
Arquitetura de Computadores

LEI – 2021/22

DI-FCT/NOVA

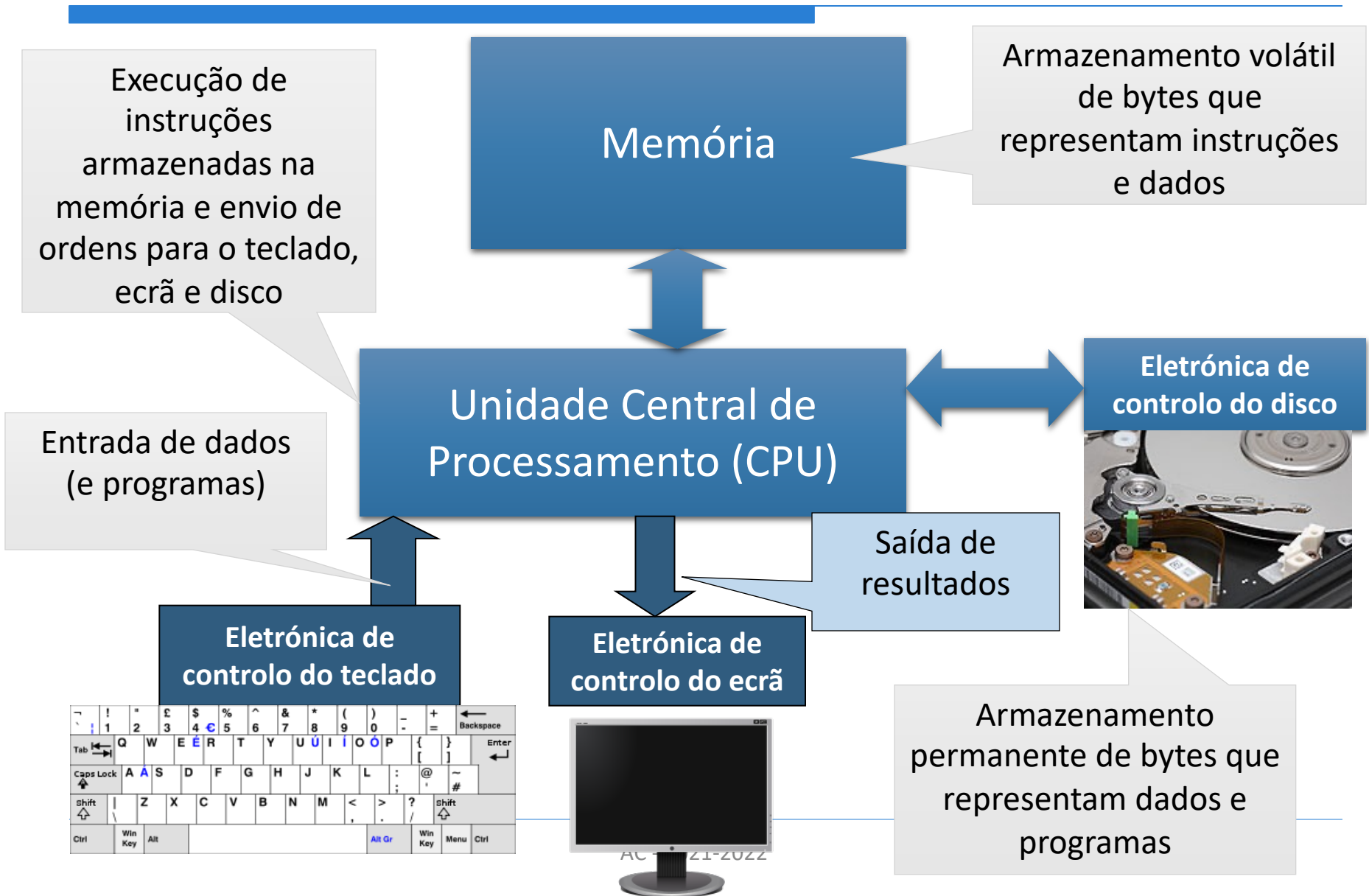
Aulas 1 e 2

Sumário

- Introdução à arquitetura de Von Neumann
- O Sistema Operativo

Arquitetura de Von Neumann

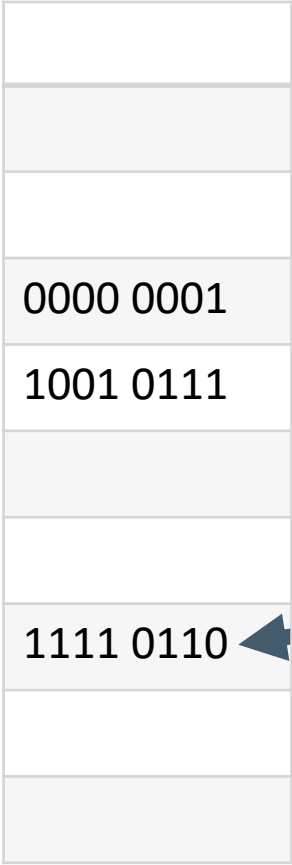
Hardware



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.

Endereço



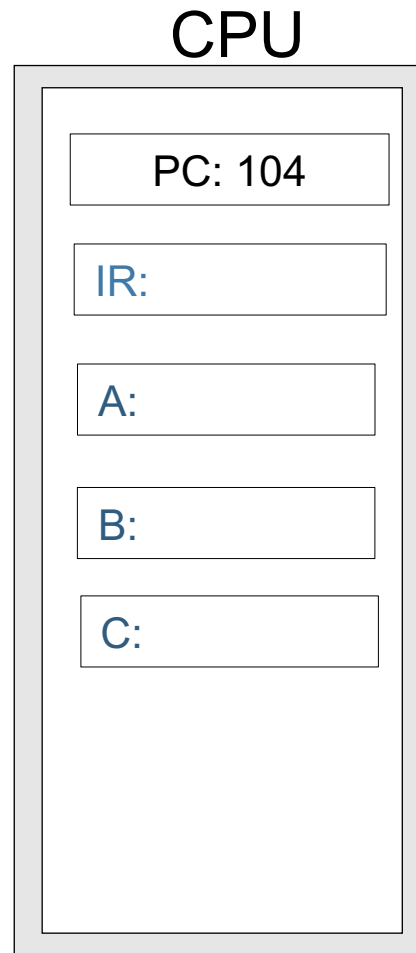
A vertical stack of memory cells. The cells are represented as a column of boxes. The 4th box from the top contains the binary value '0000 0001'. The 5th box contains '1001 0111'. The 6th box is empty. The 7th box is empty. The 8th box contains '1111 0110'. The 9th box is empty. The 10th box is empty. To the left of the boxes, addresses are listed: '101:' next to the 5th box, '102:' next to the 6th box, '103:' next to the 7th box, '104:' next to the 8th box, and '105:' next to the 9th box. A blue arrow labeled 'Endereço' points to the address '104:'. Another blue arrow labeled 'Conteúdo' points to the binary value '1111 0110' in the 8th box.

	0000 0001
101:	1001 0111
102:	
103:	
104:	1111 0110
105:	

- 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.

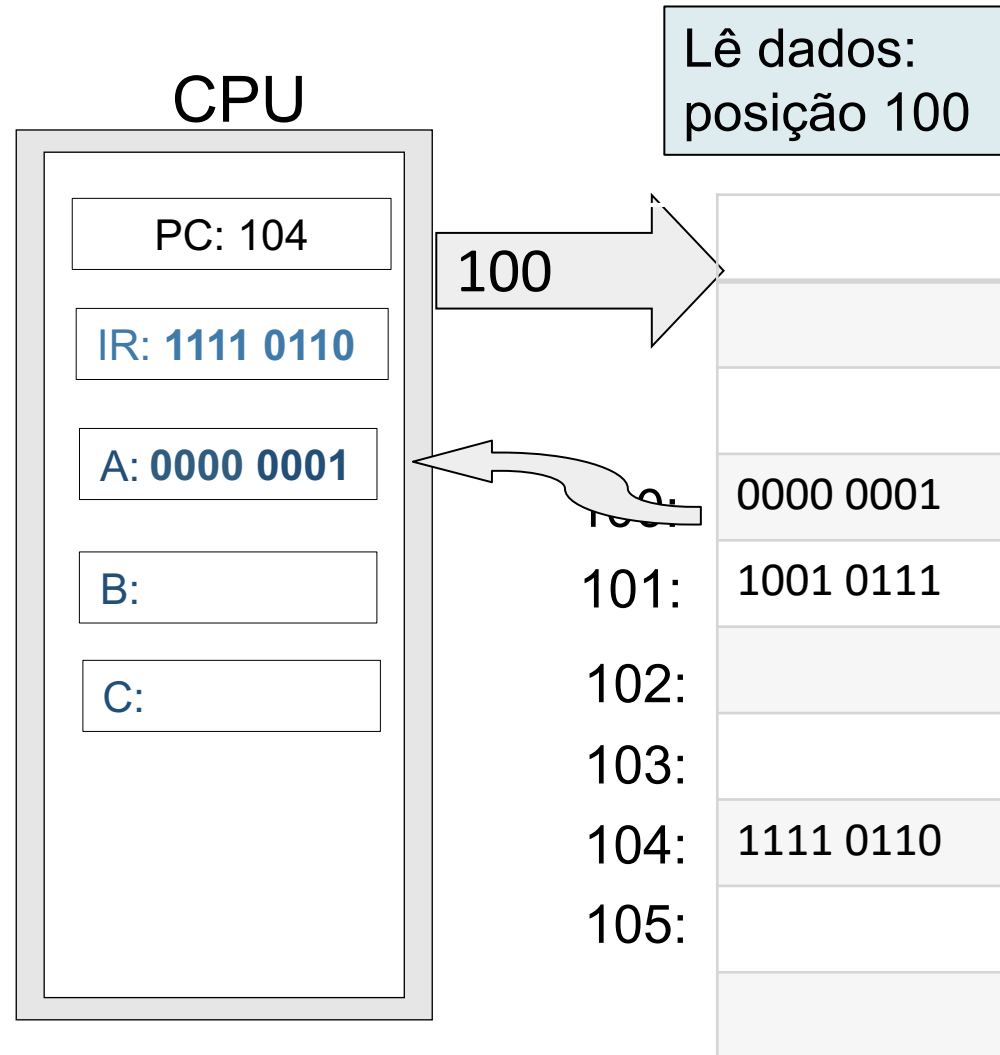


Memória (RAM)

100:	0000 0001
101:	1001 0111
102:	
103:	
104:	1111 0110
105:	

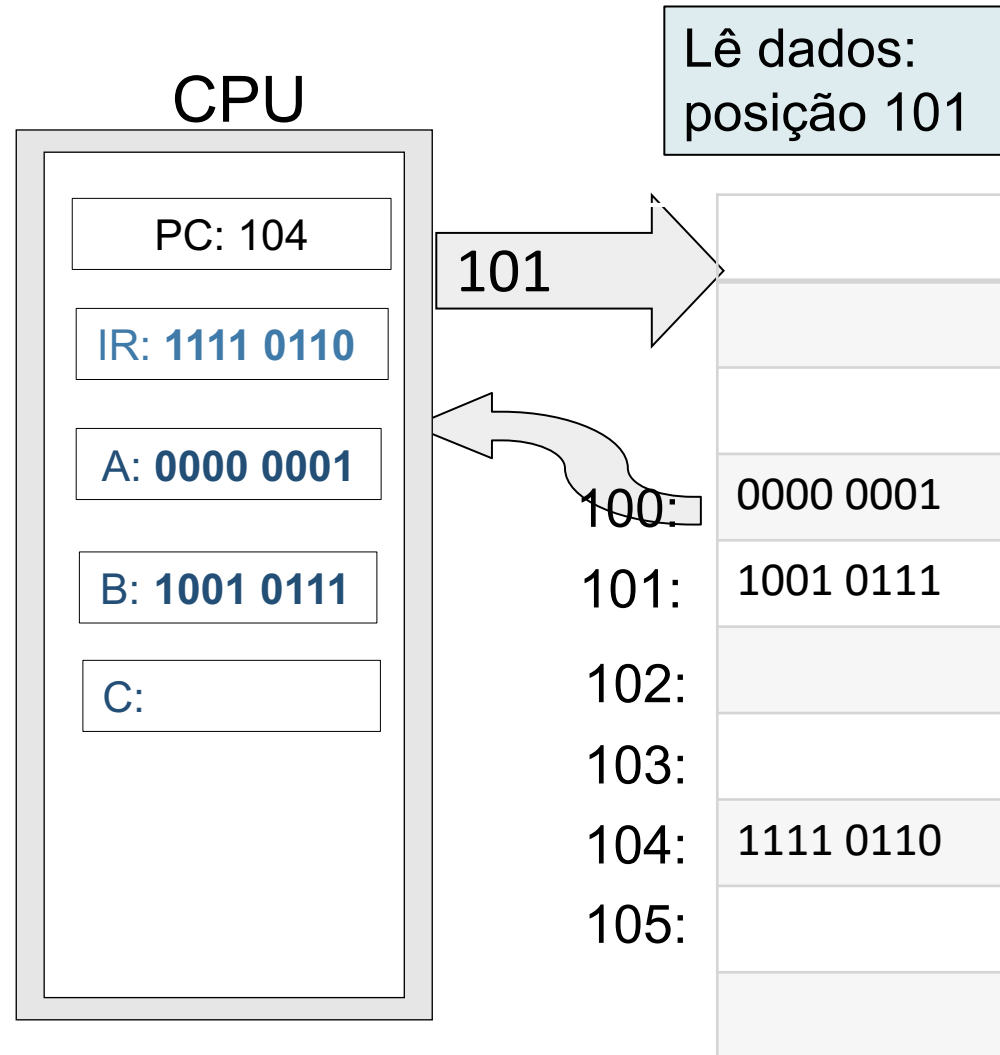
Funcionamento do CPU

- 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 104.



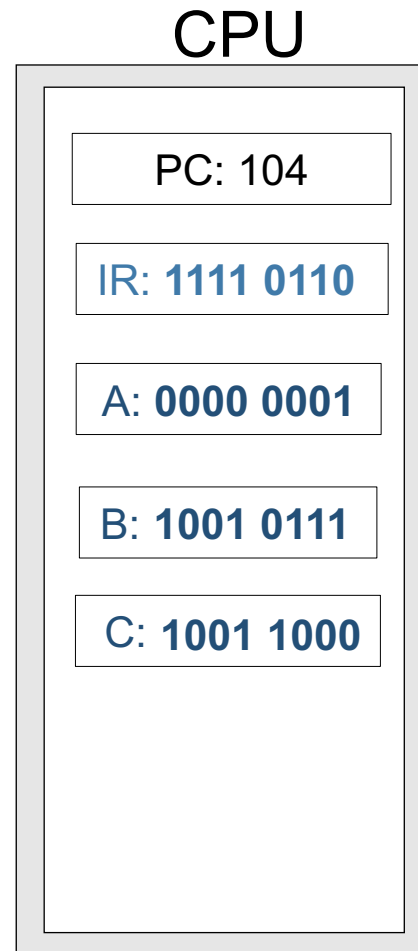
Funcionamento do CPU

- 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.



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.

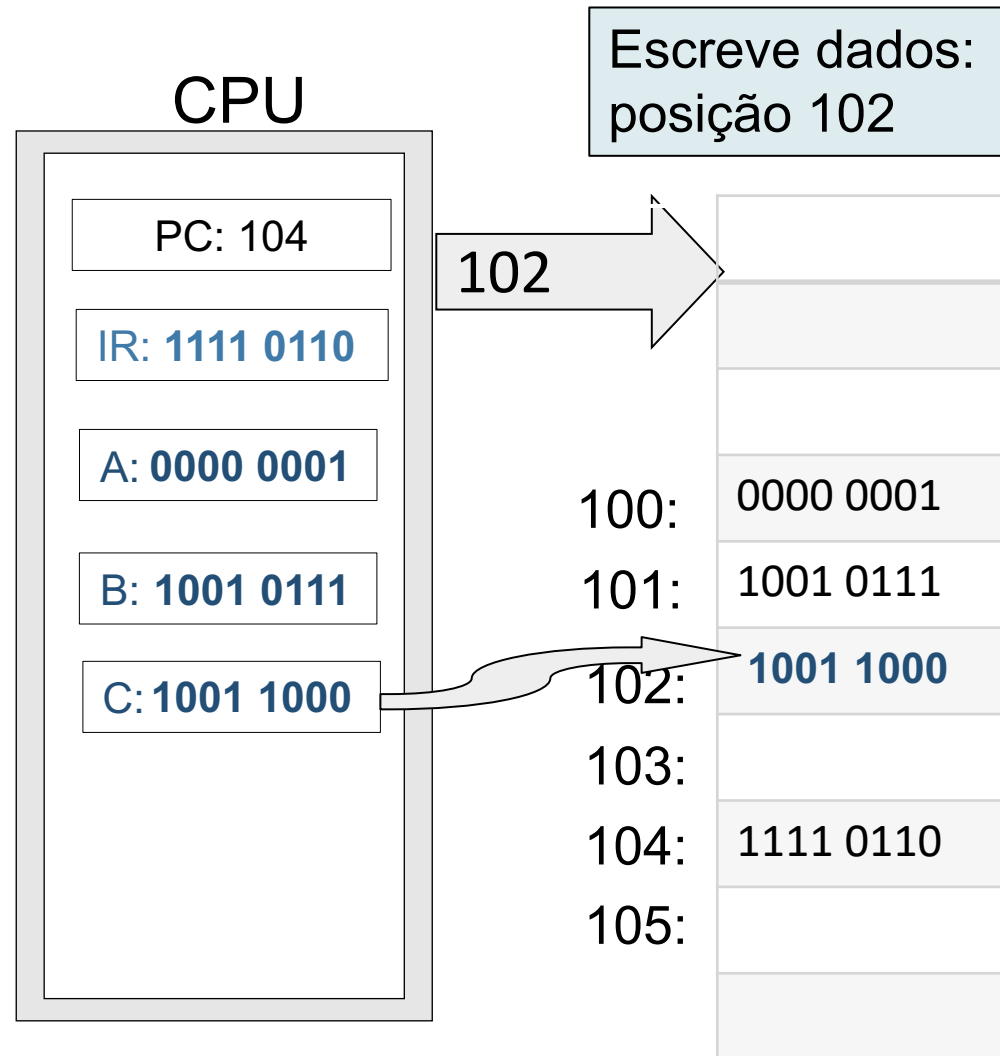


Computa a soma

100:	0000 0001
101:	1001 0111
102:	
103:	
104:	1111 0110
105:	

Funcionamento do CPU

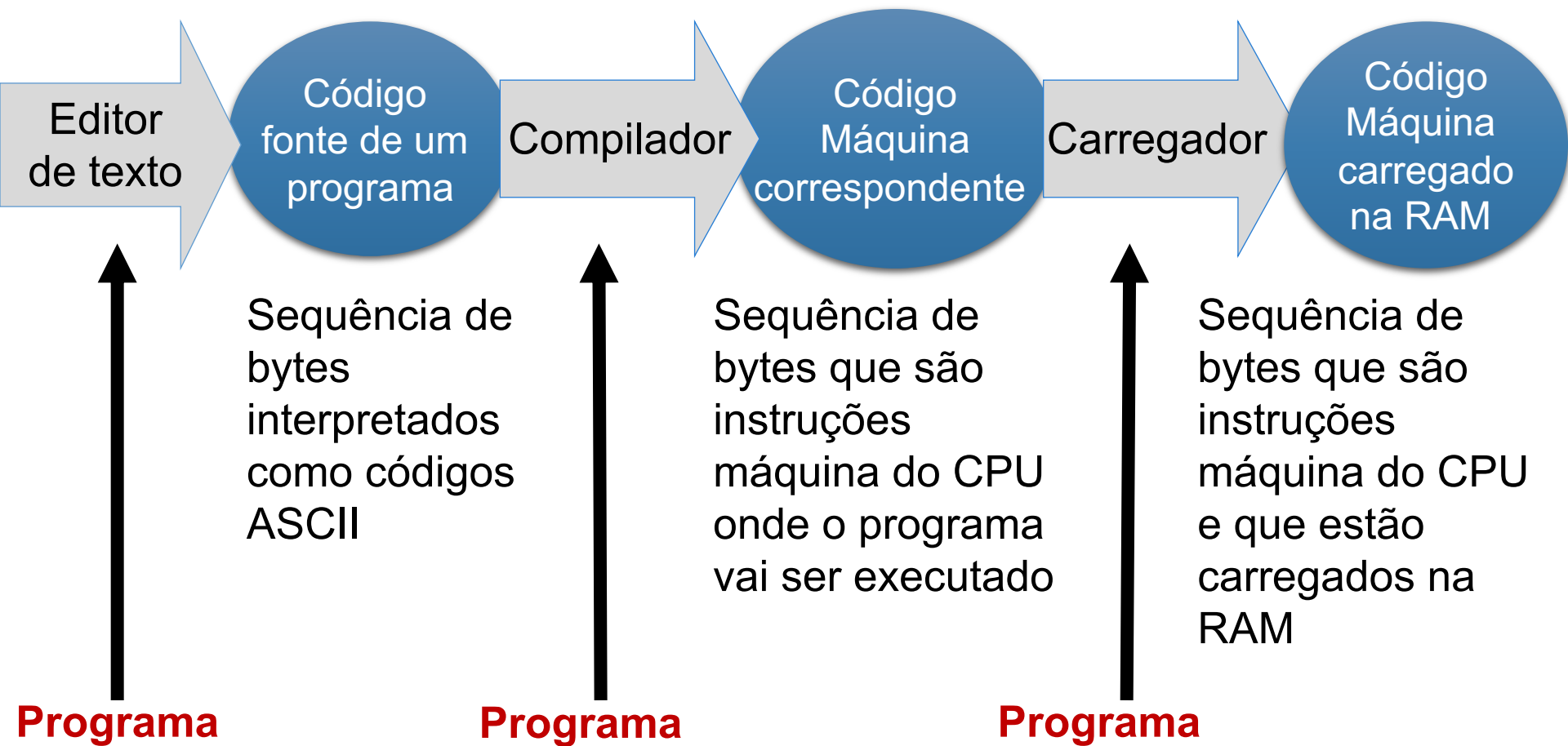
- A instrução define a ação elementar a executar
 - Ações atuam sobre dados armazenados em memória central ou num dispositivo de entrada/saída.
- Exemplo:
somar 100 101 102
 - Soma o conteúdo das posições 100 e 101 e armazena o resultado na posição 102.



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

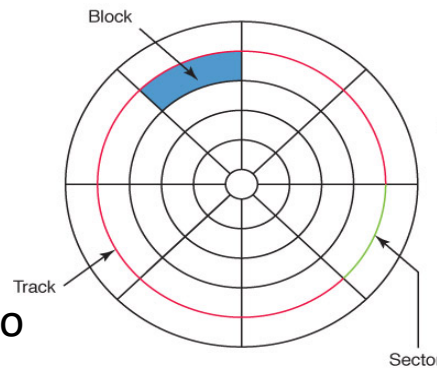
- Hardware
 - CPU + Memória
 - Dispositivos de entrada-saída
 - Interação: teclado, rato, ecrã
 - Arquivo: discos
- Software de sistema
 - Compiladores / interpretadores
 - Interpretador de comandos (incluindo carregador)
 - Sistema operativo (ou de operação)
 - Suporte das chamadas ao sistema feitas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

Ciclo de vida de um programa

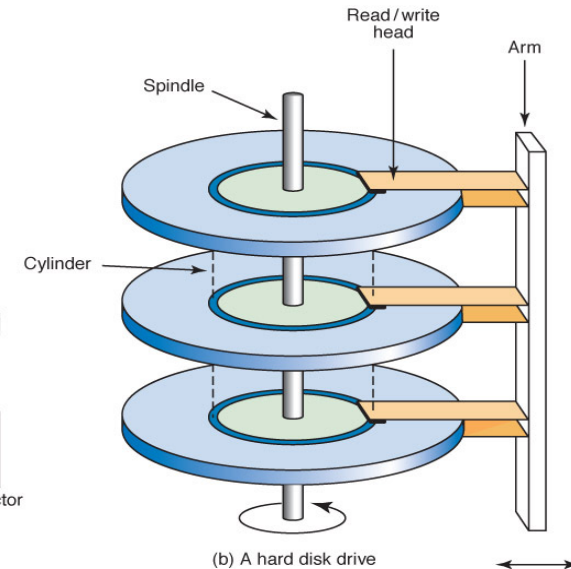


Onde guardar o código fonte e os programas?

- Num dispositivo de armazenamento permanente, por exemplo: o **disco**
- Os discos são complicados!
 - 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)



(a) A single disk



(b) A hard disk drive

- **Ficheiro:** abstração fornecida pelo sistema operativo que permite esconder a complexidade do dispositivos de armazenamento
 - Nome que é uma cadeia 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

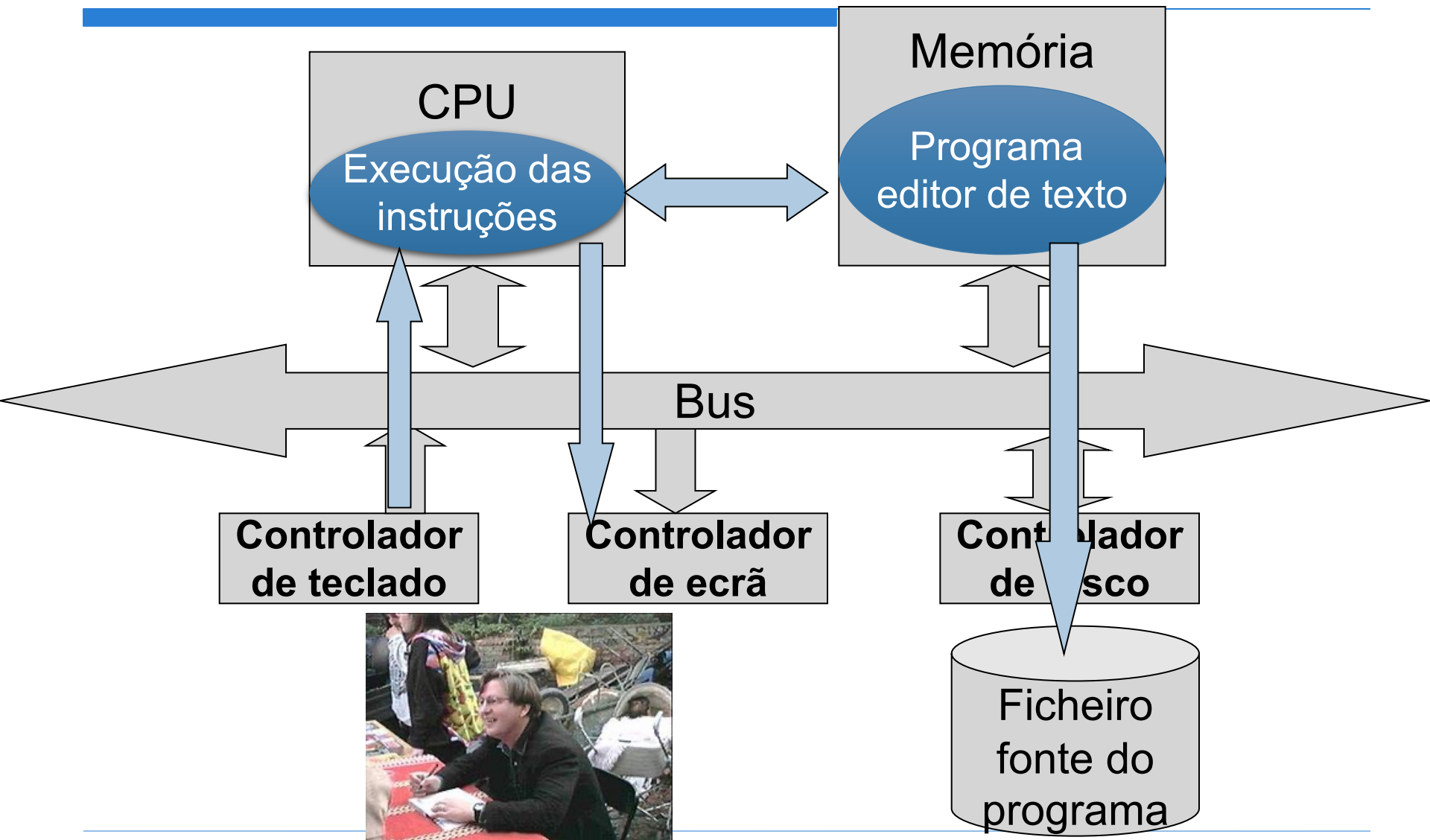
Ficheiro
executável

Carre-
gador

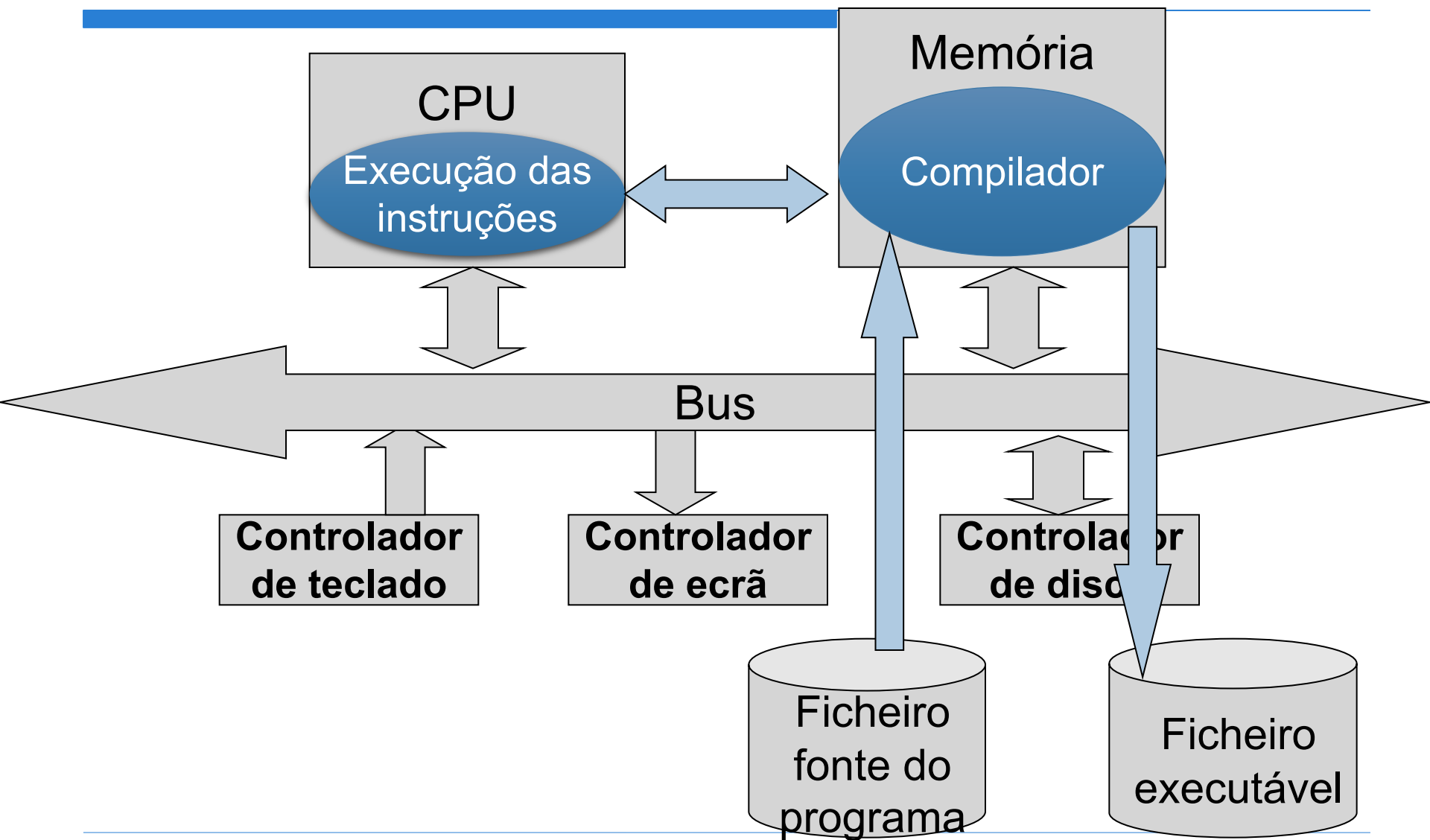
Código
carregado
na RAM



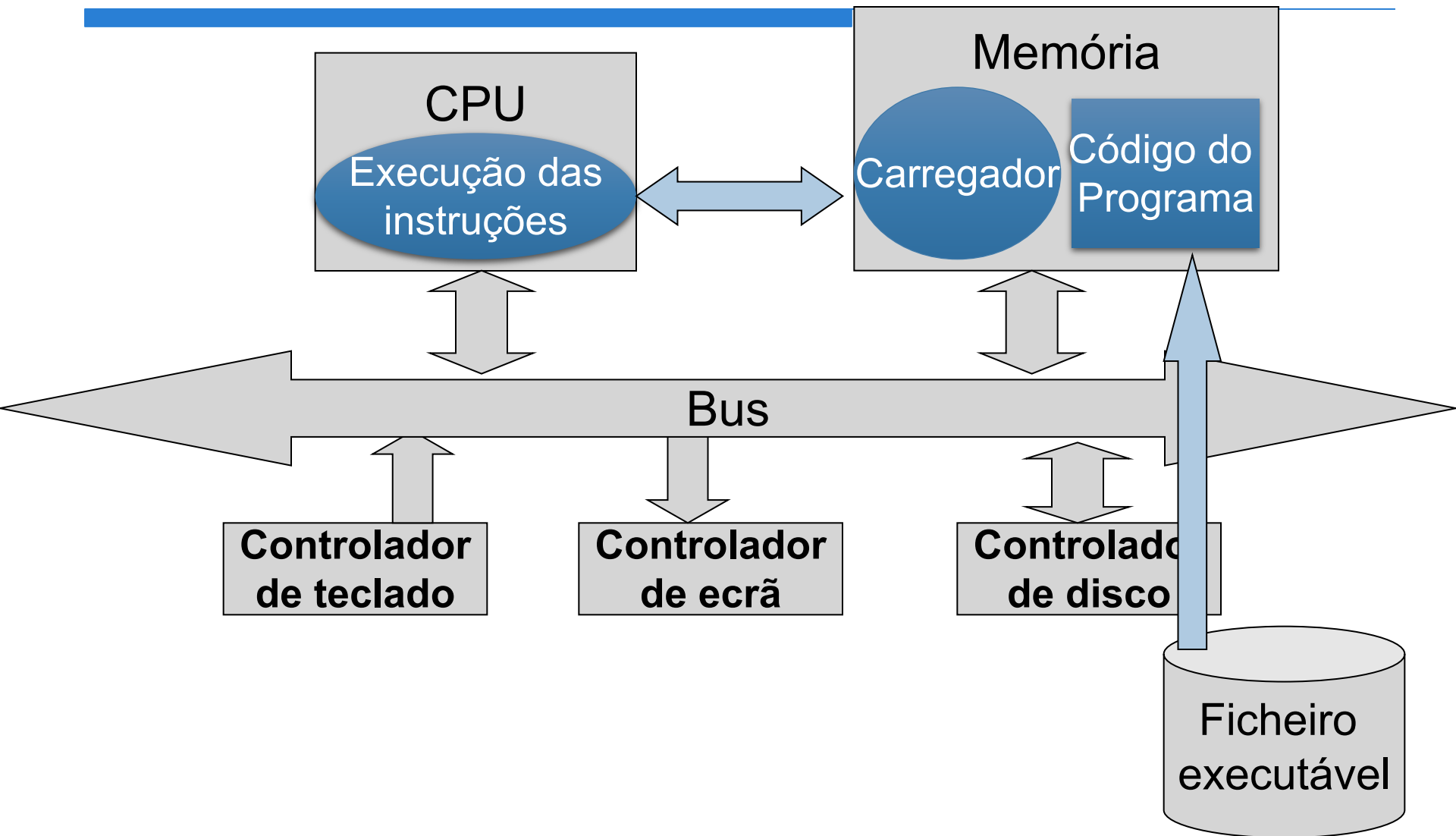
Edição do programa



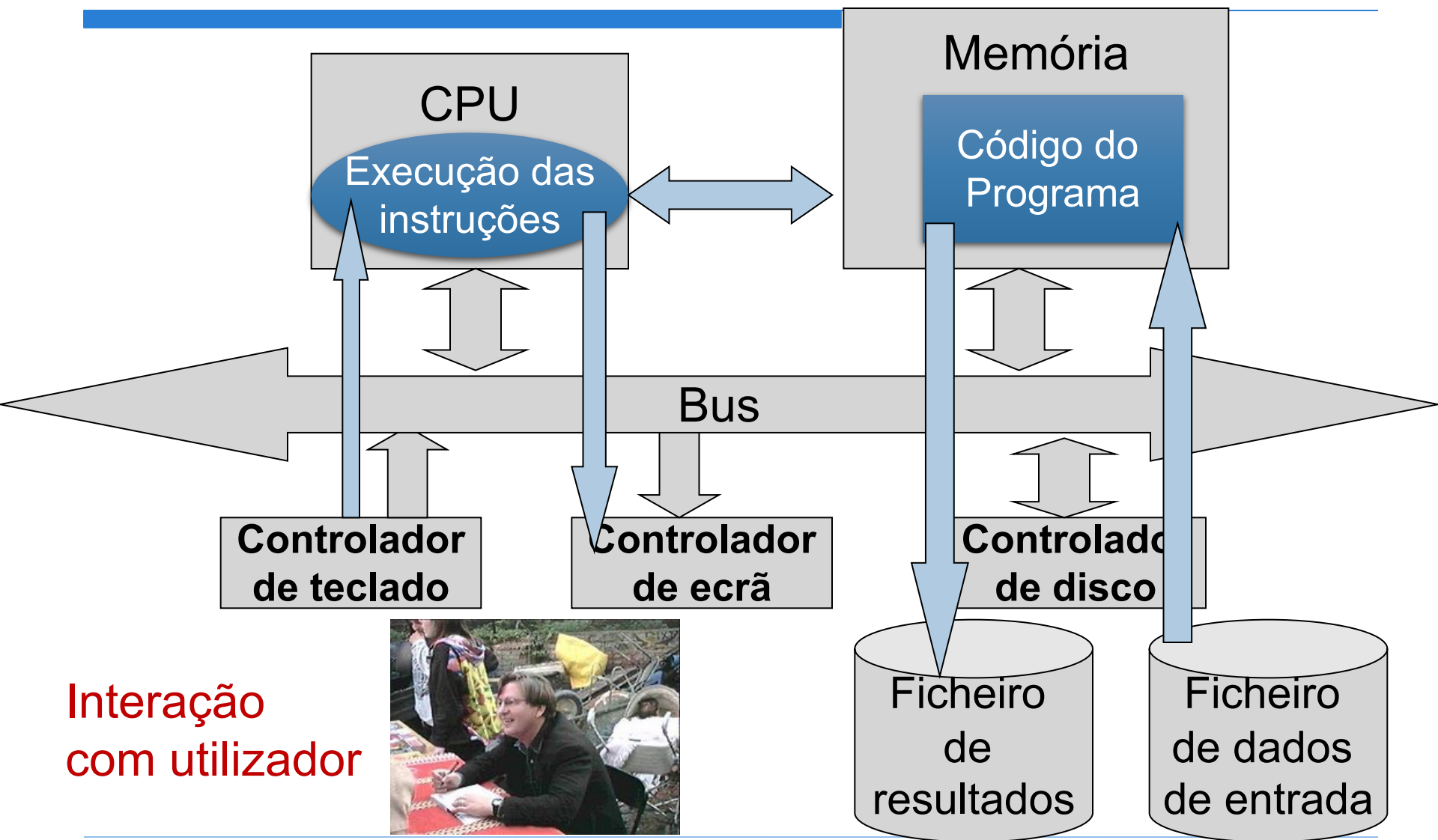
Geração do código



Carregamento



Execução

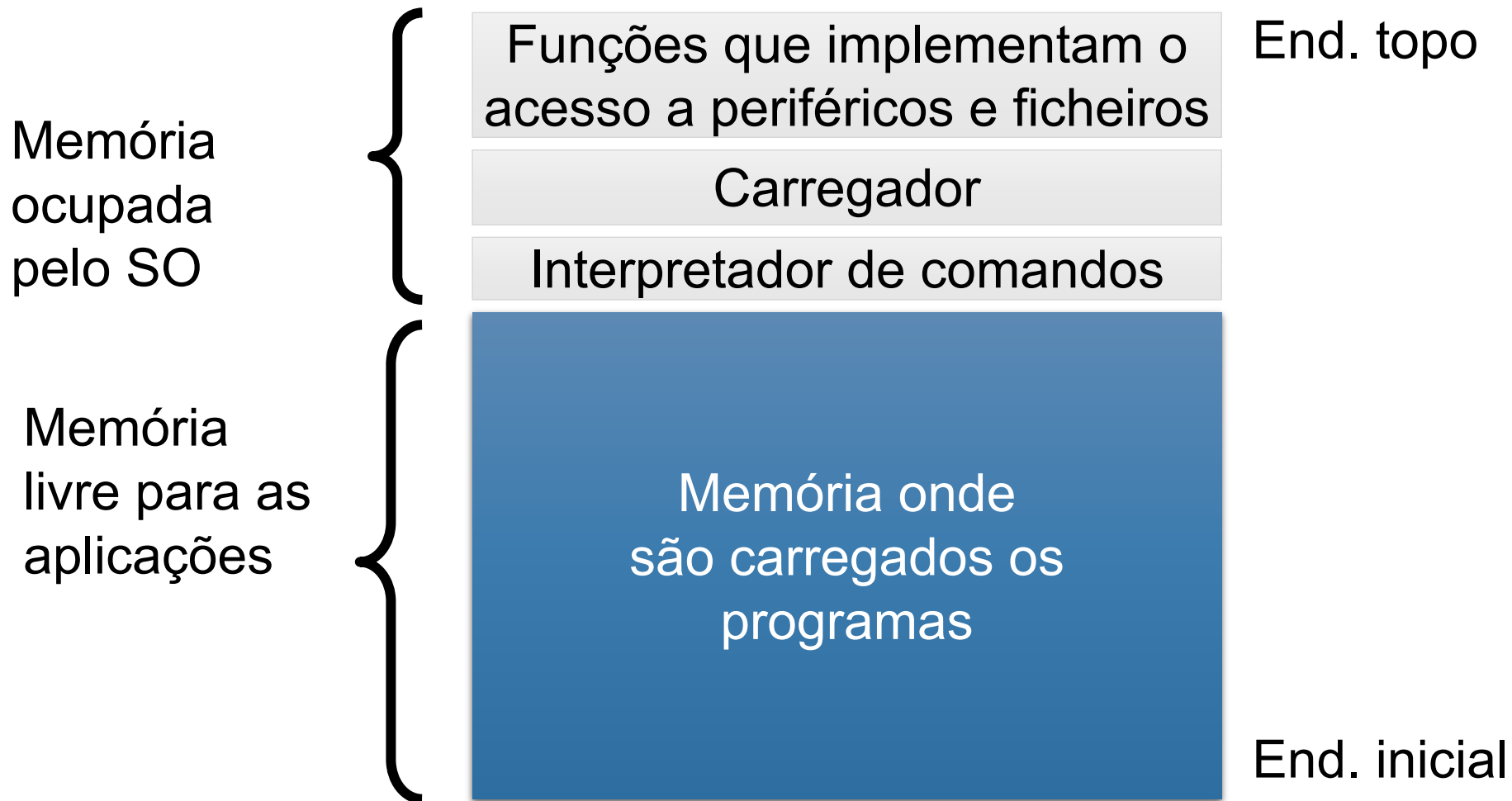


O que é que é preciso?

- 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

O Sistema de Operação (ou
Operativo) – abrev. SO

O SO na Memória

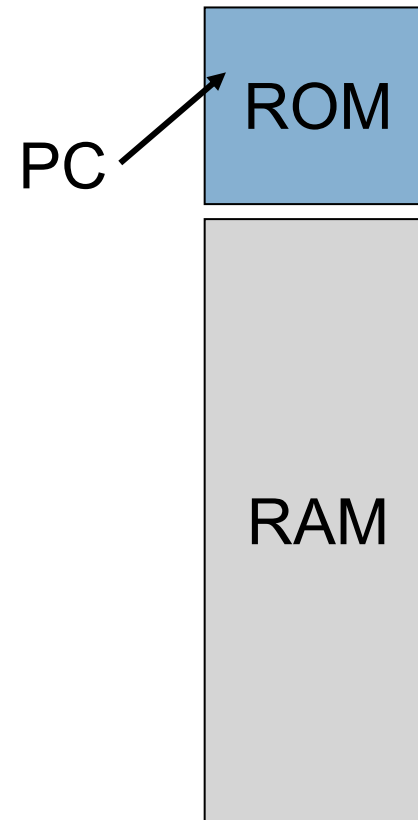


Funções de um SO

- Supervisionar a utilização dos recursos do sistema
- Controlar os periféricos
- Gestão da memória central
- Gestão de ficheiros
- Suporte da interacção com o utilizador

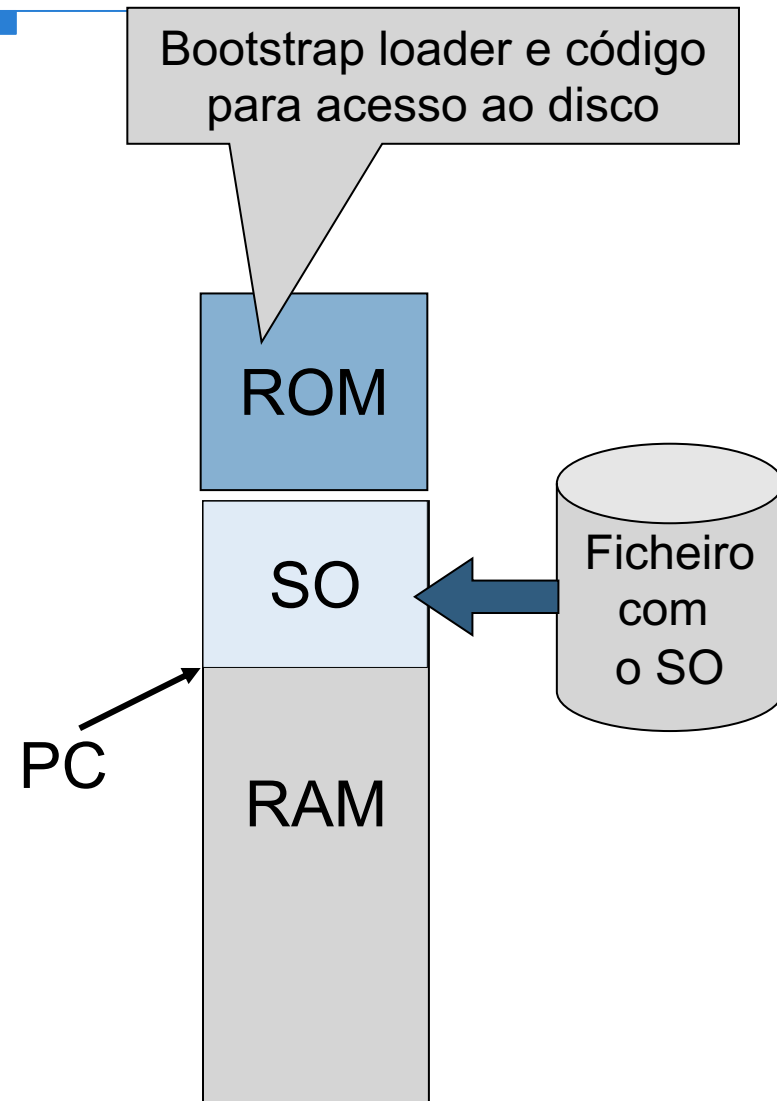
Arranque do SO

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



Arranque do SO

- É mais habitual 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 SO

- 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 interação do utilizador

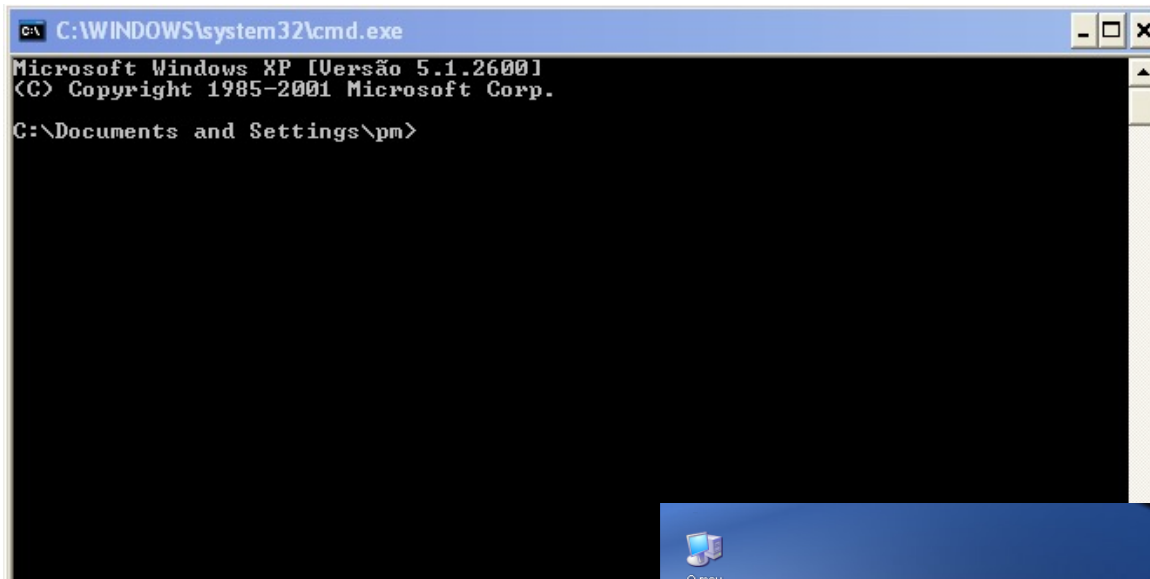
Interpretador de comandos

- Vários nomes e formas:
 - Orientado para linha de comando
 - Shell (UNIX), cmd.com (Windows)

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

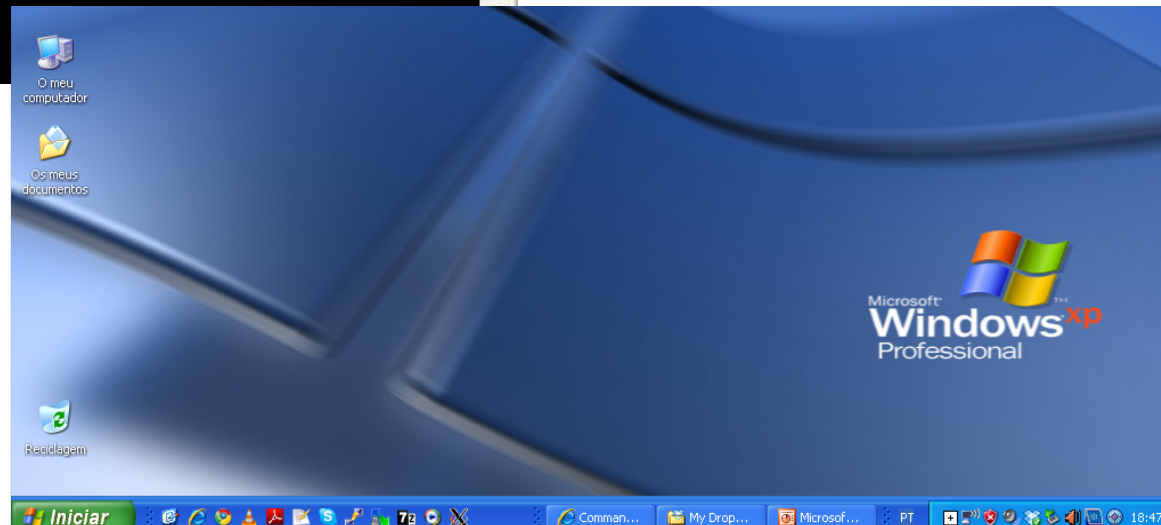
- Janelas, menus, rato ...
 - Windows, Macintosh, X-Windows (Linux/UNIX)

Interpretador de comandos



Linha de comando

Gráfico



Execução de programas

- Linha de comando
 - Escreve-se o nome do programa e pede-se ao SO para o executar
 - Podem-se passar argumentos aos programa (nomes de ficheiros ou outras informações)
- Interface Grafica (Graphical User Interface-GUI)
 - Duplo-click num ícone
 - Duplo-click num ícone que representa um ficheiro de dados. O programa associado com o ficheiro de dados é executado tendo o ficheiro de dados como argumento
- Modo não interactivo (Batch mode)
 - Sequência de comandos guardado num ficheiro de texto que são executados um a seguir ao outro sem intervenção do utilizador

Programa em execução

- O SO cria uma **máquina virtual** – ie, o S.O estende a máquina física com mais instruções e esconde a especificidade da máquina real onde as aplicações estão de facto a ser executadas

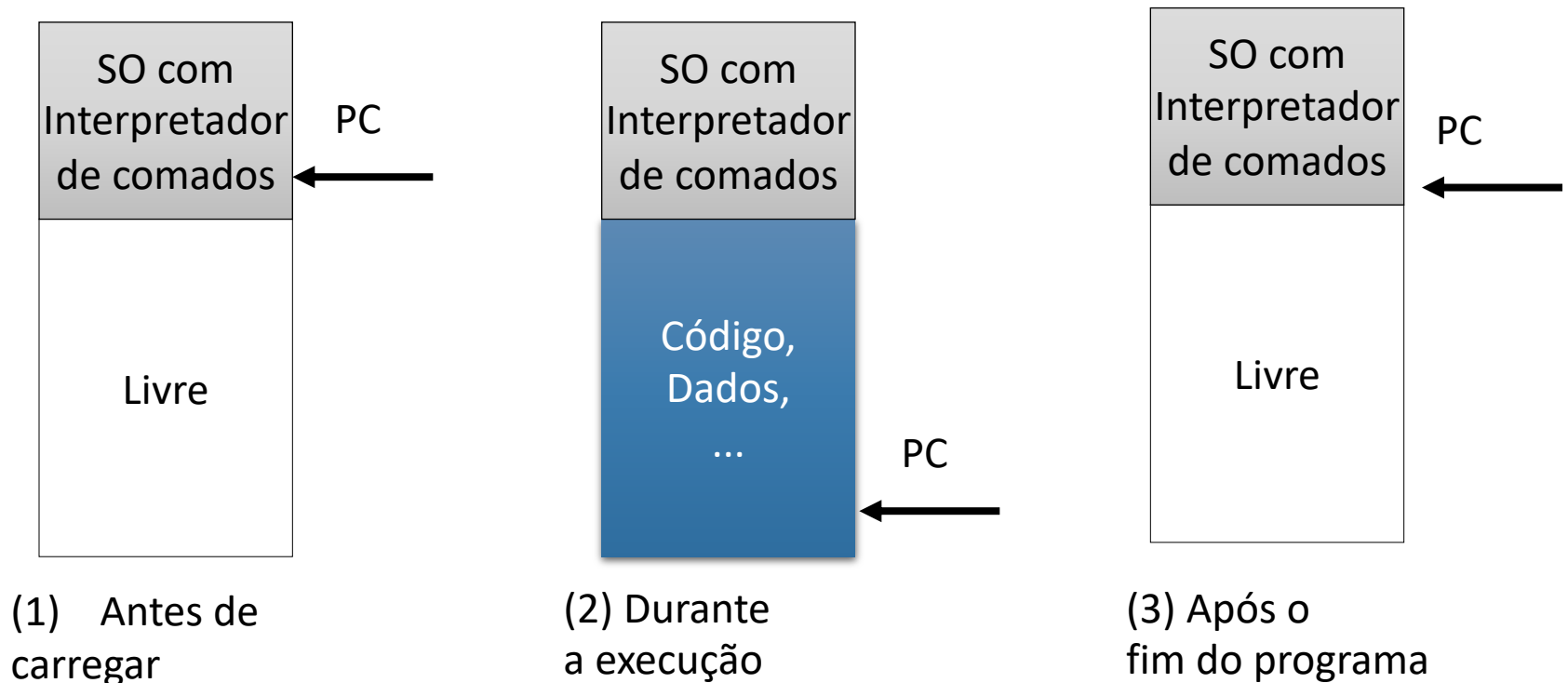
Instruções suportadas pelo sistema de operação, por exemplo as relacionadas com ficheiros



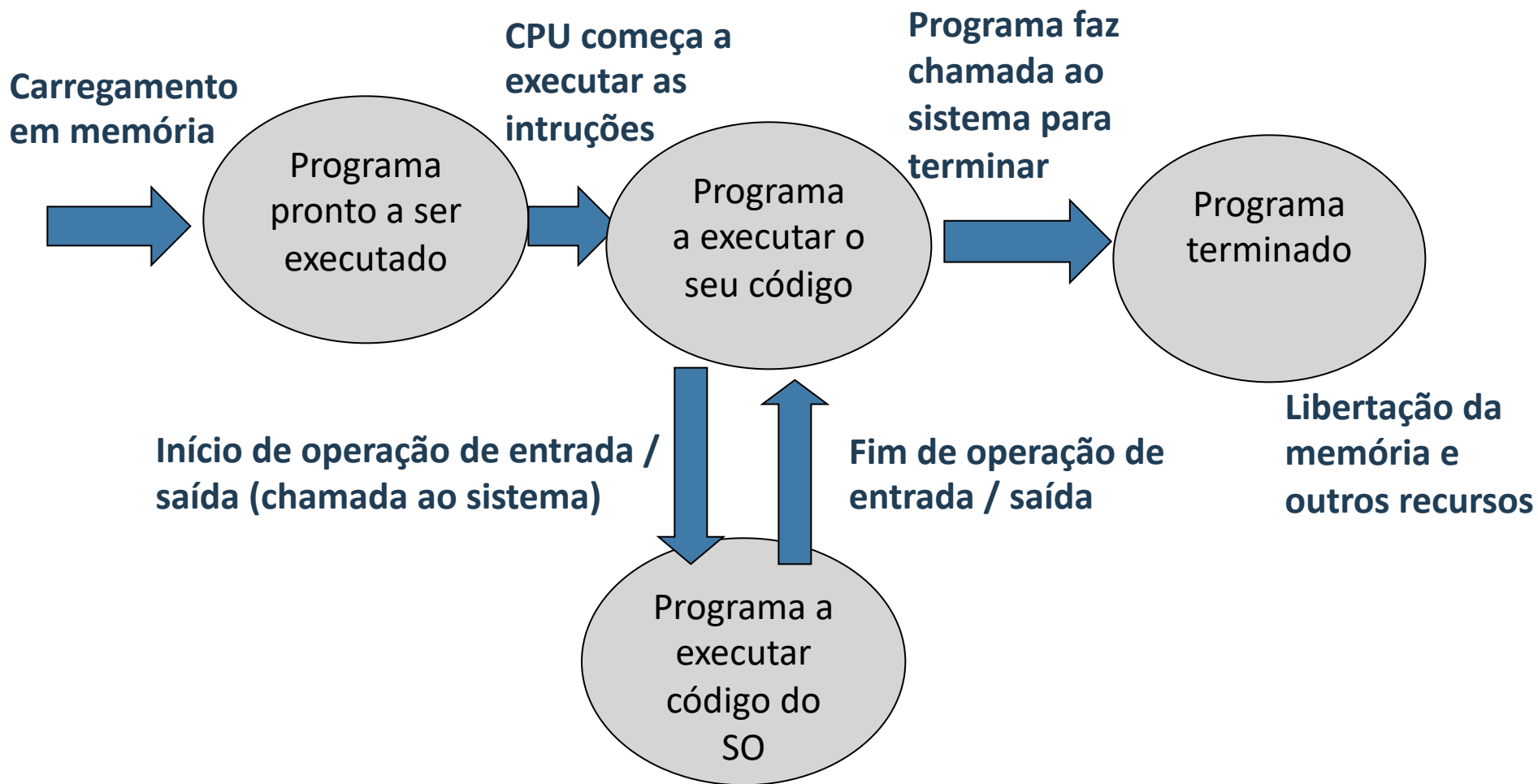
Instruções suportadas diretamente pelo hardware

Gestão da memória

- Gestão da memória da máquina virtual durante a execução de um programa



Programa em execução

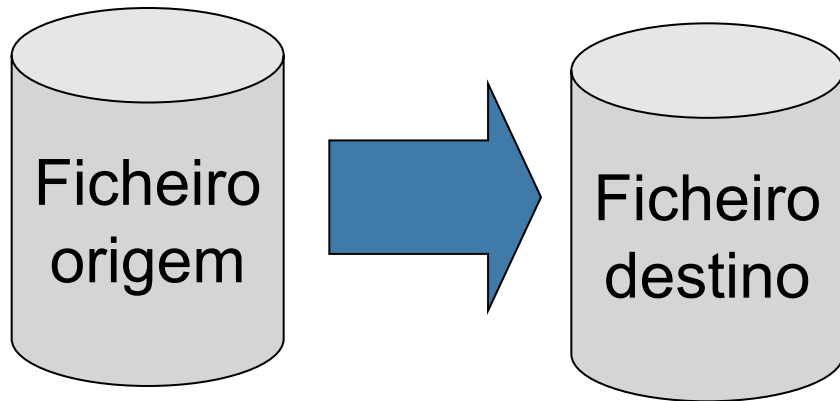


Comandos relacionados com ficheiros

- Tipos de comandos
 - Armazenamento (escrita)
 - Recuperação (leitura)
 - Organização (mudança de nome, ...)
 - Modificação de conteúdo
- Funcionalidades do sistema de gestão de ficheiros
 - Possibilidade de designar os ficheiros por um nome lógico abstraindo da posição e do periférico em que os dados estão fisicamente
 - Permitir o acesso aos dados contidos nos ficheiros sem preocupações com a localização
 - Estas funções devem estar acessíveis como chamadas ao sistema: permite que o interpretador de comandos e outros programas usem estas funcionalidades

Exemplo do uso de chamadas ao sistema

- Programa para copiar o conteúdo de um ficheiro para outro



Obter o nome do ficheiro origem

Escrever mensagem

Obter o nome

Se o nome não existir, terminar com erro

Obter o nome do ficheiro destino

Escrever mensagem

Obter o nome

Se o nome existir, terminar com erro

Abrir os ficheiros de origem e destino **do {**

ler N bytes do ficheiro origem

escrever N bytes no ficheiro destino

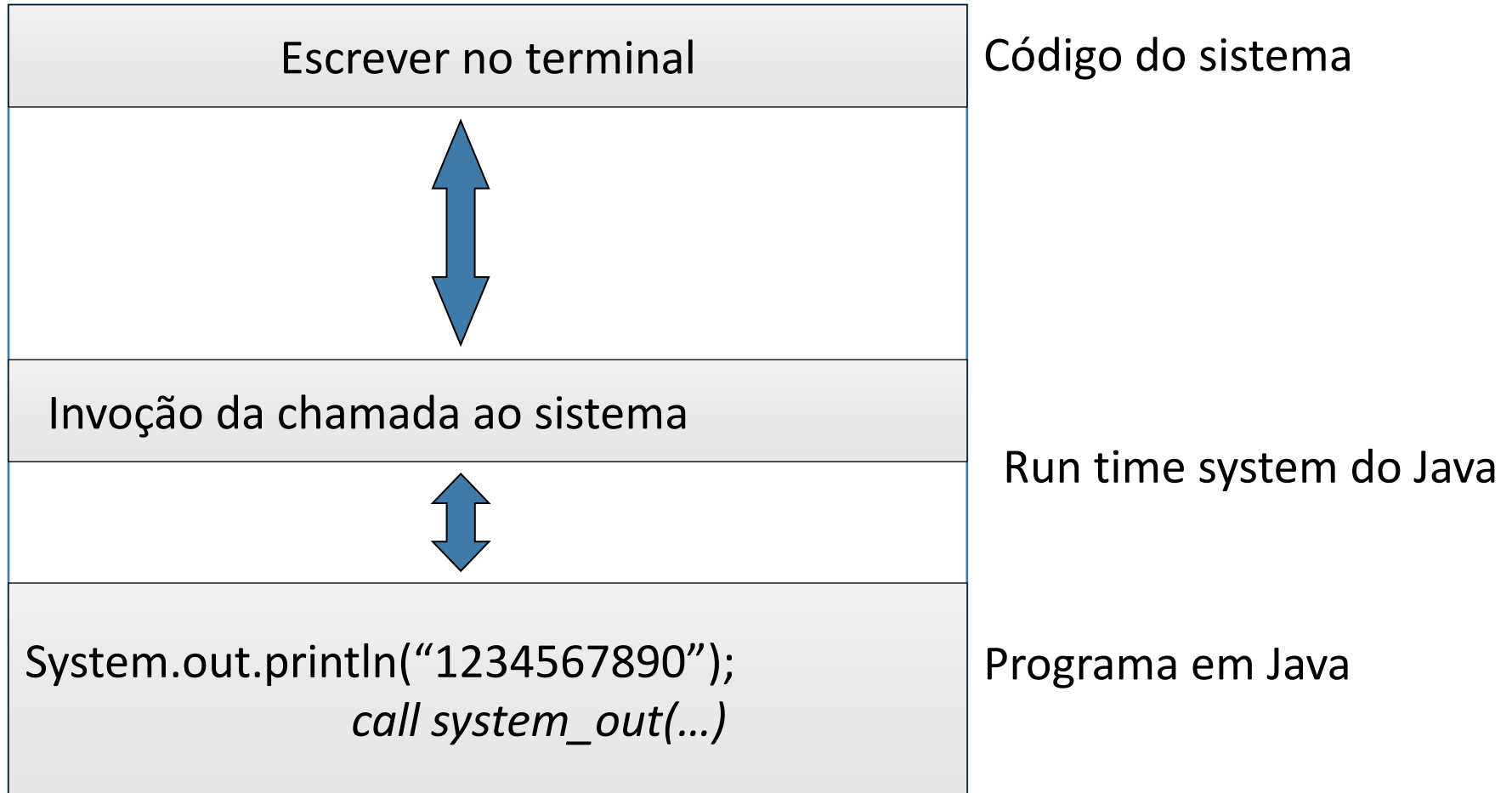
} while a leitura do ficheiro origem não dá erro

Escrever uma mensagem indicando o fim

Acesso às chamadas ao sistema

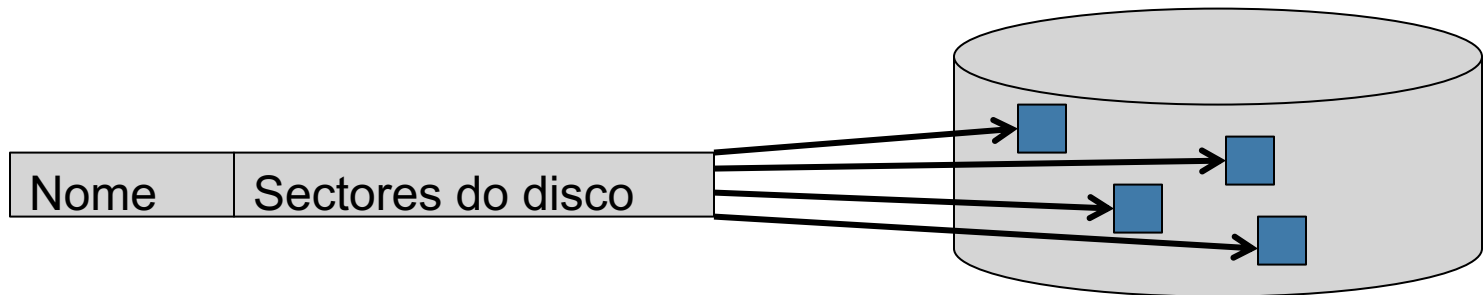
- Métodos ou funções dedicadas a isso.
- Exemplos:
 - `printf()`, ... no C
 - `System.out.println()` em Java
- Quando estas invocações aparecem no código, o compilador coloca no seu lugar a chamada a funções do chamado sistema de execução da linguagem (language runtime system)
- Estas funções é que invocam o sistema

Acesso às chamadas ao sistema



SO como gestor de recursos

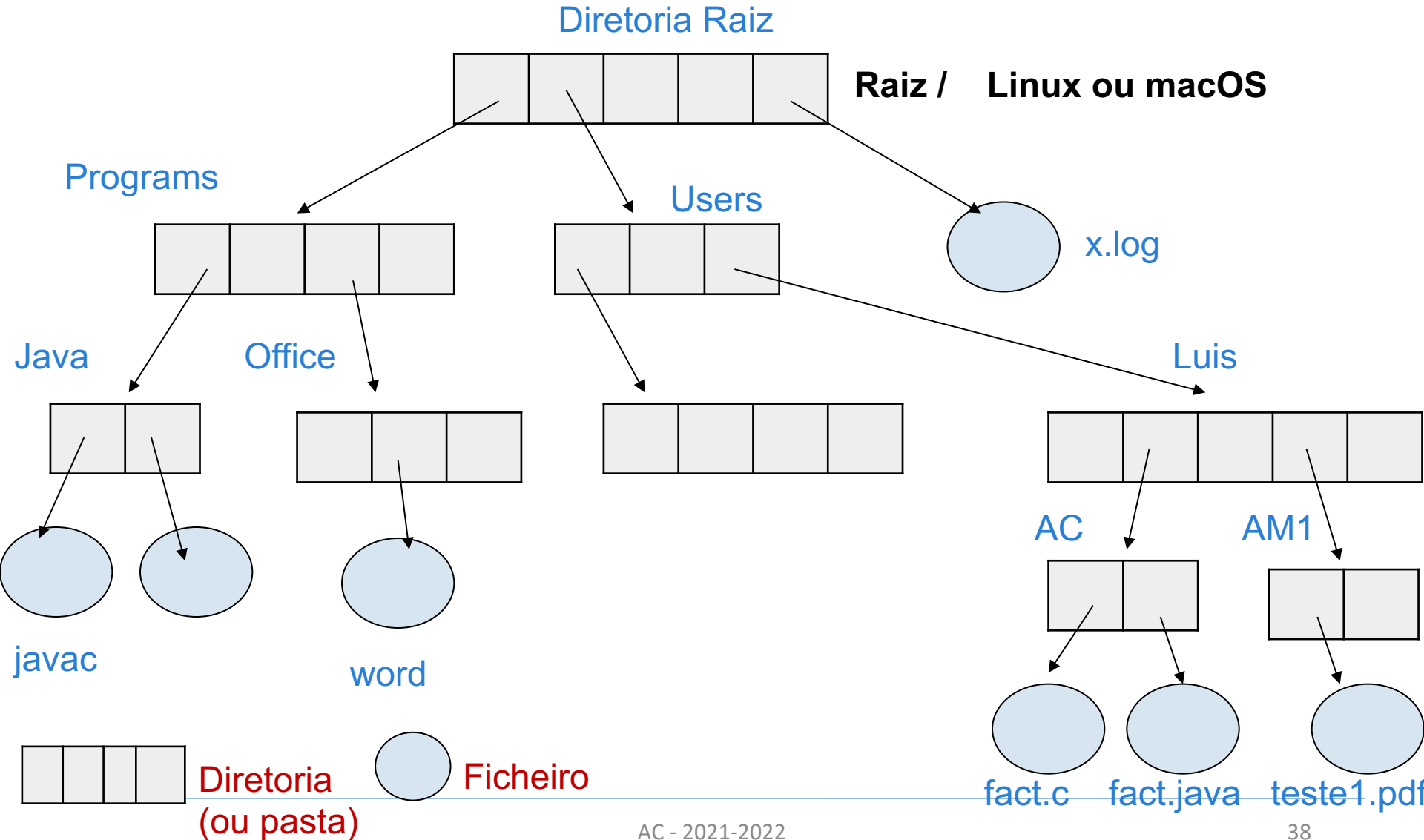
- Pode ser visto como um gestor de recursos – o SO é responsável por garantir uma exploração eficiente dos recursos; impede os programas de aplicação de fazer acesso directo aos recursos
 - Exemplo, o sistema gere o disco onde guarda os ficheiros
 - Tabela que, para cada ficheiro no disco lhe associa os sectores do disco onde está guardado



Funções do SO

- O SO é o componente do software de sistema que serve de suporte a todo software de aplicação e de sistema
- Os SOs controlam a execução de programas, a **gestão** de recursos, **protecção** e **segurança**
- O SO cria uma **máquina virtual** com instruções de alto nível que os programas podem usar fazendo chamadas ao sistema
- O SO **impede o acesso directo** aos periféricos pelos programas

Sistema de ficheiros



Sistema de ficheiros

Diagram illustrating the file system structure and a Windows File Explorer window.

File Explorer Window: The window shows the "Shawn Brink" user profile. The left sidebar lists the "Quick access" section, including Desktop, OneDrive, and the user's profile. The main pane displays the "Shawn Brink" folder contents, including 3D Objects, Contacts, Desktop, Documents, Downloads, Favorites, Links, Music, OneDrive, Pictures, Saved Games, Searches, and Videos. The "Pictures" folder is selected.

Annotations and Diagrams:

- Programs:** A diagram shows a box labeled "Programs" with arrows pointing to "Java" and "Office".
- Java:** A diagram shows a box labeled "Java" with arrows pointing to two circles labeled "javac".
- AM1:** A diagram shows a box labeled "AM1" with an arrow pointing to a circle labeled "teste1.pdf".
- Files:** At the bottom, the files "fact.c", "fact.java", and "teste1.pdf" are listed.

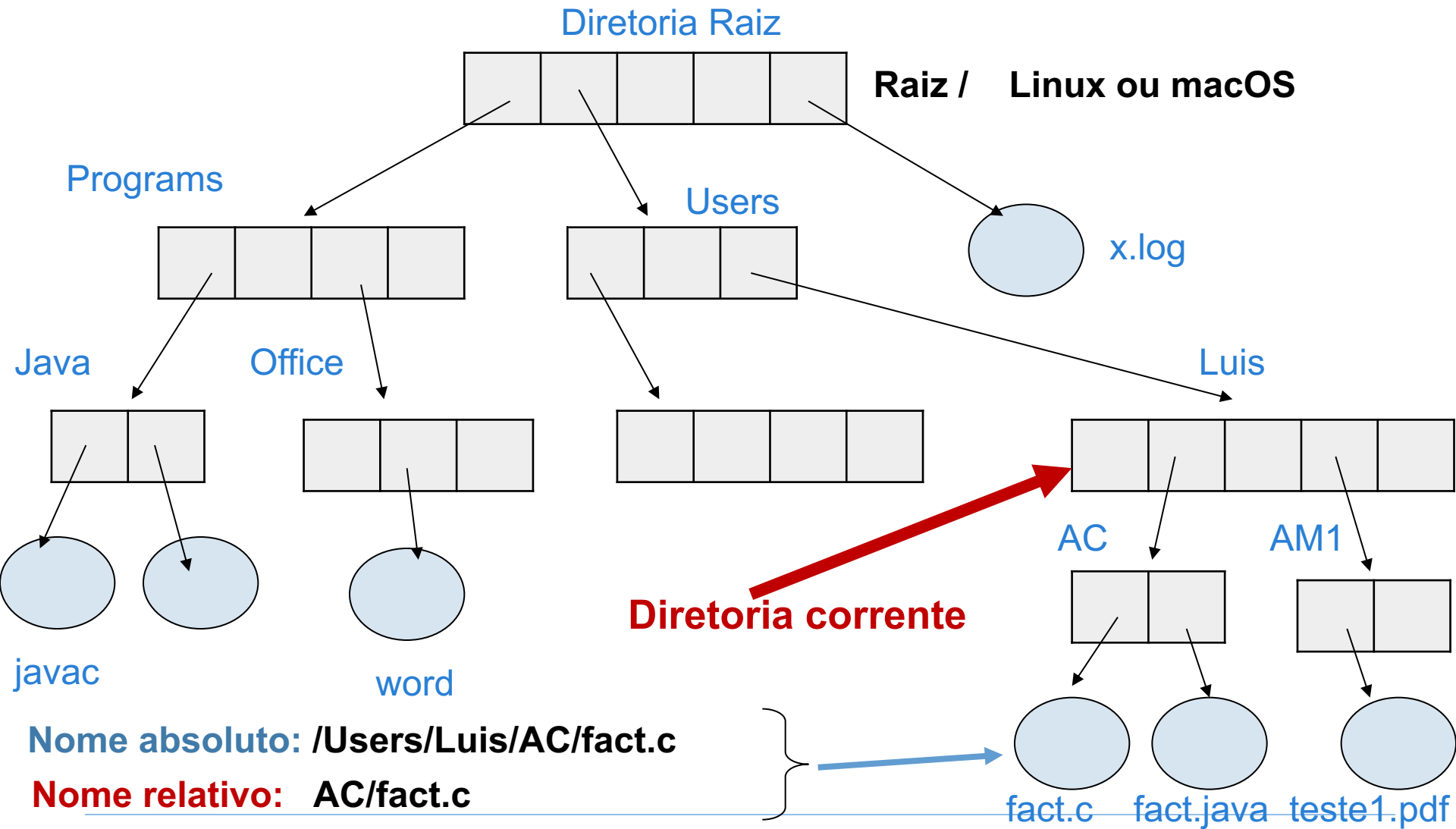
Sistema de ficheiros

- O sistema de ficheiros está organizado hierarquicamente em forma de **árvore** (“invertida”).
 - **Raiz** (a diretoria do topo, da qual “descendem” as outras diretorias e os outros ficheiros).
 - C:\ no Windows
 - **Ramos** (as outras diretorias).
 - **Folhas** (ficheiros).
- Cada diretoria (ou diretório, ou pasta) pode conter ficheiros e outras (sub-)diretorias.
- Nesta árvore, a pesquisa de um ficheiro é eficiente.

Sistema de ficheiros

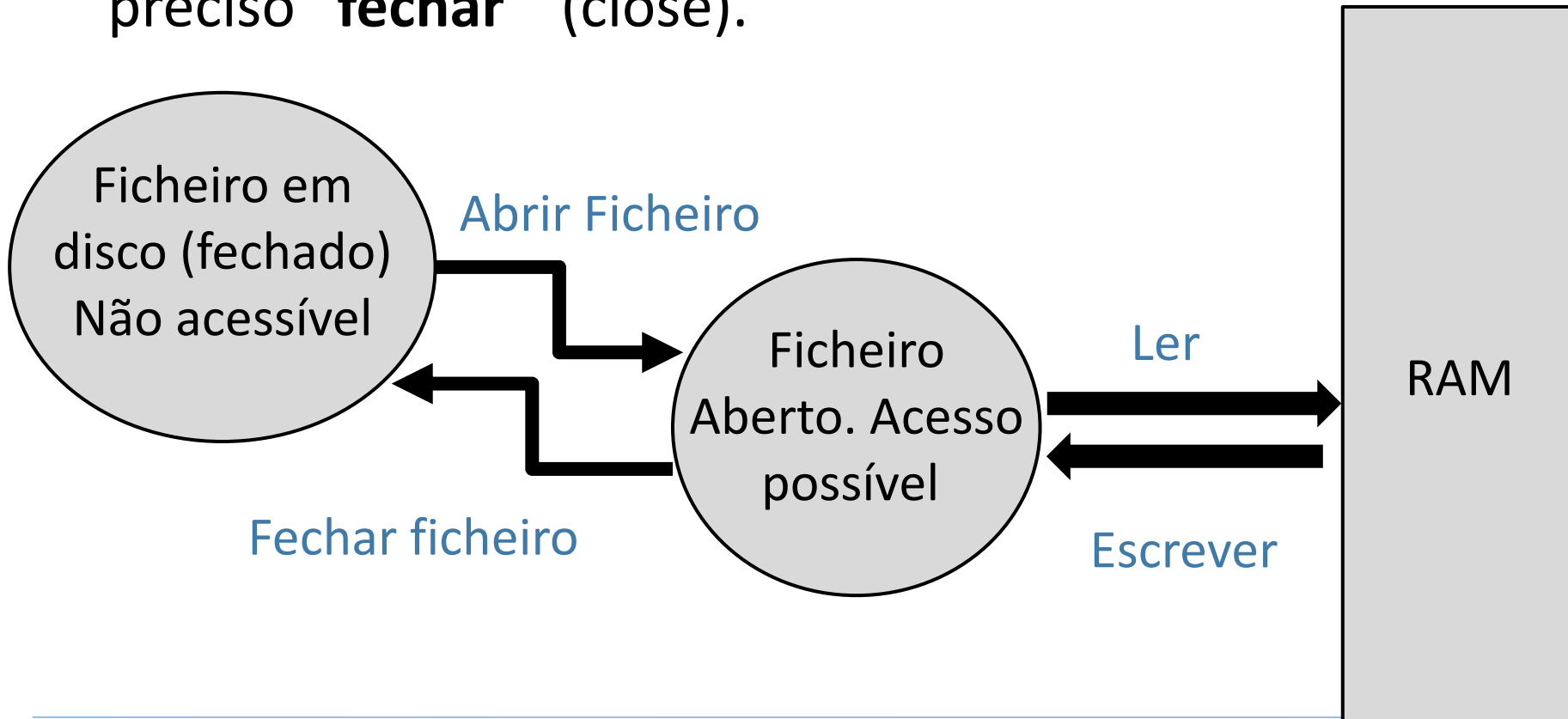
- O nome completo ou caminho para o ficheiro (pathname) pode ser:
 - **absoluto** – o caminho completo a partir da diretoria raiz;
 - **relativo** – o caminho a partir da **diretoria corrente**.
- Nos caminhos, os nomes de subdiretorias e ficheiros são separados por um carácter especial
 - \ em Windows
 - / em macOS e Linux
- Notações especiais (que podem ser usadas nos caminhos):
 - “.” representa a diretoria corrente;
 - “..” representa a diretoria pai da diretoria corrente.

Sistema de ficheiros



Sistema de ficheiros

- Para transferir dados de/para um ficheiro, é preciso “**abrir**” (open) o ficheiro. Após ler / escrever, é preciso “**fechar**” (close).



Ficheiros – Posição de Leitura /Escrita

- À medida que se lê ou se escreve, a posição corrente avança.
 - Não se lê o que já foi lido;
 - escreve-se a seguir ao que já foi escrito.

