Arquitetura de Computadores

LEI – 2022/23 - DI-FCT/NOVA

Aula T1 – 03 de março de 2023

Sumário

Apresentação.

Introdução à arquitetura de Von Neumann.

O Sistema operativo

Arquitetura de Computadores →AC

Apresentação

- Objetivos da cadeira
- Docentes
- Programa
- Funcionamento e trabalho dos alunos
- Avaliação
- Bibliografia

Objetivos

- Gerais:
  - Conhecimento!...
  - ...competência e produtividade
  - ...vantagem competitiva no mercado de trabalho
- Específicos de AC:
  - Compreender funcionamento dos computadores e como estes conseguem executar os programas
    - Organização e funcionamento interno dos vários componentes do computador
    - Níveis de abstração, interfaces de programação, mecanismos de execução suportados pelo *hardware*

A equipa

Pedro Medeiros (regente/responsável)

Maria Cecília Gomes

Kevin Gallagher

Henrique Ferreira

Programa de AC

- Níveis de um sistema computacional. O nível do Sistema operativo.
- Componente CPU
- Componente memória central (RAM)
- Componente sistemas de entradas / saídas

Laboratórios:

- LINUX
- Linguagem C
- Assembly Pentium

AC no curso (u.c. relacionadas)

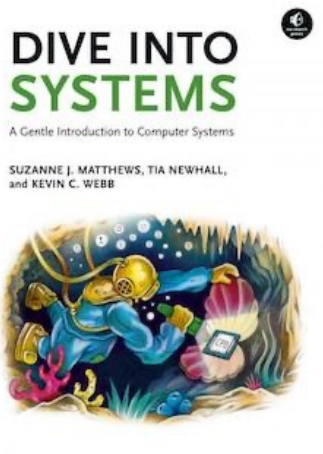
- 1º semestre:
  - Introdução à Programação
  - Sistemas Lógicos
- 2º semestre:
  - **Arquitetura de Computadores**
- Seguintes no 1º ciclo do MIEI (futura LEI):
  - Fund. Sistemas de Operação, Ling. Ambientes de Programação, Redes de Computadores

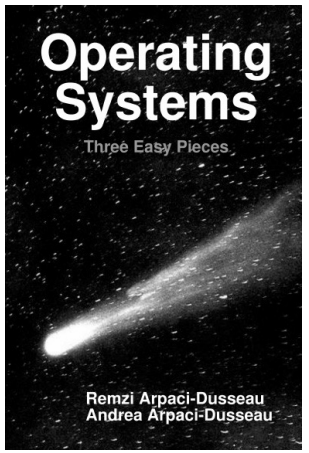
Bibliografia

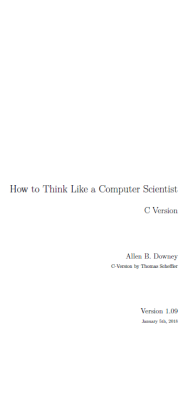
- **Dive Into Systems**: **A Gentle Introduction to Computer Systems** Suzanne J. Matthews, Tia Newhall e Kevin C. Webb, *No Starch Press,2022* *https://diveintosystems.org/book/*

- **Operating Systems: Three Easy Pieces**, Remzi Arpaci-Dusseau e Andrea Arpaci-Dusseau, *Lulu Books, 2013* *https://pages.cs.wisc.edu/~remzi/OSTEP/*

How to Think Like a Computer Scientist - C Version, Allen B. Downey (1999) , Thomas Scheffler (C version 2018) *https://freecomputerbooks.com/books/Think-C_v1.09.pdf*







Trabalho do Aluno

- 9 ECTS * 28 horas / ECTS = **252 horas**
- Horas em contacto
  - Aulas teóricas: 14 semanas * 3 h /semana
    =42 horas
  - Aulas práticas: 14 semanas * 2 h /semana
    =28 horas
- Horas em autonomia
  - Estudo: 130 horas
  - Trabalhos : 40 horas
- Avaliações 6 horas
- Total (**240 horas = 15 semanas*16 horas**)

Avaliação – Regras Gerais

- A avaliação tem duas componentes:
  - a componente teórico-prática
  - a componente laboratorial
- A frequência é conseguida através da obtenção de 6,0 valores na componente laboratorial (ver à frente)
- Fraude (detetada imediatamente ou *a posteriori*)
  - Qualquer ato que vicie o processo de avaliação (copiar, ceder cópia, assinar trabalho que não fez, etc.), pode implicar a reprovação imediata de TODOS os alunos envolvidos (os que copiaram e os que deram a copiar)

Componente Teórico-Prática

- 2 Testes Presenciais (2h de duração): incluem perguntas sobre os trabalhos práticos. Datas a confirmar
- Exame Presencial
- Nota da Componente Teórico-Prática (CompTP):
  - CompTP = (Nota_Teste1 + Nota_Teste2) /2 ou
  - CompTP = Nota_Exame
- Para obter aprovação:
  - CompTP ≥ 9.5

Componente Laboratorial

- 4 trabalhos EP1, EP2, EP3 e EP4 (grupos de 2 alunos)
  - Entrega eletrónica: data a anunciar (EP1 e EP2 antes do teste 1, EP3 e EP4 antes do teste 2)
  - **Discussão virtual** – através de perguntas nos testes. Ver detalhes no CLIP
- Nota da Componente Laboratorial (CompL):
  - CompL = (Nota EP1 + Nota EP2 + Nota EP3 + Nota EP4) / 4
- Para obter aprovação e ter frequência:
  - CompL ≥ 6.0

Avaliação (Nota final)

- Nota final (NF) dos alunos
  - NF = CompTP (se CompTP < 9.5)
  - NF = CompL (se CompL < 6.0)
  - NF = 0.25 CompL + 0.75 CompTP

Notas Anteriores

- Os alunos que obtiveram nota CompL superior a 6.0 em 2021/22:
  - Estão dispensados de realizar a componente laboratorial; podem repeti-la, inscrevendo-se nos turnos práticos.
  - Se desistirem, será usada a CompL obtida anteriormente.

Informação

- **CLIP**
  - Sumários (com os elementos de estudo correspondentes), Slides, Enunciados, Outros elementos

  Ligar as notificações no CLIP

- **Outros meios** ??

Sumário

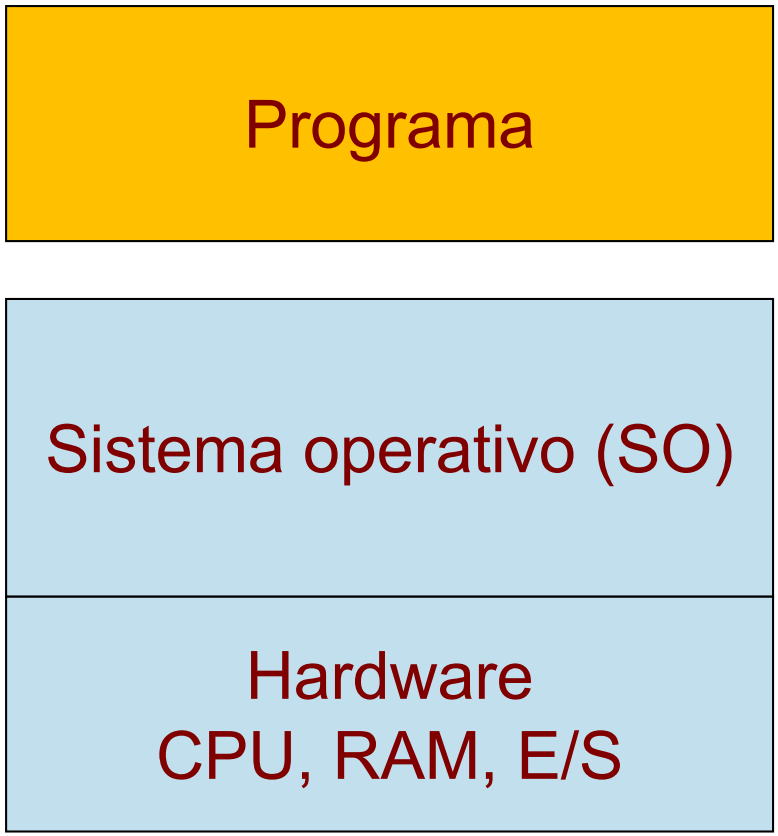
Apresentação.

Introdução à arquitetura de Von Neumann.

O Sistema operativo

Bibliografia:
Livro ***Dive Into Systems***, Capítulo 0, Capítulo 5 secção 2

Sistema Computacional



Hardware

Execução de instruções armazenadas na memória e envio de ordens para o teclado, ecrã e disco

Memória

Armazenamento volátil de bytes que representam instruções e dados

Unidade Central de Processamento (CPU)

Entrada de dados (e programas)

Eletrónica de controlo do disco

Saída de resultados

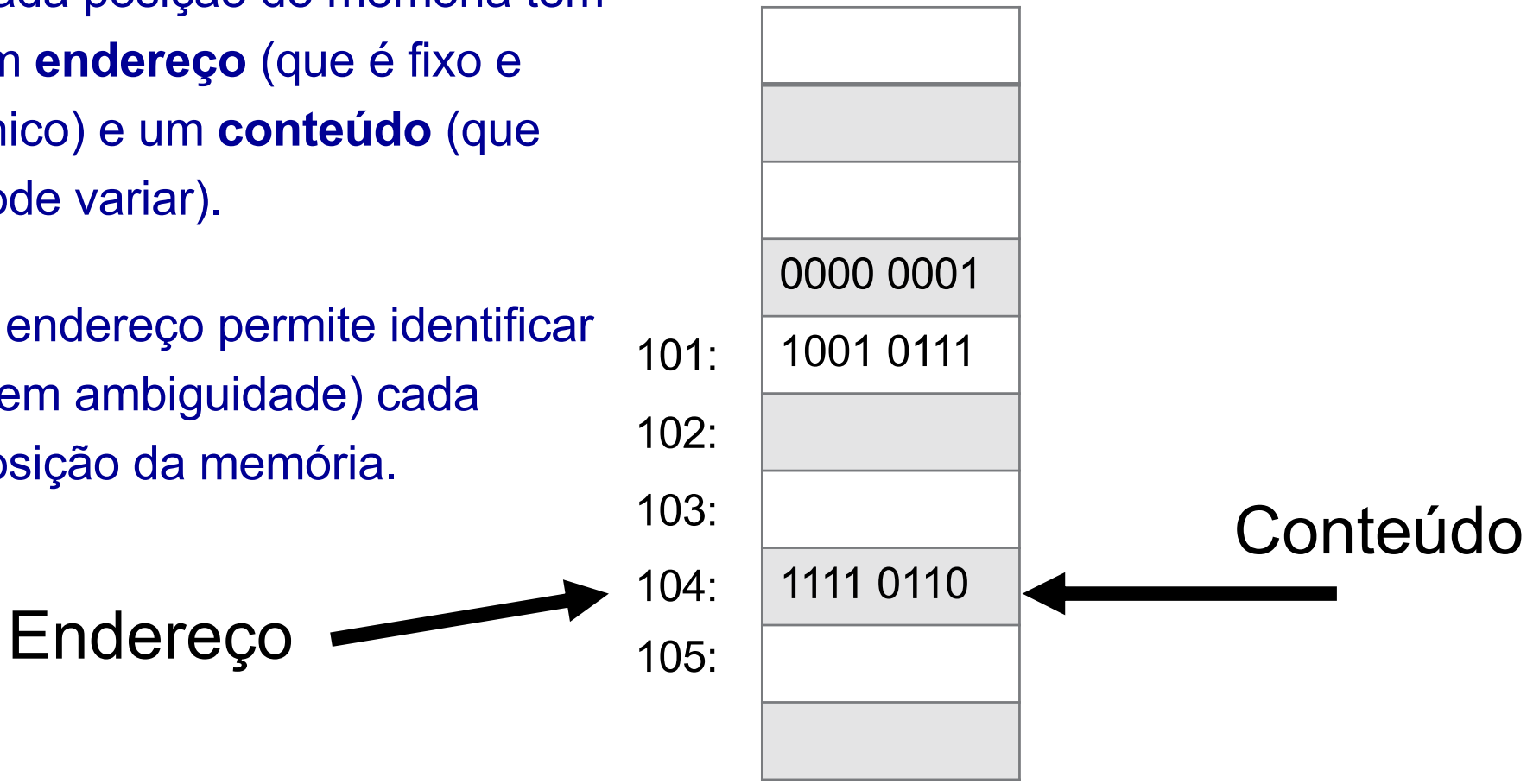
Eletrónica de controlo do teclado

Eletrónica de controlo do ecrã

Armazenamento permanente de bytes que representam dados e programas

Memória central (RAM

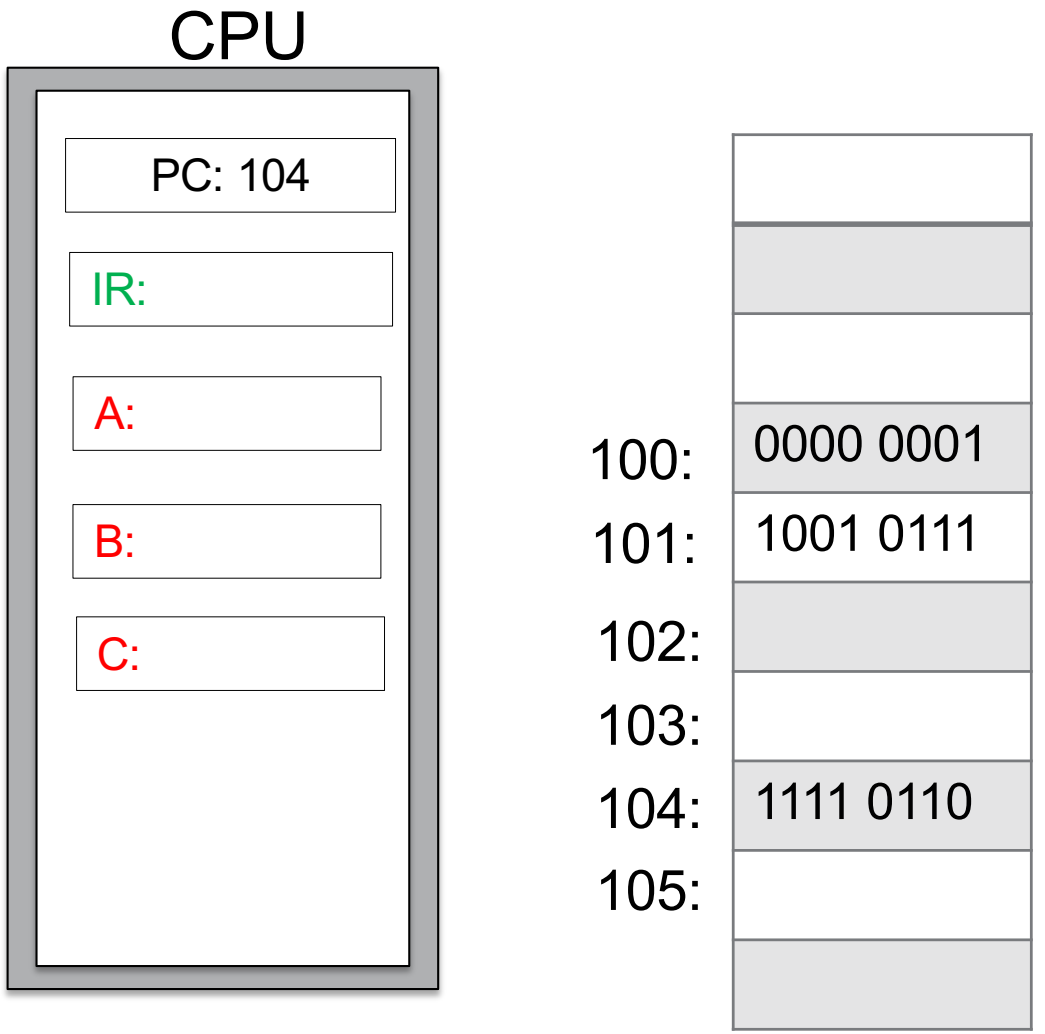
- Cada posição de memória tem um **endereço** (que é fixo e único) e um **conteúdo** (que pode variar).
- O endereço permite identificar (sem ambiguidade) cada posição da memória.



- O conteúdo da posição de memória com o endereço 104 é 1111 0110

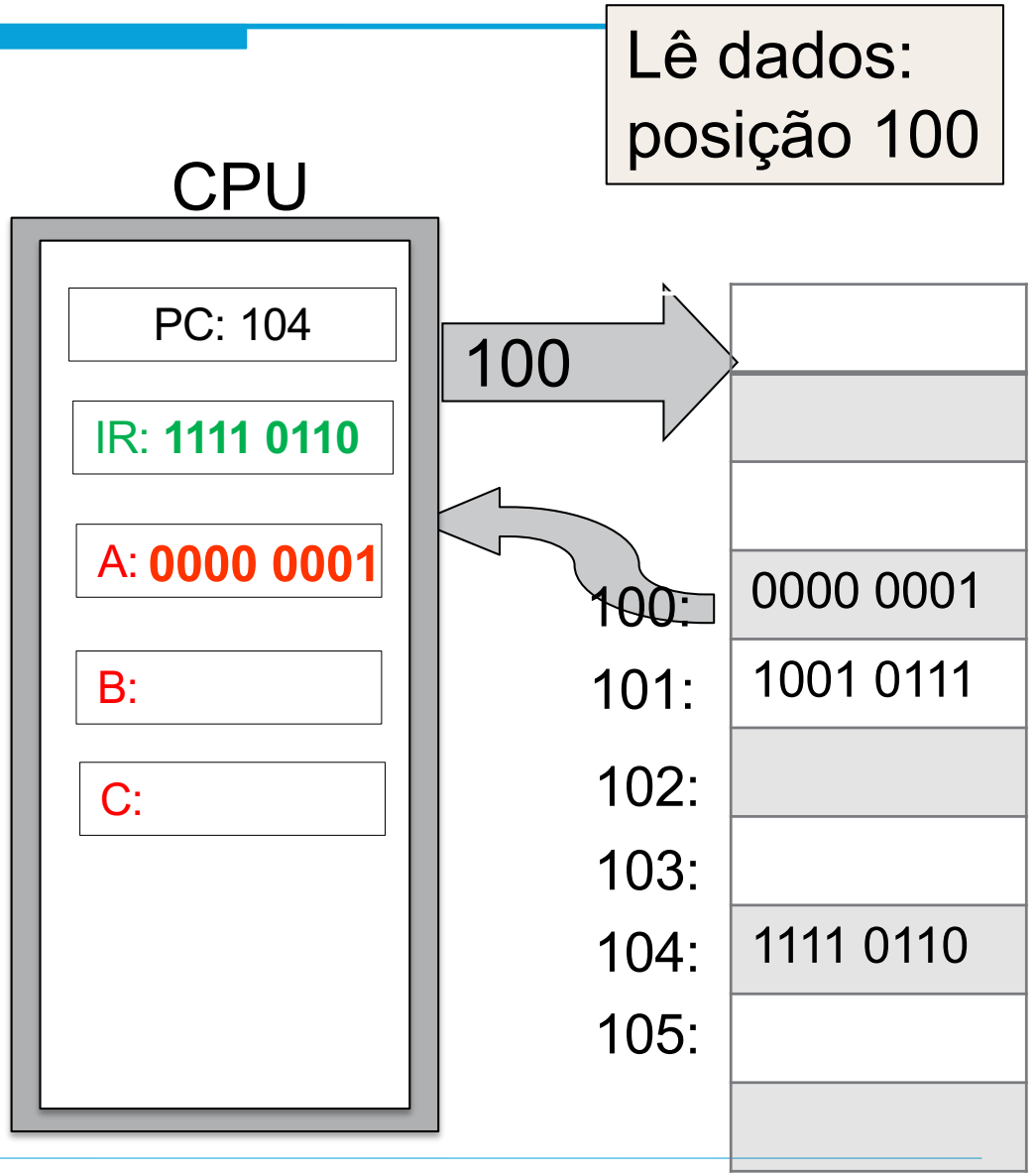
Funcionamento do CPU

- O CPU executa as instruções guardadas na memória central, de uma forma sequencial.
- Em cada momento, o CPU mantém a posição de memória da instrução que está a executar.



Funcionamento do CPU

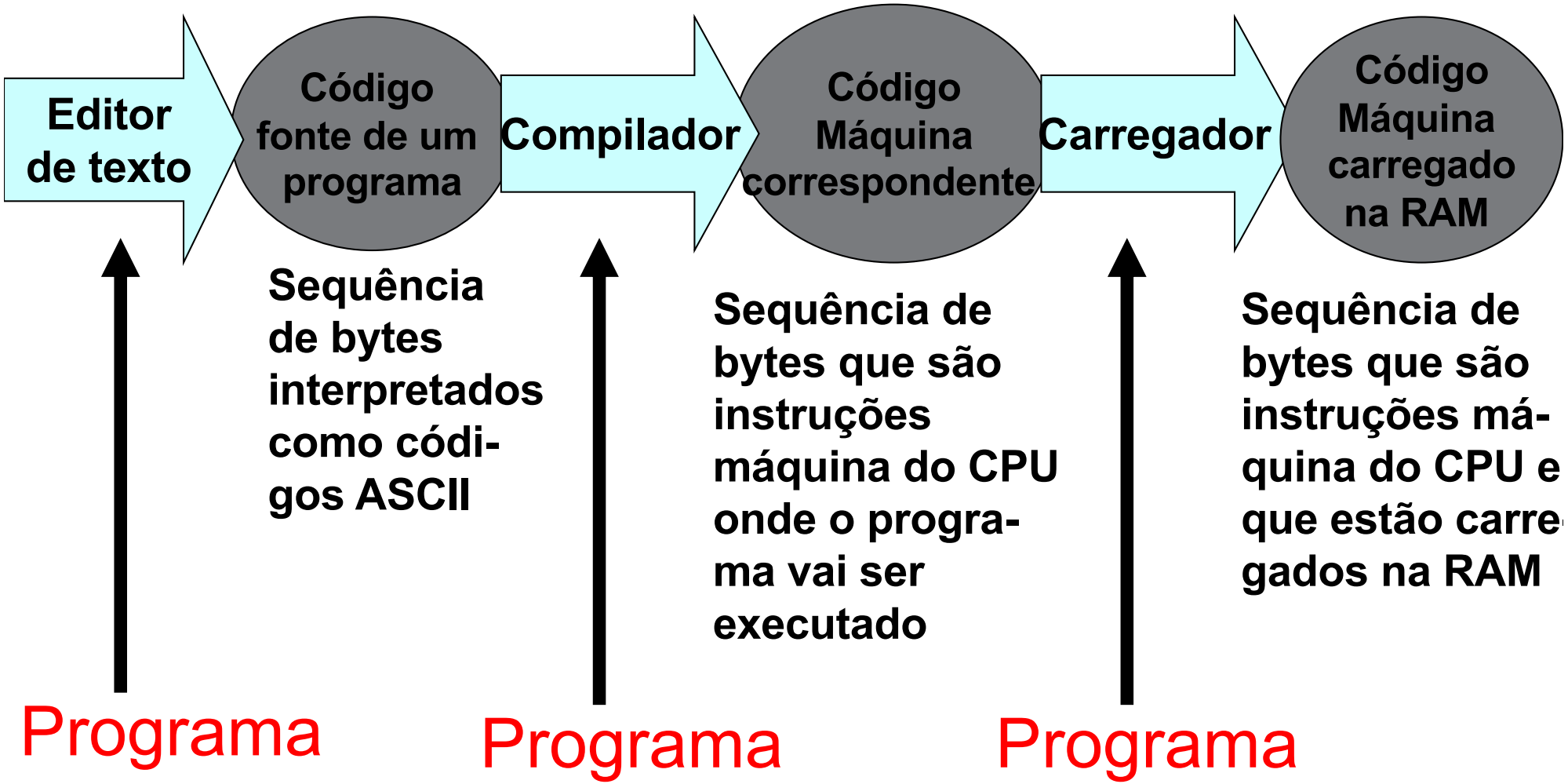
- A instrução define a ação elementar a executar
  - Ações atuam sobre dados armazenados em memória central ou num dispositivo de entrada/saída.
- Exemplo
  somar 100 101 102
  - Soma o conteúdo das posições 100 e 101 e armazena o resultado na posição 102.



O que é necessário para executar aplicações (e desenvolvê-las)

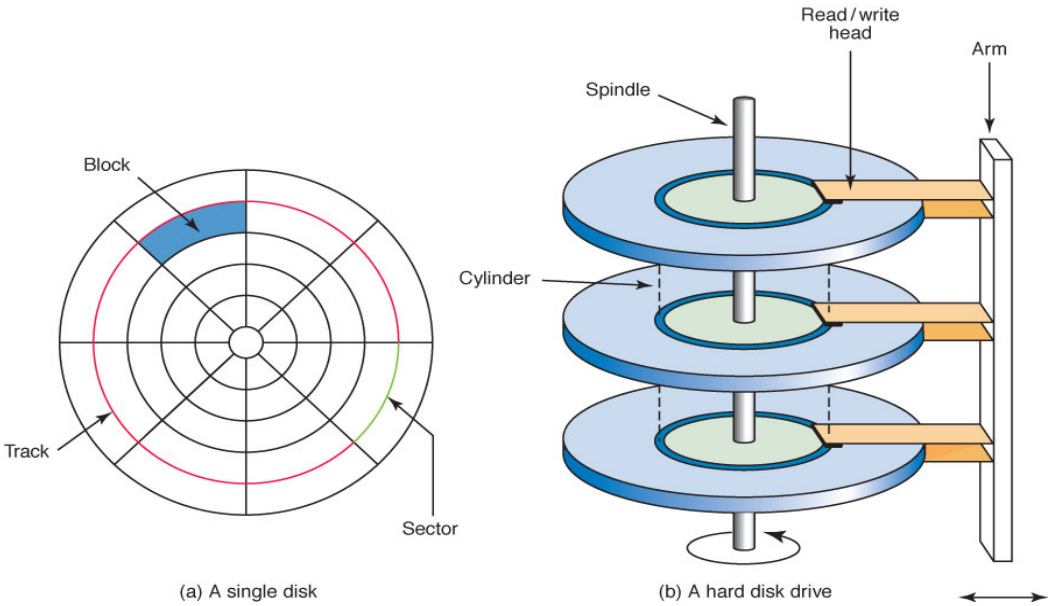
- Hardware
  - CPU + Memória
  - Dispositivos de entrada-saída
    - Interacção: teclado, rato, ecrã
    - Arquivo: discos
- Software de sistema
  - Compiladores / interpretadores
  - Interpretador de comandos (incluindo carregador)
  - Sistema operativo (ou de operação)
    - Suporte das chamadas ao sistema feitas pelas aplicações e pelos programas de sistema

Ciclo de vida de um programa



Onde guardar o código fonte e os programas?

- Em sectores do disco, porque é um dispositivo de armazenamento permanente
- Os discos são complicados!
- Disco
  - S superfícies, cada com P pistas, cada com S sectores
  - Pode ser visto como contendo N blocos todos do mesmo tamanho (por ex: 1024 bytes)



- Ficheiro: abstracção fornecida pelo sistema operativo que permite esconder a complexidade do disco
  - Nome que é uma cadeira de caracteres (ASCII)
  - Operações: abrir, fechar, ler, escrever
  - Atributos (tamanho, data de criação, …)
  - Blocos do disco onde está guardado (o último não está normalmente todo preenchido)

Ciclo de vida de um programa agora com ficheiros

Programador

Editor de texto

Ficheiro fonte do programa

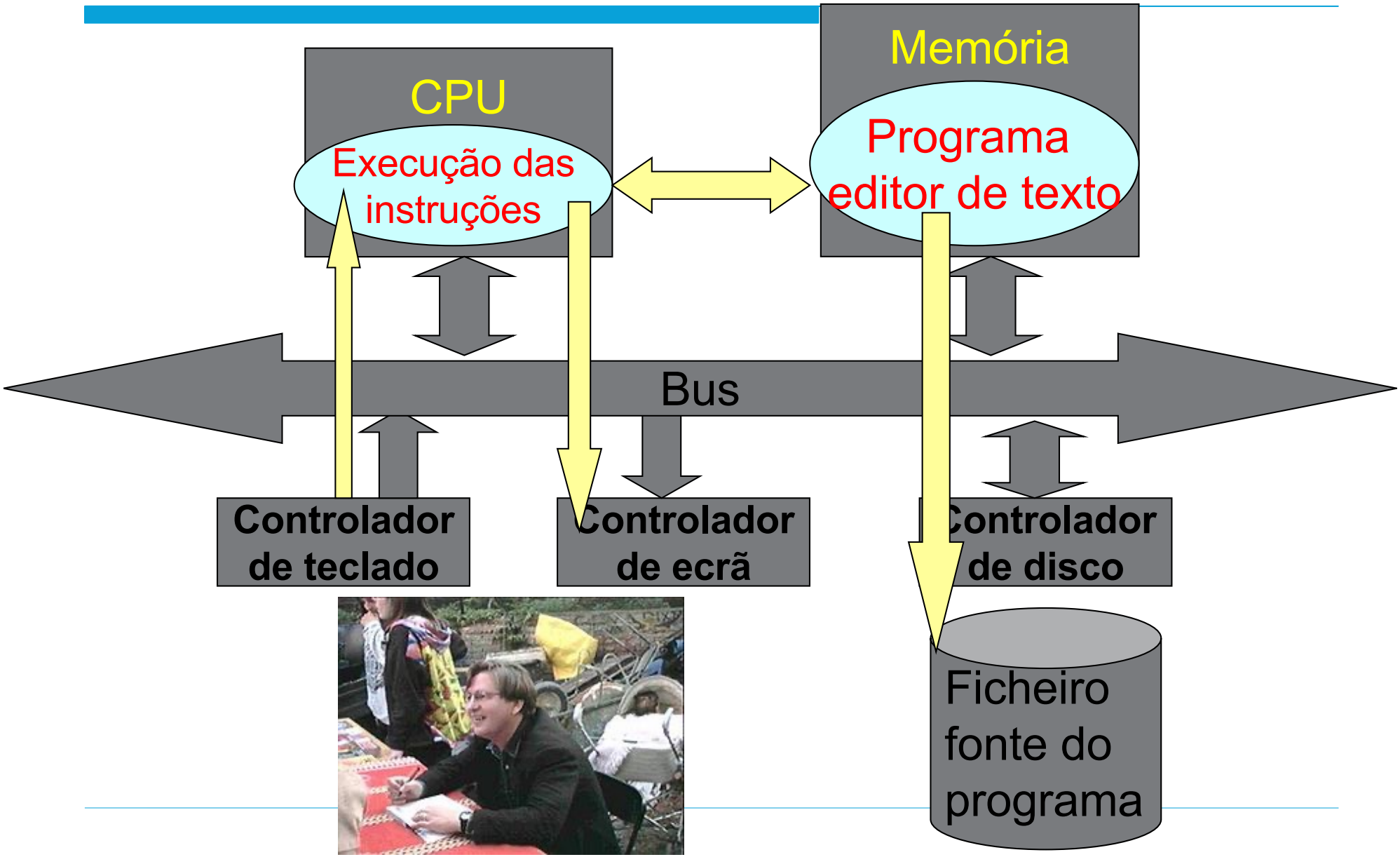
Compilador

Ficheiro executável

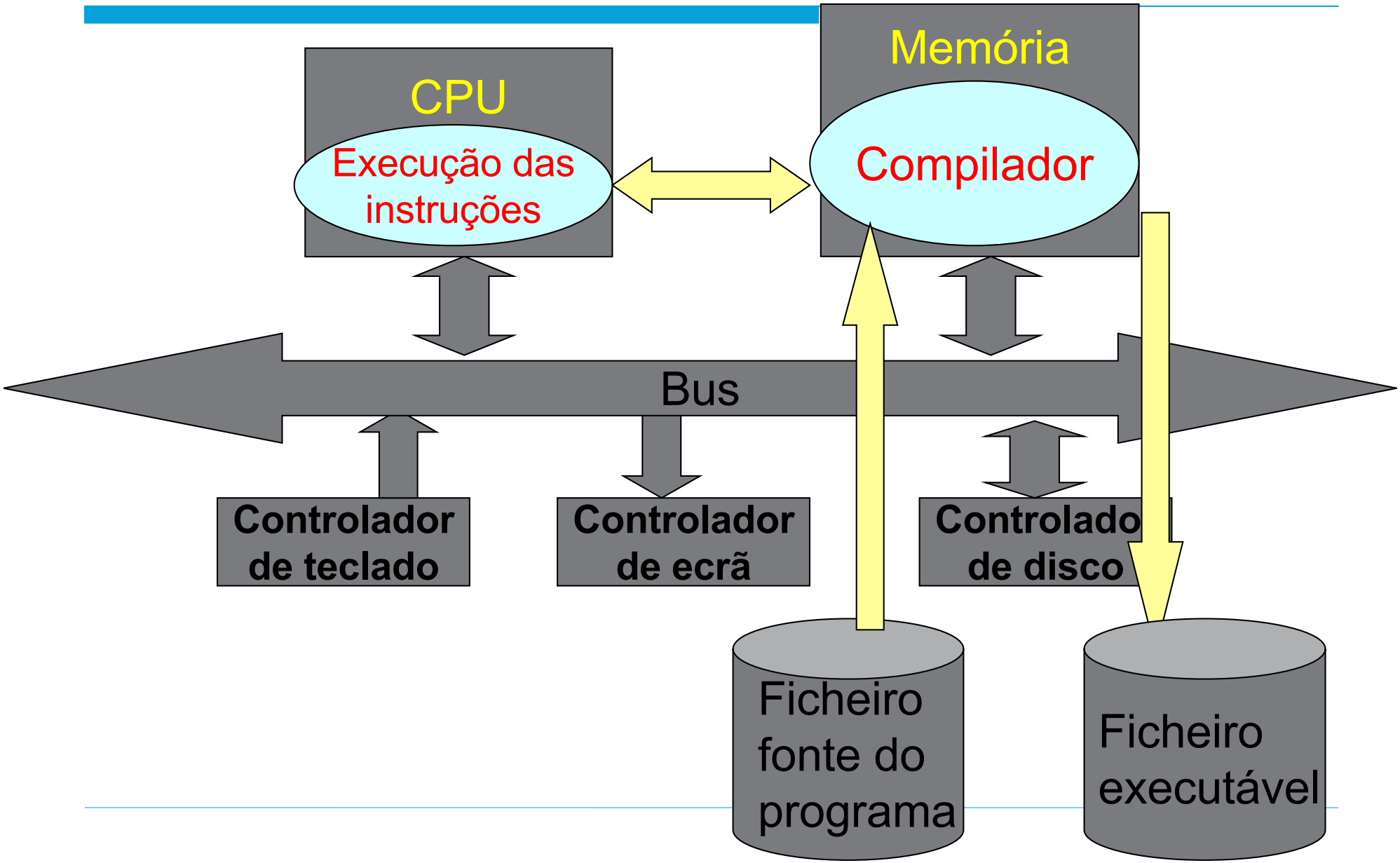
Carre-gador

Código carregado na RAM

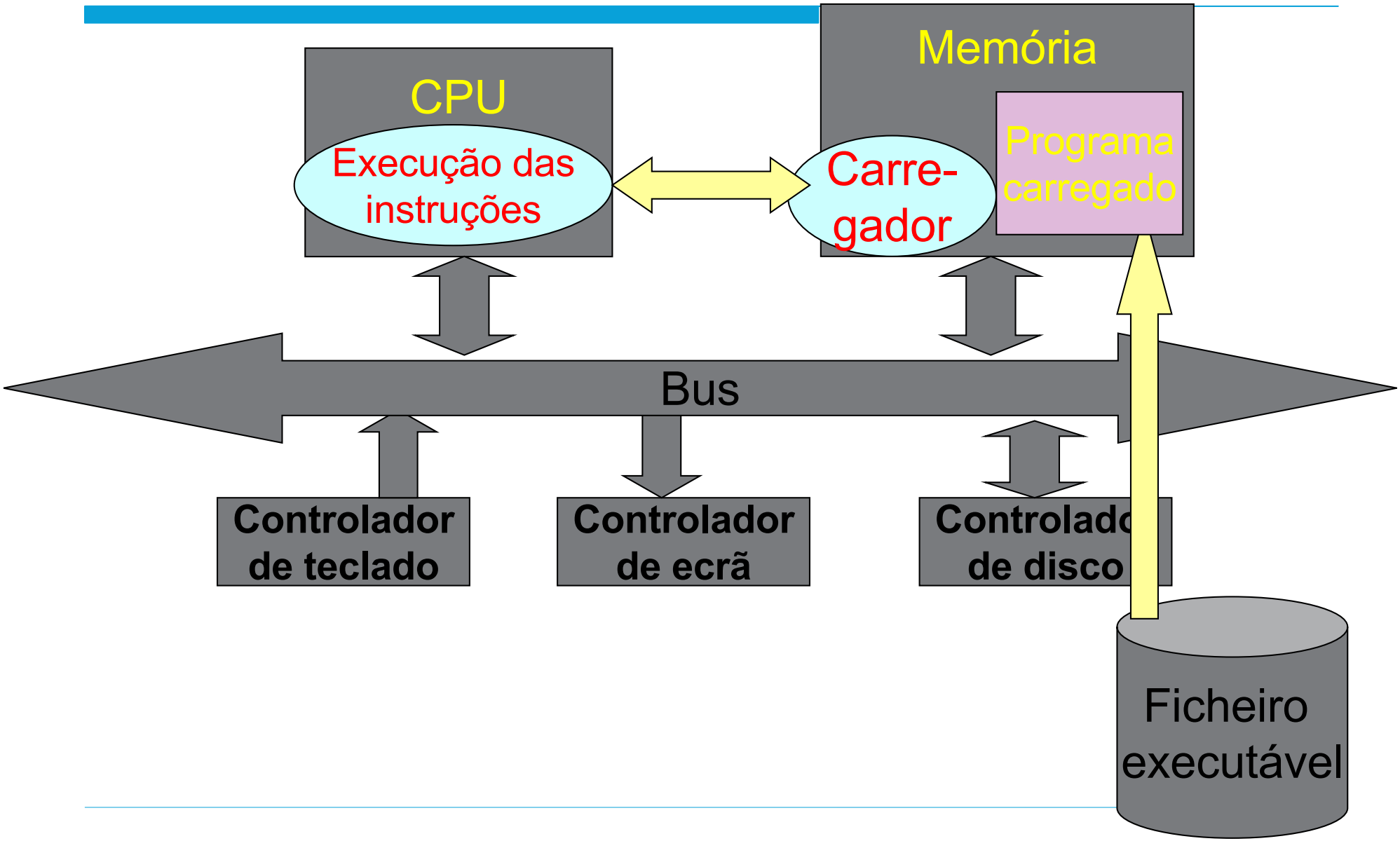
Edição do programa



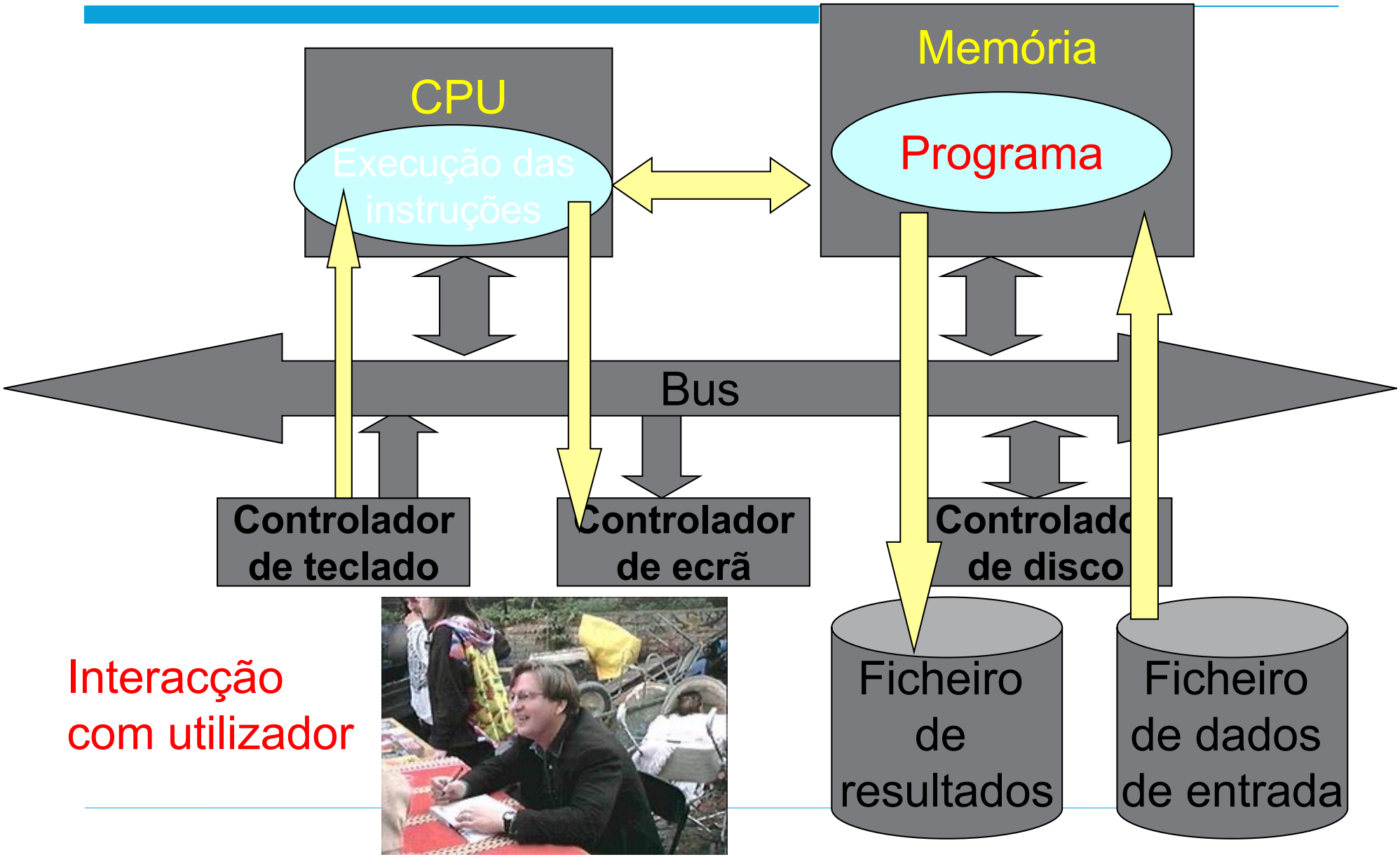
Geração do código



Carregamento



Execução



Interacção com utilizador

O que é que é preciso?

- Conjunto de código que implemente funções de acesso aos periféricos e aos ficheiros
  - Operações simples
  - Independentes do hardware
- Carregador
  - Usando as funções anteriores carrega ficheiros executáveis em memória
- Interpretador de comandos
  - Usando as funções anteriores lê comandos do teclado e carrega programas para os executar

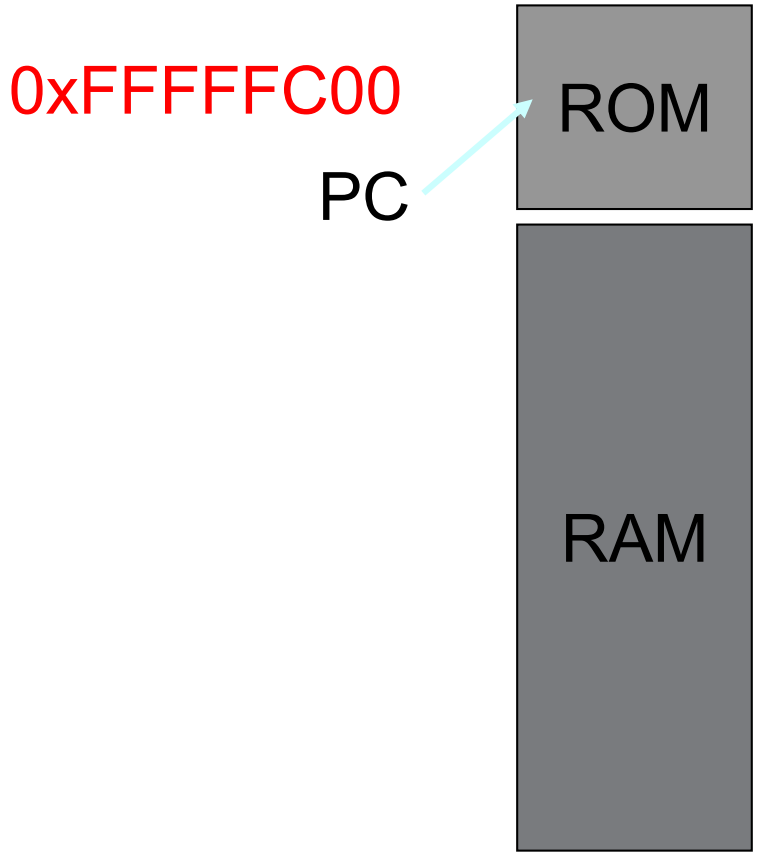
Sistema de operação (ou operativo) – abrev. SO

Funções de um SO “simples”

- Um único utilizador, um programa em execução de cada vez
- Funções essenciais
  - Supervisonar a utilização dos recursos do sistema
  - Controlar os periféricos
  - Gestão da memória central
  - Gestão de ficheiros
  - Suporte da interacção com o utilizador

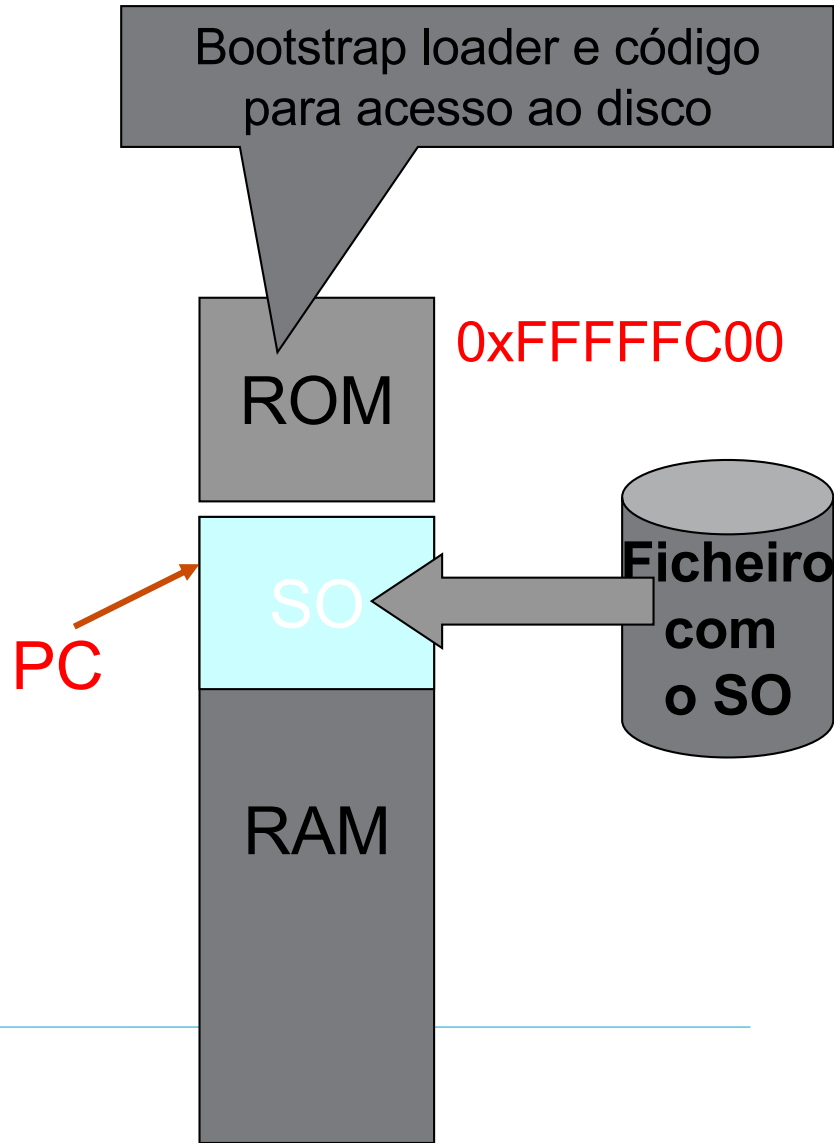
Quando a energia é ligada salta-se para o código do SO (1)

- Como?
  - Quando há um “power on” ou um “reset” a máquina de controlo do CPU carrega o PC com um valor fixo F (por ex. 0xFFFFFC00)
  - Nessa zona de memória, há uma ROM (ie uma memória não volátil); essa ROM contém o código adequado



Quando a energia é ligada salta-se para o código do SO (2)

- É mais habitual o código que começa em F carregar o código do SO a partir de um ficheiro no disco (bootstrap loader)
- O bootstrap loader carrega o SO para a "parte de cima" da RAM
- Quando o carregamento do SO acaba o "bootstrap loader" carrega o PC com o endereço da rotina de inicialização do sistema
- Quando o código de inicialização do SO acaba o PC salta para o código do interpretador de comandos



Inicialização do S.O.

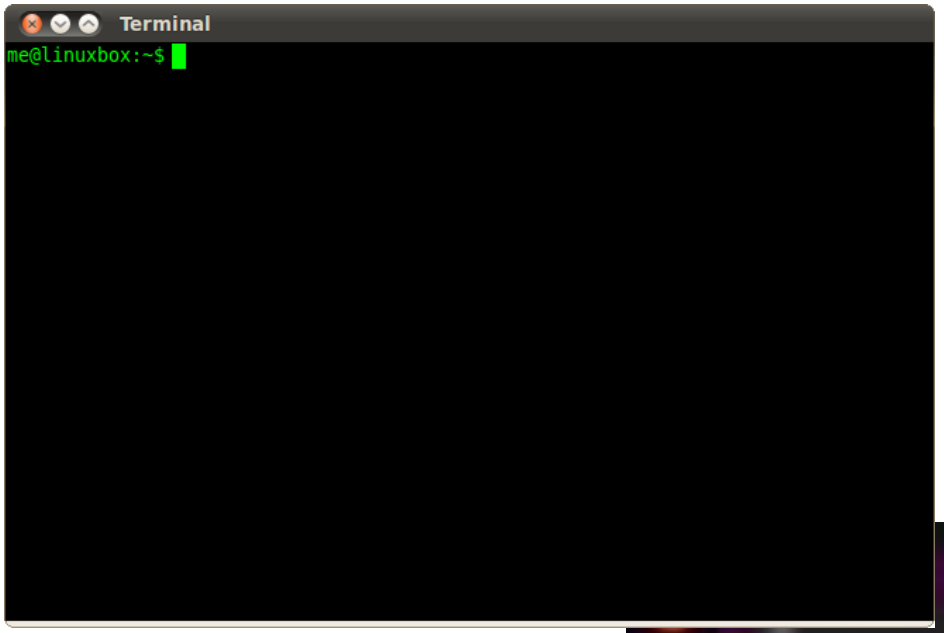
- A parte do SO que reside sempre em memória chama-se núcleo ou *kernel* do SO
- Inicialização do hardware
  - Controladores dos periféricos
- Inicialização das estruturas de dados
  - representam os vários recursos de sistema
  - Usados pelos algoritmos de gestão desses recursos
- Execução do interpretador de comandos
  - ***Espera por interacção do utilizador***

Interpretador de comandos

```
while(TRUE) do
      - apresentar prompt
      - ler comando
      - executar comando
```

- Vários nomes e formas:
  - Orientado para linha de comando
    - Shell (UNIX), cmd.com (Windows)
  - Janelas, menus, rato …
    - Windows, Macintosh, X-Windows (Linux/UNIX)

Interpretador de comandos



Linha de comando

Gráfico