇÃO DA RAIZ QUADRADA!

Linguagem C - Funções

- Função Recursiva
- Uma função denomina-se recursiva quando dentro dela se faz uma chamada para ela mesma;
- Um exemplo prático seria o cálculo do fatorial de um número.
- Faça esse exemplo utilizando o do-while! Obs: Laço infinito, repetindo a função até que o usuário digite um número negativo!)

INSTITUTO FEDERAL
Bahia

Linguagem C - Matrizes

- Uma Matriz é um conjunto de variáveis de mesmo tipo que compartilham um mesmo nome;
- Com Matriz agora podemos armazenar mais de um valor para depois serem manipulados através de um índice;
- ESSE ÍNDICE REFERENCIA UM DOS ELEMENTOS!

Linguagem C - Matrizes

- Para criar uma matriz:
  - Definir um tipo;
  - Definir um nome;
  - Definir a quantidade de elementos ([ ]);
- int mat[5];

Linguagem C - Matrizes

- Referenciando elementos de uma matriz:
  - x = mat[10];

Linguagem C - Matrizes

- Referenciando elementos de uma matriz:
  - x = mat[10]; x recebe o elemento de mat na posição 10;
  - mat[10] = 20;

Linguagem C - Matrizes

- Referenciando elementos de uma matriz:
  - x = mat[10]; x recebe o elemento de mat na posição 10;
  - mat[10] = 20; o elemento de mat na posição 10 recebe o valor 20;

Linguagem C - Matrizes

- Inicialização de matrizes:
  - Matriz inicializada com os valores 1,2 e 3:
  - int mat[3] = 1;2;3;

Linguagem C - Matrizes

- Inicialização de matrizes:
  - Matriz inicializada com os valores 1,2 e 3:
  - int mat[3] = 1;2;3;
- Exercício: Implementar um programa que calcula a media de três notas utilizando uma matriz.

Linguagem C - Ponteiros

- O que é um ponteiro?
  - O ponteiro nada mais é do que uma variável que guarda o endereço de outra variável.
- Declarando um ponteiro:
  - int *ptr;
  - **Essa declaração define a variável ptr como um ponteiro para uma variável do tipo int (número inteiro).**

Linguagem C - Ponteiros

- Inicializando um ponteiro:
  - Para se inicializar um ponteiro é necessário apenas atribuir-se um endereço de memória.

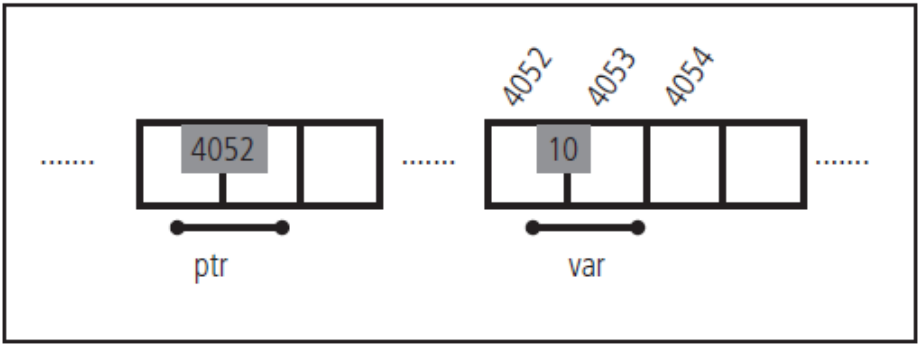
```
int var;

int *ptr;

var = 10;

ptr = &var;
```

Linguagem C - Ponteiros



Linguagem C - Ponteiros

- Com ptr apontando para var, é possível realizar operações com esta última de forma indireta, a partir de ptr:
  - int newVar = *ptr;

Linguagem C - Ponteiros

- Com ptr apontando para var, é possível realizar operações com esta última de forma indireta, a partir de ptr:
  - int newVar = *ptr;
  - *ptr = 20;

Linguagem C - Ponteiros

- Com ptr apontando para var, é possível realizar operações com esta última de forma indireta, a partir de ptr:
  - int newVar = *ptr;
  - *ptr = 20;
- Exercício: Crie um programa que solicite ao usuário 2 números e retorne a soma. As operações devem ser feitas utilizando ponteiros.

Linguagem C - Exercícios para fixação - Funções

1. Fazer um programa que calcule o volume de uma esfera, sendo que o volume de uma esfera é raio*raio*raio. Crie uma função que faça esse cálculo.

2. Elabore programa que leia "n" números digitados e apresente sua média.

3. Escreva uma função que receba dois números e retorne o menor número.

4. Faça uma função que recebe a idade de uma pessoa em anos, meses e dias e retorna essa idade expressa em dias.

Linguagem C - Exercícios para fixação - Matrizes

1. Faça um programa que tenha uma matriz (vetor) denominada *A* que armazene 6 números inteiros. O programa deve executar os seguintes passos:

   - Atribua os seguintes valores ao vetor: 1, 0, 5, -2, -5, 7;
   - Armazene em uma variável inteira simples a soma entre os valores das posições A[0], A[1] e A[5] da matriz e mostre na tela;
   - Modifique a posição 4, atribuindo o valor 100;
   - Mostre na tela cada valor do vetor A, um em cada linha.

Linguagem C - Exercícios para fixação - Matrizes

1. Faça um programa que tenha uma função que leia uma matriz (vetor) de 10 posições e conte quantos números pares ele possui.

2. Faça uma função que leia uma matriz 4x4, conte e escreva quantos valores maiores que 10 ela possui.

Dúvidas?

- Dúvidas?