



Estruturas de Dados  
Aula 1: Introdução e  
conceitos básicos

Web site



- <http://www.inf.ufes.br/~pdcosta/ensino/>
- Profa. Patrícia Dockhorn Costa
- Email: [pdcosta@inf.ufes.br](mailto:pdcosta@inf.ufes.br)
- Monitoria: Alexandro  
[alexandrosouzaramos@gmail.com](mailto:alexandrosouzaramos@gmail.com)

## Introdução



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

## 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

## Background da turma



## Programa da disciplina



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

## 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

## Critérios de avaliação



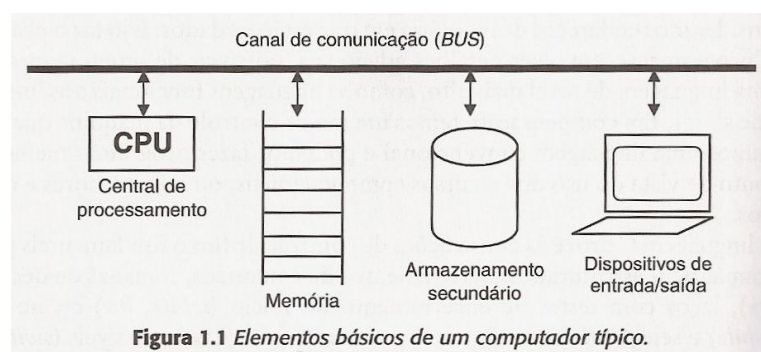
- Duas provas parciais e trabalhos. A média parcial é calculada por:  $MP = 0,5 \cdot P + 0,5 \cdot T$   
onde: P é a média aritmética das provas parciais e T é a média aritmética das notas dos trabalhos.  
A média final será:  
 $MF = MP$ , se  $MP \geq 7,0$ .  
 $MF = (PF + MP)/2$ , se  $MP < 7,0$ . (PF é a nota da prova final)
- Se  $MF \geq 5,0$  -> Aprovado.  
Se  $MF < 5,0$  -> Reprovado.

## Bibliografia



- 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 Cientificos 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

## Modelo de um Computador



## Armazenamento de Dados



- Organização da memória:
- Bits
  - Menor unidade
  - Valores 0 ou 1
- Bytes
  - Sequência de 8 bits
- Célula
  - Sequência de bits
  - Menor unidade de endereçamento
- Palavras
  - Sequência de bytes
  - Varia conforme arquitetura

		0	1	2	3	4	5	6	7
1	0	0	1	1	1	0	0	1	0
	1	1	1	0	0	1	1	1	0
	2	0	1	1	1	0	0	1	0
2	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0
	2	1	1	1	1	1	1	1	1
3	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0
	2	1	1	1	1	1	1	1	1

## Armazenamento de Dados

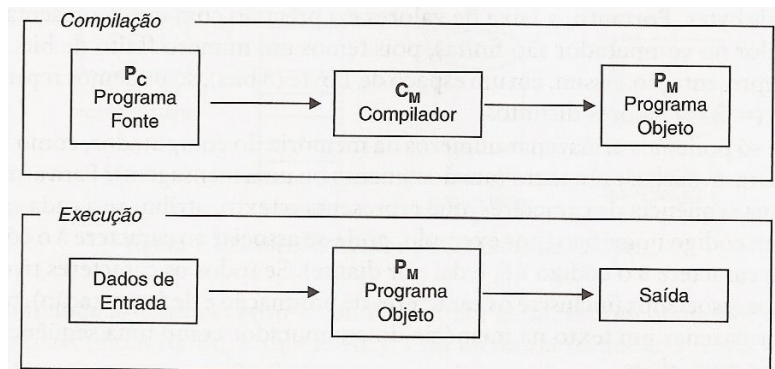


- Espaço de memória finito
  - Ex., espaço de 1 byte (8 bits), podemos representar apenas  $2^8$  (=256) valores distintos
- Podemos representar texto
  - Associando valores aos caracteres
  - Ex., A(65) e B (66)
- Representar um programa na memória
  - Programas são executados em **linguagem de máquina**
  - Programas executáveis são sequências de instruções (códigos numéricos)
    - 10110000 01100001 (MOV AL, 61h)

## 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 ( $P_M$ )



## 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 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

## Ciclo de Desenvolvimento (cont.)

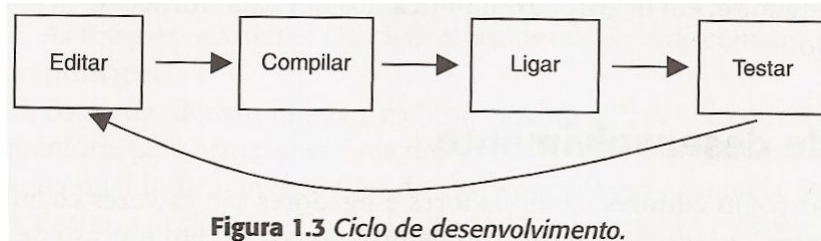


Figura 1.3 Ciclo de desenvolvimento.

## Tipos de Dados



- Definição
  - Conjunto de valores (domínio)
  - Possíveis operações
  - Ex. int
    - Domínio (conjunto dos números inteiros)
    - Operações (soma, subtração, multiplicação, etc.)
  - Ex. boolean
    - Domínio (true, false)
    - Operações (and, or, not)



## Tipos de Dados (2)



- Tipos estruturados de dados
  - Alguns tipos de dados possuem organização, estrutura interna
  - Ex. fração
    - Sinal (+, -)
    - Numerador
    - Denominador (diferente de zero)
    - Operações (soma, subtração, normalização, etc)
  - Endereço
    - Logradouro (praça, rua, etc)
    - Número
    - Bairro
    - Cidade
    - CEP
    - País

## Tipos de Dados (3)



- Podemos também definir o domínio de um tipo de dado usando apenas *comportamento*
  - Não impõe uma estrutura interna fixa
  - Usa operações para criar valores de um determinado tipo
  - Define o tipo somente em termos de operações
- Ex. frações
  - CRIA\_FRACAO (N, D)
  - VALOR\_FRACAO = CRIA\_FRACAO (N, D)
- Ex. endereço
  - VALOR\_ENDERECO = CRIA\_ENDERECO ()
  - MODIFICA\_LOGRADOURO (VALOR\_ENDERECO, "Av. Fernando Ferrari")

## Exercício



- Defina os seguintes tipos de dados usando as duas abordagens apresentadas anteriormente (estrutura e comportamento)
  - Tipo ponto (coordenadas plano bidimensional)
  - Tipo reta
  - Tipo polinômio (grau 2)