



Estruturas de Dados
Aula 1: Introdução e
conceitos básicos

1



Website

- <https://classroom.google.com/c/MjUyNDMxOTg3MzMw>
- Profa. Patrícia Dockhorn Costa
- Comunicação via Google Classroom.

2

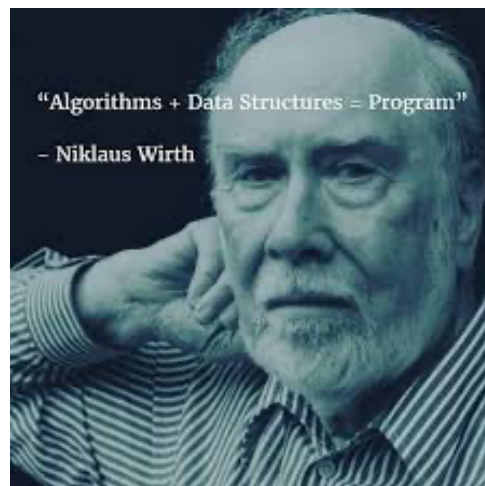
Introdução



- O que são estruturas de dados?
- Benefícios?
 - Organização da informação
 - Melhora o desempenho
 - Proporciona o reuso de código
 - Proporciona interoperabilidade
 - Diminui custos

3

Estruturas de Dados...



4

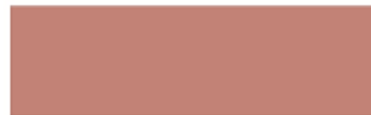
Estruturas de Dados...



Bad programmers worry
about the code. Good
programmers worry about
data structures and their
relationships.

Linus Torvalds

PICTUREQUOTES.COM



PICTUREQUOTES

5

Estruturas de Dados...



Smart data structures and
dumb code works a lot better
than the other way around.

Eric S. Raymond



6

Objetivos gerais



- Familiarizar os alunos com as principais estruturas de dados, e suas correspondentes abstrações.
- Final do curso o aluno poderá projetar e implementar diversas estruturas de dados, conhecendo suas vantagens e desvantagens

7

Background da turma



8

Programa da disciplina



- Introdução
- Uso eficiente de memória
 - Alocação estática e alocação dinâmica
 - Estruturas de dados encadeadas
 - Manipulação de ponteiros em C
- Tipos Abstratos de Dados
 - Definição de estrutura abstrata
 - Pré e pós condições
 - Conceito de software em camadas
- Listas
 - Lista com alocação estática e alocação dinâmica
 - Listas duplamente encadeadas
 - Listas circulares e listas duplamente encadeadas circulares

9

Programa da disciplina (2)



- Pilhas
 - Pilhas com estruturas estática e dinâmica
 - Aplicações
- Filas
 - Filas com estruturas estática e dinâmica
 - Aplicações
- Recursão
- Árvores
 - Árvore binária
 - Algoritmos de travessia
 - Árvore binária de busca
 - Árvore com Número Variável de filhos
 - Árvore genérica
- Introdução a algoritmos de busca e ordenação
- Tabela Hash

10

Critérios de avaliação



- Dois trabalhos computacionais e listas de exercícios.
 - A média parcial é calculada por: $MP = (((T1 + T2)/2) * 0,7) + ((SOMA(listas)/NL) * 0,3)$
 - T1 é a nota do primeiro Trabalho de Programação e T2 é a nota do segundo Trabalho de Programação
 - SOMA(listas) é a soma das notas das listas de exercícios
 - NL é o número de listas de exercícios
 - A média final será:
 - $MF = MP$, se $MP \geq 7,0$.
 - $MF = (PF + MP)/2$, se $MP < 7,0$. (PF é a nota da prova final)
- Se $MF \geq 5,0 \Rightarrow$ Aprovado.
Se $MF < 5,0 \Rightarrow$ Reprovado.

11

Links úteis



<https://opendatastructures.org>

https://upload.wikimedia.org/wikipedia/commons/5/5b/Data_Structures.pdf

<https://www.studytonight.com/data-structures/>

<https://www.geeksforgeeks.org/data-structures/>

12

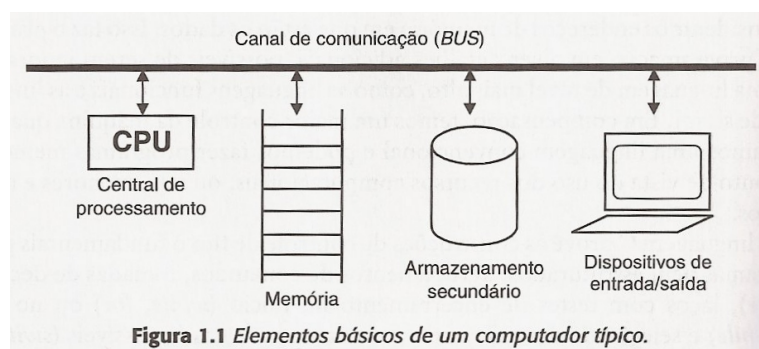
Bibliografia



- Ebook na biblioteca da UFES:
<https://integrada.minhabiblioteca.com.br/books/9788595156654>
- Celes, Cerqueira e Rangel. Introdução a Estruturas de Dados, Editora Campus
- ZIVIANI, Nivio Projetos de Algoritmos com implementações em Pascal e C , Livraria Pioneira Informática , 1993
- SEDGEWICK, Robert. Algorithms in C, Parts 1-4, Third Edition, Addison- Wesley, 1997
- SZWARCFITER, Jayme e MARKENZON Lilian Estruturas de Dados e seus Algoritmos, 2a Edicao, LTC - Livros Tecnicos e Científicos S.A., 1997
- CORMEN, Thomas H., LEISERSON, Charles E., RIVEST, Ronald L., STEIN, Cliff. Introduction to Algorithms (Second Edition), MIT Press, 2001
- TENENBAUM, LANGSAM, and AUGENSTEIN Estruturas de Dados usando C, Pearson, 2005
- KNUTH, Donald E. The Art of Computer Programming, Volume I, Third Edition, Addison-Wesley, 1997

13

Modelo de um Computador

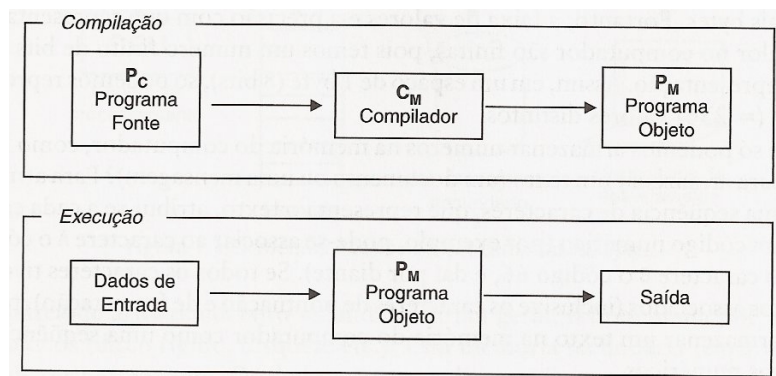


14

Compilação de Programas



- Compilação: “tradução” de código fonte (P_C) para linguagem de máquina (M).
- Compilador (C_M), escrito em M : lê o programa P_C e traduz cada instrução para M , escrevendo o programa objeto (P_M)



15

Ciclo de Desenvolvimento



- Programas em C geralmente são divididos em vários arquivos
- Cada arquivo pode ser compilado separadamente
- Para gerar um executável, precisamos reunir os os códigos dos arquivos separados (juntamente com as bibliotecas usadas) : **Ligador**
- Bibliotecas: permitem que funções de interesse geral sejam usadas por vários programas
- O ligador pode ser usado automaticamente pelo compilador (biblioteca padrão), ou deve ser explicitamente acionado

16

Ciclo de Desenvolvimento (cont.)

