

Tópicos Especiais em Programação

I

Jordana S. Salamon

jssalamon@inf.ufes.br

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Revisão de conceitos de Programação I

- ▶ O que é um computador?



Revisão de conceitos de Programação I

- ▶ O que é um computador?
- ▶ “Um computador é uma seleção de componentes que realizam operações lógicas e aritméticas sobre um grande volume de dados” F. K. Miyazawa



nemo

Revisão de conceitos de Programação I

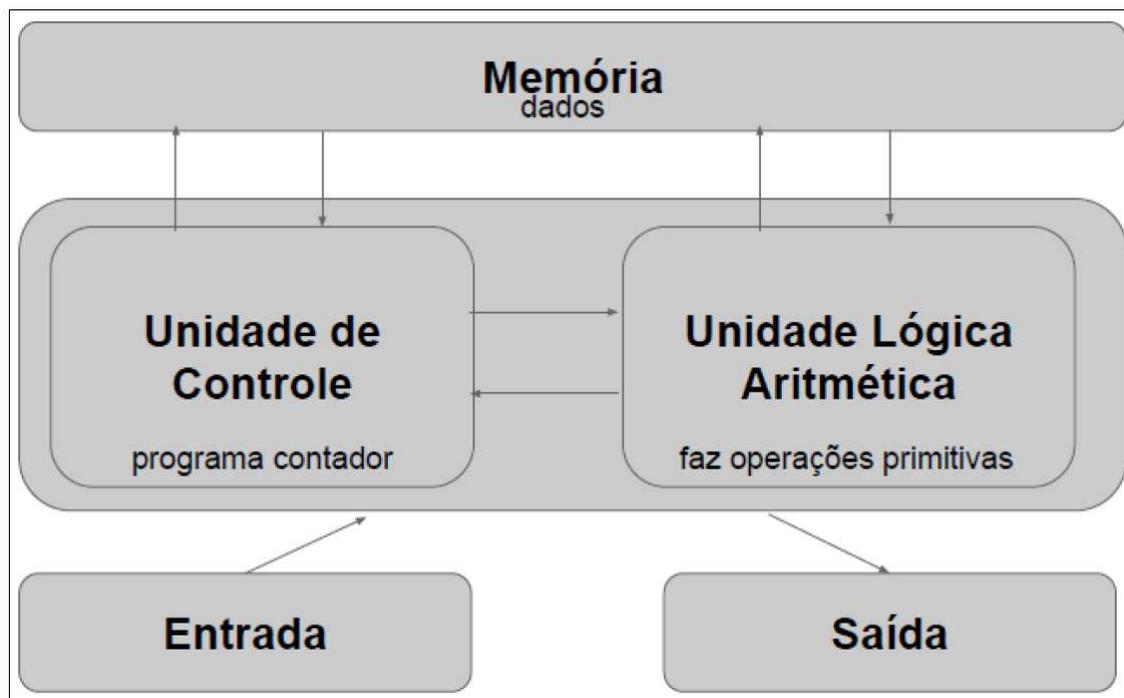
- ▶ Qual a arquitetura básica de um computador?



nemo

Revisão de conceitos de Programação I

- Qual a arquitetura básica de um computador?



Revisão de conceitos de Programação I

- ▶ O que é programar?

Revisão de conceitos de Programação I

- ▶ O que é programar?
- ▶ Identificar, organizar e codificar o conhecimento para a descrição de um conceito para o computador
- ▶ Programação é o processo de escrita, teste e manutenção de um programa de computador. (Fonte: Wikipedia)



nemo

Revisão de conceitos de Programação I

- ▶ Qual o processo que deve ser seguido para programar?



nemo

Revisão de conceitos de Programação I

- ▶ Qual o processo que deve ser seguido para programar?
 1. compreender o problema a ser resolvido;
 2. planejar uma solução;
 3. descrever a solução por meio de uma linguagem de programação;
 4. testar o programa com o objetivo de verificar se ele resolve o problema;



nemo

Revisão de conceitos de Programação I

- ▶ O que é uma linguagem de programação?



nemo

Revisão de conceitos de Programação I

- ▶ O que é uma linguagem de programação?
- ▶ A linguagem de programação é um método padronizado, formado por um conjunto de regras sintáticas e semânticas, que informará instruções de processamento ao computador.
- ▶ Permite que um programador especifique precisamente quais os dados que o computador irá atuar, como estes dados serão armazenados ou transmitidos e, quais ações devem ser tomadas de acordo com as circunstâncias. Linguagens de programação podem ser usadas para expressar algoritmos com precisão. (Fonte: Wikipedia)

Revisão de conceitos de Programação I

- ▶ O que é um algoritmo?

Revisão de conceitos de Programação I

- ▶ O que é um algoritmo?
- ▶ Um algoritmo é uma sequência lógica finita de passos para realizar uma tarefa ou resolver um problema.
- ▶ Ou seja, um algoritmo é uma descrição passo a passo de como o computador irá executar uma operação específica.
- ▶ Um programa, por outro lado, é uma entidade que na verdade implementa uma ou mais operações de forma que seja útil para as pessoas que o utilizam. (Fonte: Wikipedia)

Revisão de conceitos de Programação I

- ▶ Do que precisamos para escrever um algoritmo em C?

Revisão de conceitos de Programação I

- ▶ Do que precisamos para escrever um algoritmo em C?
- ▶ Função Main;
- ▶ Variáveis;
- ▶ Tipos de dados;
- ▶ Comandos de atribuição;
- ▶ Expressões lógicas, aritméticas ou relacionais;
- ▶ Estruturas condicionais;
- ▶ Comandos de repetição;
- ▶ Funções;
- ▶ Estruturas sequenciais;
- ▶ Estruturas de dados;

O que são estruturas de dados (ED)?

- ▶ Uma estrutura de dados é uma coleção tanto de valores (e seus relacionamentos) quanto de operações (sobre os valores e estruturas decorrentes).
- ▶ Para definir uma estrutura de dados na linguagem C utilizamos a palavra-chave **struct**.



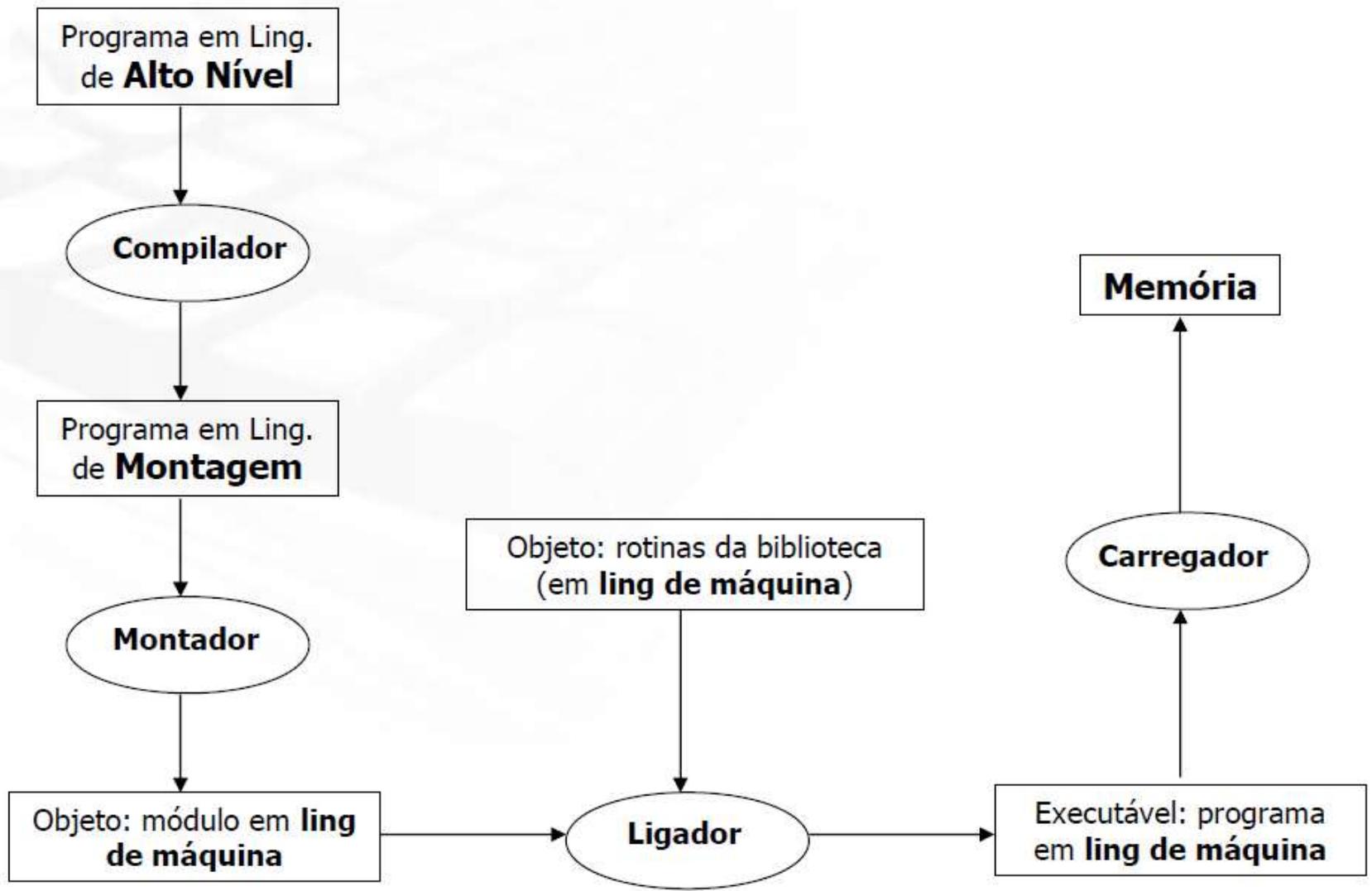
nemo

Benefícios e necessidade de ED

► Benefícios?

- Organização da informação
- Melhora o desempenho
- Proporciona o reuso de código
- Proporciona interoperabilidade
- Diminui custos

Modelo básico de memória



Modelo básico de memória

► Compiladores

- São programas que recebem como entrada arquivos texto contendo módulos escritos em linguagem de alto nível e geram como saída arquivos objeto correspondentes a cada módulo.
- Se todas as bibliotecas ou módulos são apresentados como entrada, geram um programa executável diretamente.



nemo

Modelo básico de memória

► Interpretadores

- Recebem como entrada arquivos texto contendo programas em linguagem assembly ou linguagem de alto nível, ou arquivos binários com instruções de máquina, e os executam diretamente.
- Interpretadores percorrem os programas, a partir de seu ponto de entrada, executando cada comando.
- Processadores são interpretadores implementados em hardware!

Modelo básico de memória

▶ Montadores (Assemblers)

- ▶ Montam um programa em linguagem de máquina a partir de sua versão em linguagem de montagem. Geram um arquivo objeto. Em geral, não pode ser executado diretamente pela máquina, por conter referências a sub-rotinas e dados especificados em outros arquivos.



nemo

Modelo básico de memória

▶ Ligadores (Linkers)

- ▶ São programas especiais que recebem como entrada arquivos objetos e geram como saída o programa final em linguagem de máquina. Gera um programa executável a partir de um ou mais arquivos objeto.

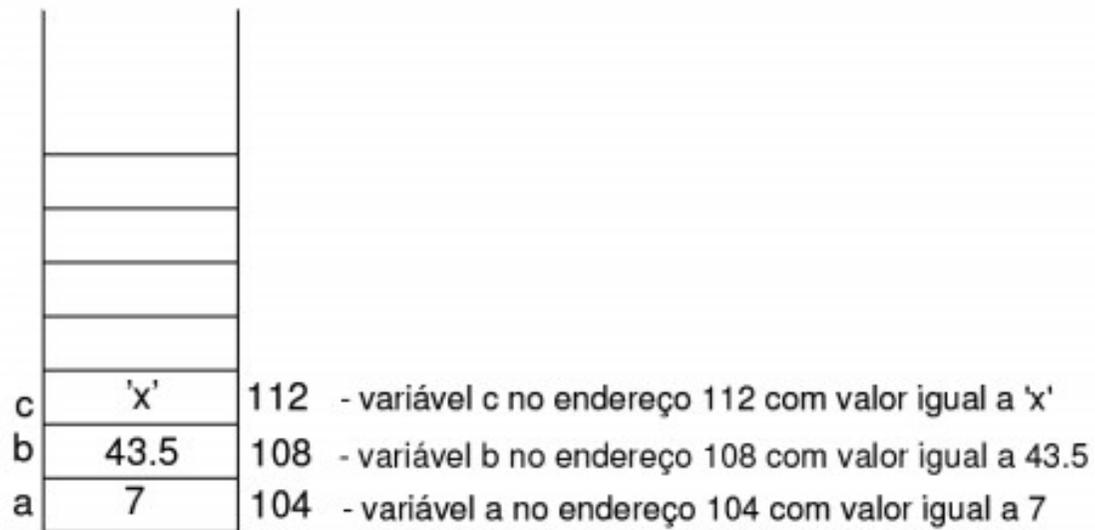
▶ Carregadores (Loaders)

- ▶ Para executar um programa, um loader deve ser utilizado. O carregador é, em geral, parte do sistema operacional.

Modelo básico de memória

- ▶ Variáveis locais têm escopo local
- ▶ Funções são independentes entre si
- ▶ Transferência de dados entre funções através de
 - Passagem de parâmetros
 - Valor de retorno
- ▶ Parâmetros em C são passados por valor
- ▶ Pilha de Execução: Coordena comunicação entre a função que chama e a função chamada
 - Permite passagem de parâmetros e valores de retorno

Modelo básico de memória



Modelo básico de memória

► Variáveis Globais

- Declaradas fora do escopo das funções
- São visíveis a todas as funções
- Existem enquanto o programa existir (não estão na pilha de execução)
- Utilização:
 - Devem ser usadas com critério
 - Podem criar muita dependência entre as funções
 - Dificulta o entendimento e o reuso de código

Modelo básico de memória

► Variáveis Estáticas

- Declaradas no escopo de funções
- Existem enquanto o programa existir (não estão na pilha de execução)
- Somente são visíveis dentro das funções nas quais são declaradas
- Utilização
 - Quando for necessário recuperar o valor de uma variável na execução passada da função

Modelo básico de memória

- ▶ Variáveis Estáticas

```
void imprime (float a)
{
    static int n=1;
    printf (" %f ", a);
    if ((n%5) == 0) printf ("\n");
    n++;
}
```

Modelo básico de memória

- ▶ Variáveis estáticas e globais são inicializadas com zero, quando não forem explicitamente inicializadas
- ▶ Variáveis globais estáticas
 - ▶ - São visíveis para todas funções subsequentes
 - ▶ - Não podem ser acessadas por funções de outros arquivos
- ▶ Funções estáticas
 - ▶ - Não podem ser acessadas por funções de outros arquivos

Tipos de dados simples e estruturados

► Tipos de dados simples

Tipo	Num de bits	Formato para leitura com scanf	Intervalo	
			Início	Fim
char	8	%c	-128	127
unsigned char	8	%c	0	255
signed char	8	%c	-128	127
int	16	%i	-32.768	32.767
unsigned int	16	%u	0	65.535
signed int	16	%i	-32.768	32.767
short int	16	%hi	-32.768	32.767
unsigned short int	16	%hu	0	65.535
signed short int	16	%hi	-32.768	32.767
long int	32	%li	-2.147.483.648	2.147.483.647
signed long int	32	%li	-2.147.483.648	2.147.483.647
unsigned long int	32	%lu	0	4.294.967.295
float	32	%f	3,4E-38	3.4E+38
double	64	%lf	1,7E-308	1,7E+308
long double	80	%Lf	3,4E-4932	3,4E+4932

Tipos de dados simples e estruturados

- ▶ Tipos de dados estruturados
- ▶ São tipos de dados que combinam tipos de dados simples, formando uma informação mais complexa;
- ▶ Exemplos:
 - ▶ Uma pessoa que possui nome, idade, peso e altura.
 - ▶ Um aluno que possui nome, número de matrícula e curso.
 - ▶ Um livro que possui título, gênero, autor(es), ISBN, etc.
 - ▶ Um jogo que possui nome, gênero, valor, empresa desenvolvedora.

That's all Folks!



nemo