

Algoritmos e Programação I

ASTI - 1º Semestre
Professora Andréia Machion

Esta Aula

- ▶ **O que é Programação**
 - ▶ Para que programar
 - ▶ O que é um Algoritmo
 - ▶ Como interpretar um problema e transformá-lo em um algoritmo
 - ▶ Alguns exercícios para fixar o conhecimento



Primeiros Passos

- ▶ Para que serve a programação?
 - ▶ Comandar o computador e periféricos
 - ▶ Resolver problemas rotineiros (de banco de dados, por exemplo)
 - ▶ Realizar tarefas repetitivas, exaustivas ou monótonas
- ▶ O que é necessário?
 - ▶ Entendimento do problema
 - ▶ Organização do pensamento para solução
 - ▶ Transcrição para uma “linguagem” que o computador entenda



O que é um “programa” ?

- ▶ Um algoritmo codificado em uma linguagem de programação
- ▶ Algoritmo?
 - ▶ Sequência lógica e finita de instruções para resolver um problema
- ▶ Então ...??
 - ▶ Entendemos o problema
 - ▶ Descrevemos os passos para resolvê-lo, organizando o raciocínio
 - ▶ Traduzimos para linguagem específica



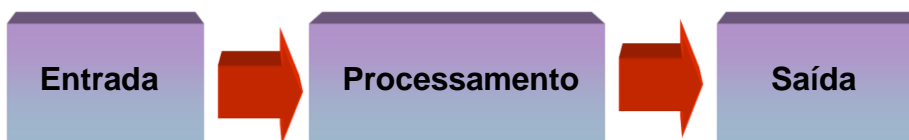
Exemplo

- ▶ Num determinado curso, você tem duas avaliações. Como é que você faz para saber a sua nota final, sabendo que a nota final é a média das duas avaliações?

▶

O algoritmo

- ▶ Um algoritmo é composto basicamente de 3 elementos



▶

Como identificar esses elementos?

- ▶ **Entrada:**
 - ▶ Toda e qualquer informação que nós usuários precisamos fornecer ao programa
- ▶ **Processamento:**
 - ▶ O propósito pelo qual o programa deve existir
 - ▶ O que “ele” (programa) deve fazer
- ▶ **Saída**
 - ▶ Toda e qualquer informação que o programa precisa informar ao usuário



Voltando ao exemplo...

- ▶ **Entrada**
 - ▶ Os valores das notas (Nota1 e Nota2)
- ▶ **Processamento**
 - ▶ Somar as notas
 - ▶ Dividir o resultado da soma por 2
- ▶ **Saída**
 - ▶ A média calculada



Praticando

- ▶ Dona Maricota é uma senhora muito simpática, que vai toda santa sexta-feira à farmácia do Sr. Godofredo comprar remédio para pressão alta.
- ▶ Ela sempre ganha 15% de desconto no valor de sua compra, mas, como ela sempre compra coisas a mais, nunca sabe o quanto está economizando (em reais). Como podemos ajudá-la?



Identificando os elementos

- ▶ **Entrada:**
 - ▶ Valor da compra que a D. Maricota realizou

Total

- ▶ **Processamento**
 - ▶ Calcular 15% do Total

$* 15 / 100$ } Esse é o valor do desconto

- ▶ **Saída**
 - ▶ O valor em R\$ do desconto que a D. Maricota teve

Desconto



Criação, compilação e execução de um programa

Criação do programa

- ▶ A criação dos programas fonte em linguagem C faz-se com o auxílio de um editor de texto genérico, ou específico de um ambiente de desenvolvimento.
- ▶ Os arquivos de texto produzidos deverão ter a extensão .c
- ▶ Nomes de arquivos NÃO devem ter acento, espaço em branco, cedilha ou qualquer outro carácter especial, a menos de sublinhado ()



Compilação

- ▶ “Transforma” o programa fonte em programa executável
- ▶ É feita pela invocação de um compilador
 - ▶ em linha de comando (p. ex. no UNIX, o comando `cc`), ou
 - ▶ por uma opção de menu de um ambiente de programação
- ▶ O comando de compilação deverá ser seguido pelo nome do arquivo que contém o código fonte (geralmente com a extensão `.c`), quando a compilação é feita em linha de comando

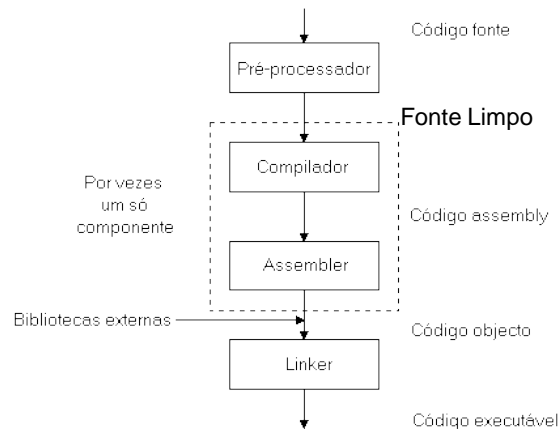


Execução do programa

- ▶ Se a operação anterior tiver sucesso, a execução do programa compilado produzido faz-se simplesmente invocando-o como se fosse um comando do sistema operacional.
- ▶ Durante a execução podem tornar-se evidentes mais alguns erros:
 - ▶ erros de execução (p. ex. divisão por zero),
 - ▶ erros que levem a que o programa não se comporte como esperado.
- ▶ Neste caso é necessário voltar à edição do programa fonte para corrigir a sua lógica, e depois efetuar também uma nova compilação para produzir a nova versão do código executável.



O modelo de compilação da linguagem C



O pré-processador

- ▶ O pré-processador atua apenas no nível do código fonte, ou seja, trabalha apenas com texto.
- ▶ Algumas das suas funções são:
 - ▶ remover os comentários de um programa;
 - ▶ interpretar diretivas especiais a si dirigidas, que começam pelo carácter #.

O compilador

- ▶ Alguns compiladores traduzem o código fonte (texto) recebido do pré-processador para linguagem assembly (também texto).
- ▶ São também comuns os compiladores capazes de gerarem diretamente código objeto (instruções do processador já em código binário).



O assembler

- ▶ O assembler traduz código em linguagem assembly (texto) para código objeto. Pode estar integrado no compilador. O código objeto é geralmente armazenado em arquivos com a extensão .o (unix) ou .obj (ms-dos).



O linker

- ▶ Combina todos os objetos com o resultado compilado das funções das bibliotecas num único arquivo, gerando o código executável, se o programa referencia funções da biblioteca standard ou outras funções contidas em arquivos com código fonte diferentes do principal (que contém a função `main()`).
 - ▶ As referências a variáveis globais externas também são resolvidas pelo linker.
-



A utilização de bibliotecas

- ▶ A linguagem C é muito compacta.
 - ▶ Muitas das funções que fazem parte de outras linguagens não estão diretamente incluídas na linguagem C: as operações de entrada/saída, a manipulação de strings e certas operações matemáticas.
 - ▶ A funcionalidade correspondente a estas e outras operações não faz parte integrante da linguagem, mas está incluída numa biblioteca externa, bastante rica e padrão.
 - ▶ Todas essas operações são executadas por via da invocação de funções externas definidas nessa biblioteca standard.
 - ▶ Qualquer programador poderá desenvolver a sua própria biblioteca de funções, podendo até substituir algumas das funções standard, e também utilizar outras bibliotecas comerciais já existentes (p. ex. NAG, PHIGS etc.)
-



Características da linguagem C

- ▶ Algumas das características do C que levaram a que se tornasse uma das mais populares linguagens de programação, são:
 - ▶ Pequeno tamanho da sua definição
 - ▶ Subdivisão do código e grande utilização de funções
 - ▶ Alguma conversão automática entre tipos de dados (ao contrário do Pascal)
 - ▶ Linguagem estruturada
 - ▶ Disponibilidade de operadores para programação de baixo nível
 - ▶ Utilização fácil e extensa de apontadores para aceder memória, vetores, estruturas e funções
-



Características da linguagem C

- ▶ Algumas das razões que tornaram o C uma das linguagens prediletas dos programadores profissionais são:
 - ▶ A possibilidade de usar construções de alto nível
 - ▶ A possibilidade de utilizar operadores de baixo nível
 - ▶ A produção de código executável eficiente
 - ▶ A disponibilidade de compiladores em praticamente todos os sistemas de computação
 - ▶ Neste momento todos os compiladores seguem o *standard* internacional conhecido por ANSI C (*American National Standards Institute*).
-

