

MC-102 — Aula 01

Introdução à Programação de Computadores

Instituto de Computação – Unicamp

Primeiro Semestre de 2014

Roteiro

- 1 Por que aprender a programar?
- 2 Hardware e Software
- 3 Organização de um ambiente computacional
- 4 Algoritmos
- 5 Um pouco de história
- 6 A linguagem C
- 7 Relembrando
- 8 Informações Extras

Por que aprender a programar?

- Neste curso vocês aprenderão o básico para se criar programas para computador.
- Exemplos de programas: Firefox , Quake, MatLab, Media Player.
- Aprender a programar é uma atividade básica de um cientista ou engenheiro da computação.

Por que aprender a programar?

- *Eu não sou da computação !!!* Por que programar?
- Possíveis Respostas:
 - ▶ “I think everybody in this country should learn how to program a computer because it teaches you how to think. Computer science should be a liberal art” (Steve Jobs)
 - ▶ Porque é legal!
 - ▶ Posso resolver problemas complexos!
 - ▶ Posso ter algum retorno financeiro com isso!

Por que aprender a programar?

Eu sou das engenharias!

Alguns exemplos:

- Como engenheiro você deverá ser capaz de automatizar algum processo.
 - ▶ Você poderá criar programas para gerenciar e automatizar algum processo que hoje é manual.
- Como engenheiro você deverá ser capaz de desenvolver novas ferramentas ou protótipos.
 - ▶ Para criar ferramentas/protótipos você deverá fazer simulações computacionais para fazer testes preliminares.
- Você poderá enxergar situações onde uma solução computacional pode trazer benefícios.
 - ▶ Mesmo que você não implemente (programe) a solução você poderá propô-la e será capaz de “conversar” com o pessoal de TI para implementar a solução.

Por que aprender a programar?

Eu sou de uma área científica! Matemática, Estatística, Física, Química etc.

Alguns exemplos:

- Como cientistas vocês devem propor uma hipótese e testá-la.
 - ▶ Em vários casos onde os sistemas podem ser “modelados matematicamente”, são criados programas que fazem a simulação do sistema para verificação de uma hipótese.
- Você deverá resolver sistemas de equações complexos que não necessariamente podem ser resolvidos por softwares padrões (como MatLab).
 - ▶ Vocês deverão implementar seus próprios “resolvedores”.
- Simulações.
 - ▶ Muitos dos modelos propostos para explicar algum fenômeno são simulados computacionalmente. Implementar os modelos é uma tarefa básica.
 - ▶ Fluidodinâmica computacional, método de elementos finitos, simulação de mercado financeiro, análise de risco, negociadores automáticos, simulação do universo, ...

Por que aprender a programar?



The Making of a Fly: The Genetics of Animal Design (Paperback)

by Peter A. Lawrence

[Return to product information](#)

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about [Safe Online Shopping](#) and our [safe buying guarantee](#).

Price at a Glance

List Price: \$70.00
Used: from **\$35.54**
New: from **\$1,730,045.91**
Have one to sell? [Sell yours here](#)

[All](#)
[New](#) (2 from \$1,730,045.91)
 [Used](#) (15 from \$35.54)

Show ☒ New ☐ Prime offers only (0)

Sorted by Price + Shipping

New 1-2 of 2 offers

Price + Shipping	Condition	Seller Information	Buying Options
\$1,730,045.91 + \$3.99 shipping	New	Seller: profmath Seller Rating: *****: 93% positive over the past 12 months. (8,193 total ratings) In Stock. Ships from NO, United States. Domestic shipping rates and return policy. Brand new, Perfect condition, Satisfaction Guaranteed.	Add to Cart or Sign in to turn on 1-Click ordering.
\$2,198,177.95 + \$3.99 shipping	New	Seller: bordeebok Seller Rating: *****: 93% positive over the past 12 months. (125,891 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy. New item in excellent condition. Not used. May be a publisher overstock or have slight shelf wear. Satisfaction guaranteed!	Add to Cart or Sign in to turn on 1-Click ordering.

	profmath	bordeebok	profmath over previous bordeebok	bordeebok over profmath
8-Apr	\$1,730,045.91	\$2,198,177.95		1.27059
9-Apr	\$2,194,443.04	\$2,788,233.00	0.99830	1.27059
10-Apr	\$2,783,493.00	\$3,536,675.57	0.99830	1.27059
11-Apr	\$3,530,663.65	\$4,486,021.69	0.99830	1.27059
12-Apr	\$4,478,395.76	\$5,690,199.43	0.99830	1.27059
13-Apr	\$5,680,526.66	\$7,217,612.38	0.99830	1.27059

<http://www.michaeleisen.org/blog/?p=358>

O que esperar deste curso

- Vocês aprenderão o básico para desenvolver programas.
- Utilizaremos a linguagem C.
- Vocês **NÃO** vão aprender a usar programas neste curso (como office etc).
- Vocês **VÃO** ter porém, uma boa noção de como criar programas como o office.

O que será necessário

- Você deverá ter acesso a um computador.
- Para criar um programa, utilizamos um *editor de texto* (para escrever o código do programa) e um *compilador*.
- O compilador transforma o código em um programa executável.
- Se você usa linux ou MAC OS, você poderá utilizar qualquer editor simples como *emacs*, *kyle* etc. Será preciso instalar o compilador *gcc*.

O que será necessário

Para ir bem neste curso:

- Faça todos os laboratórios.
- Faça e implemente as listas de exercícios.
- E finalmente faça e implemente as listas de exercícios.

O que será necessário

Para ir bem neste curso:

- Faça todos os laboratórios.
- Faça e implemente as listas de exercícios.
- E finalmente faça e implemente as listas de exercícios.

O que será necessário

Para ir bem neste curso:

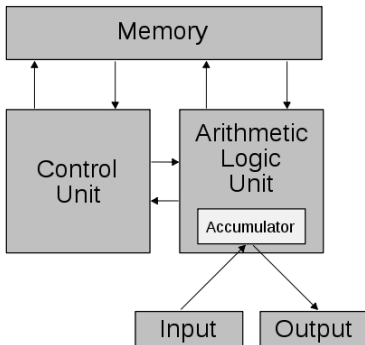
- Faça todos os laboratórios.
- Faça e implemente as listas de exercícios.
- E finalmente faça e implemente as listas de exercícios.

O que é um computador?

- Computador: o que computa, calculador, calculista. (dicionário Houaiss).
- Um computador é uma máquina que, a partir de uma entrada, realiza um número muito grande de cálculos matemáticos e lógicos, gerando uma saída.

Hardware e dispositivos

- Usualmente chamamos de *Hardware* todos os dispositivos físicos que compõem um computador.
- Temos por exemplo: CPU, Disco Rígido, Memória etc.
- Estes dispositivos seguem uma organização básica como na figura (Arq. de Von Neumann).



Hardware e dispositivos

Todo o hardware opera com sinais digitais: sem energia e com energia. Normalmente usamos valores 0 e 1.

- Chamamos estes sinais de Bit → Valores 0 ou 1.
- Chamamos de *Byte* → um agrupamento de 8 bits.
- Todas as informações armazenadas no computador são representados por números 0s e 1s (letras, símbolos, imagens, programas etc).

Software

São os programas que executam tarefas utilizando o hardware de um computador.

- Os softwares são compostos por um conjunto de instruções que operam o hardware.
- Temos abaixo, por exemplo, três instruções para um computador de 32 bits.
- Um software é composto por milhares de instruções deste tipo.

```
0100 0010 0011 0101 0101 0100 0011 0110
0100 1110 1100 1100 1001 0110 0110 1000
0000 0101 1111 1110 1101 0011 0000 1100
```

Organização básica de um ambiente computacional

- Um ambiente computacional é organizado como uma hierarquia de funções, onde cada uma é responsável por uma tarefa específica.

Programas de Aplicação
Compiladores
Sistema operacional
Hardware

Organização básica de um ambiente computacional

Programas de Aplicação.

- Como usuários, interagimos com os programas de aplicação.
- Neste curso iremos descer nesta hierarquia, para construirmos novos programas de aplicação.
- Para construir novos programas podemos escrever diretamente códigos digitais que serão executados por um computador.
- Uma maneira mais simples é usar um compilador para uma linguagem de programação específica.

Programas de Aplicação
Compiladores
Sistema operacional
Hardware

Organização básica de um ambiente computacional

Compiladores e Linguagens de Programação.

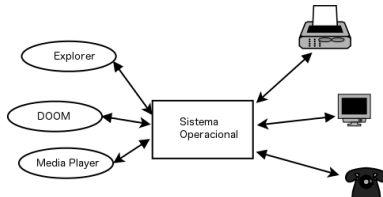
- Uma linguagem de programação é um conjunto de comandos que são mais “próximos” da linguagem humana do que os sinais digitais.
- Neste curso estamos interessados no estudo da *linguagem de programação C*.
- Um *compilador* é um programa que lê um código de uma linguagem de programação e o transforma em um programa executável.
- Na verdade o compilador realiza esta tarefa juntamente com um *assembler*.

for(i=0; i< 10; i++)	loop: add c, a, b	0100 0010 0011 0101 0101 0101
c = a+b;	add i, i,1	0110 0110 0111 0101 0101 0101
	bnq i, 10, loop	1111 0000 0111 0101 0101 0101

Organização básica de um ambiente computacional

Sistema Operacional.

- Os programas possuem instruções que são executadas no hardware.
- Mas o acesso ao hardware, como disco rígido, memória, processador, é controlado por um software especial conhecido como *sistema operacional*.
- O *sistema operacional* é o responsável pelo controle de hardware, segurança dentre outros.
- Exemplos de sistema operacional: *Windows, Mac OS, Linux, Android, iOS*.



Algoritmos

Ao criarmos um programa para realizar uma determinada tarefa devemos ser capazes de construir algoritmos.

- Algoritmo: Seqüência de passos, precisos e bem definidos, para a realização de uma tarefa.
- Algoritmos podem ser especificados de várias formas, inclusive em português.

Exemplo de algoritmo

Como calcular 2345×4567 usando lápis, papel e uma tabuada?

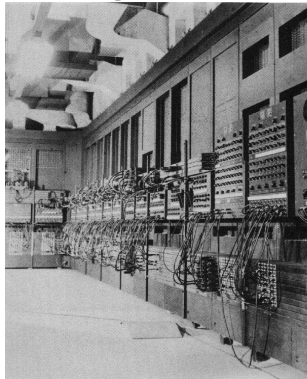
De algoritmos a programas

- Neste curso vamos aprender a criar algoritmos simples.
- Usaremos a linguagem C para descrever os algoritmos.
- Após compilar os programas escritos em C, teremos um programa para realizar a tarefa especificada.

Um pouco de história

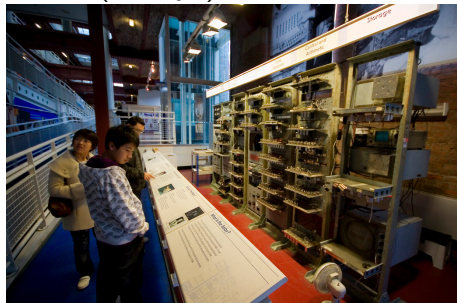
Os primórdios da programação: programação em código absoluto ou binário (apenas 0s e 1s).

ENIAC



University of Pennsylvania, USA

SSEM ("Baby")



University of Manchester, UK

Um pouco de história

Uma melhoria: A Linguagem Assembly.

- Cria-se uma linguagem de baixo nível (Linguagem Assembly) para representar as instruções em código binário.
- Um programa, chamado montador ou assembler, faz a transformação em código absoluto.

```
LOOP:  MOV A, 3  
       INC A  
       JMP LOOP
```

Um pouco de história

Uma brilhante idéia: Criação de linguagens de alto nível e compiladores para estas.

- Mais distantes da máquina e mais próximas de linguagens naturais (inglês, português, etc.).
- Mesmo mais compreensíveis, elas não são ambíguas.
- Um compilador as transforma em código executável.

Exemplos de linguagens

- C
- Pascal
- Java

Primeiro programa em C

Um programa em C é um arquivo texto, contendo declarações e operações da linguagem. Isto é chamado de *código fonte*.

```
#include <stdio.h>

main() {
    printf("Hello, world!\n");
}
```

Como executar este programa

- Para executar um programa a partir do seu código fonte é necessário compilá-lo, gerando **código binário** ou **executável**.
- Este pode ser executado como qualquer outro programa de aplicação.

```
$ gcc hello.c -o hello
```

```
$ hello
```

```
Hello, world!
```

Relembrando

- Hardware e Software.
- Pilha de um ambiente computacional: Programas de Aplicações, Compilador, Sistema Operacional, Hardware.
- Código Binário, Assembly, Linguagem de Alto Nível.
- Algoritmos.

Informações Extras: O que são erros de compilação?

Caso o programa não esteja de acordo com as regras da linguagem, erros de compilação ocorrerão. **Ler** e entender estes erros é muito importante.

```
#include <stdio.h>
main() {
    printf("Hello, world!\n");
```

```
$ gcc hello.c -o hello
hello.c: In function 'main':
hello.c:5: error: syntax error at end of input
```

Informações Extras: O que são erros de execução?

Acontecem quando o comportamento do programa diverge do esperado e podem acontecer mesmo quando o programa compila corretamente.

```
#include <stdio.h>
main() {
    printf("Hello, world! $#%#@%\n");
}
```

```
$ gcc hello.c -o hello
```

```
$ hello
```

```
Hello, world! $#%#@%
```

Informações Extras: O que é um depurador?

- Ferramenta que executa um programa passo a passo.
- Ajuda a encontrar erros de execução (bugs).

Exemplo

- gdb