



FSO aula teórica nº 02

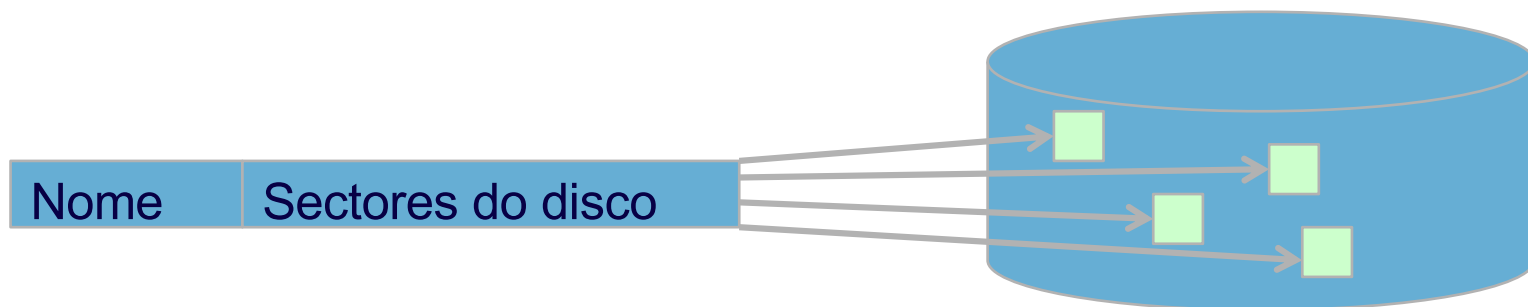
12/09/2018

Suporte de processos. Ficheiros

Bibliografia: livro Operating System: Three Easy Pieces (OSTEP) Secções 2.1, 2.4

Papel do Sistema de Operação

- Pode ser visto como *um gestor de recursos* – o S.O. 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



EXECUTAR PROGRAMAS, COMO ?

É preciso oferecer a cada programa uma máquina com

- *CPU*: para executar instruções
- *RAM*: para guardar código dados e pilha
- *Canais de entrada e saída*: Formas de comunicar com o exterior

EXECUTAR PROGRAMAS, COMO ?

- **Virtualização:** O sistema operativo cria as **máquinas virtuais** para cada programa executar a partir dos recursos físicos
 - **CPUs**
 - **RAM**
 - **Dispositivos de entrada saída** (discutido mais tarde)
- **Partilha de recursos:** Os vários programas em execução partilham os recursos, sob a supervisão do S.O.

Duas Abstracções Suportadas pelo SO

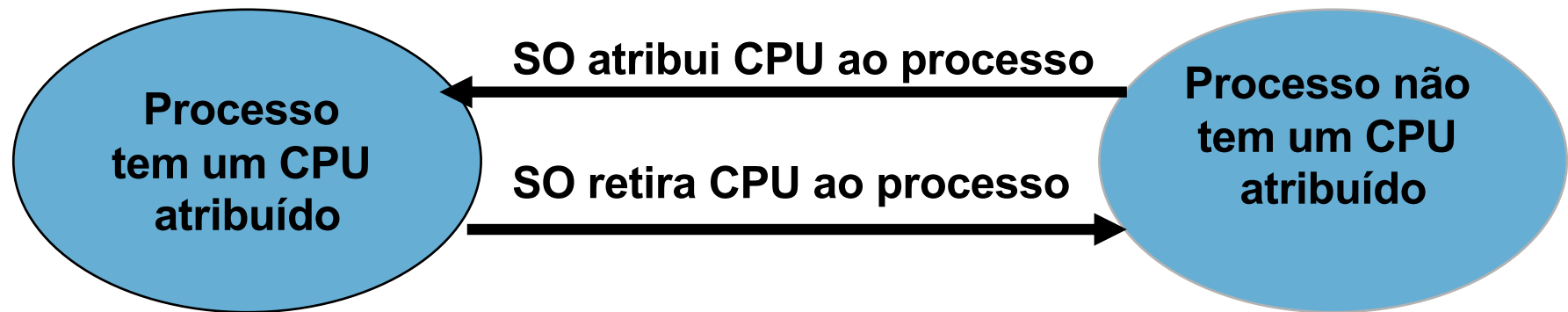
- **Ficheiro**
 - Disponibilizado ao programador como forma de esconder os detalhes de manipulação da memória persistente (discos)
- **Processo**
 - Disponibilizado ao programador como forma de esconder os detalhes da gestão de recursos necessários à execução de programas

Processo

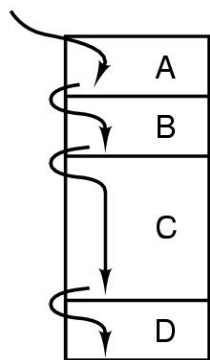
- **SO suporta para cada processo**
 - Um **CPU virtual** suportado na partilha do tempo do CPU (ou CPUs) reais
 - Um **conjunto de memória** (imagem do processo) obtida por partilha espacial da memória
 - Um conjunto de canais de entrada-saída que permitem enviar e receber bytes
 - De/Para Periféricos
 - De/Para Ficheiros
 - De/Para Outros processos

VIRTUALIZAÇÃO DO(s) CPU(s)

A repartição do CPU é feita temporalmente, sendo atribuídos a cada processo um CPU físico durante alguns milisegundos consecutivos

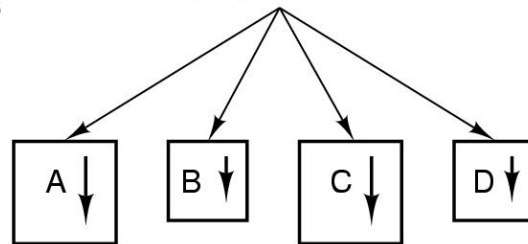


One program counter

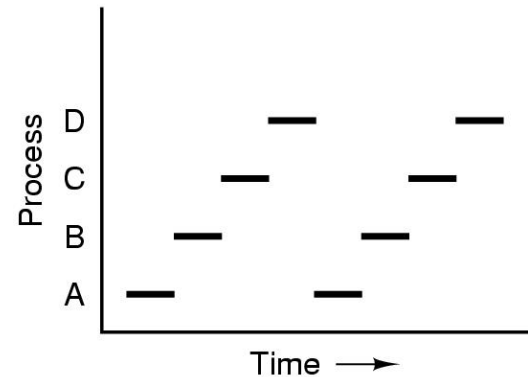


(a)

Four program counters



(b)

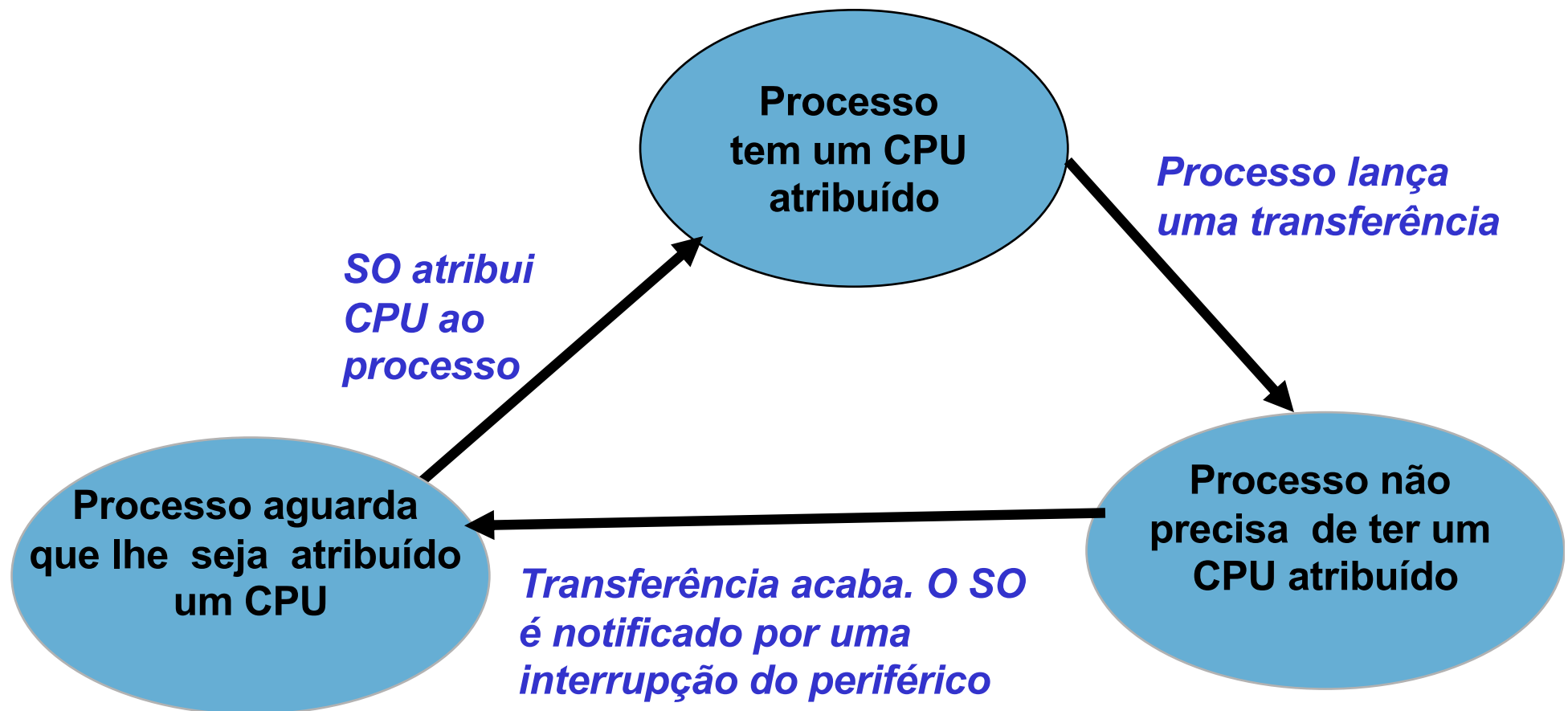


(c)

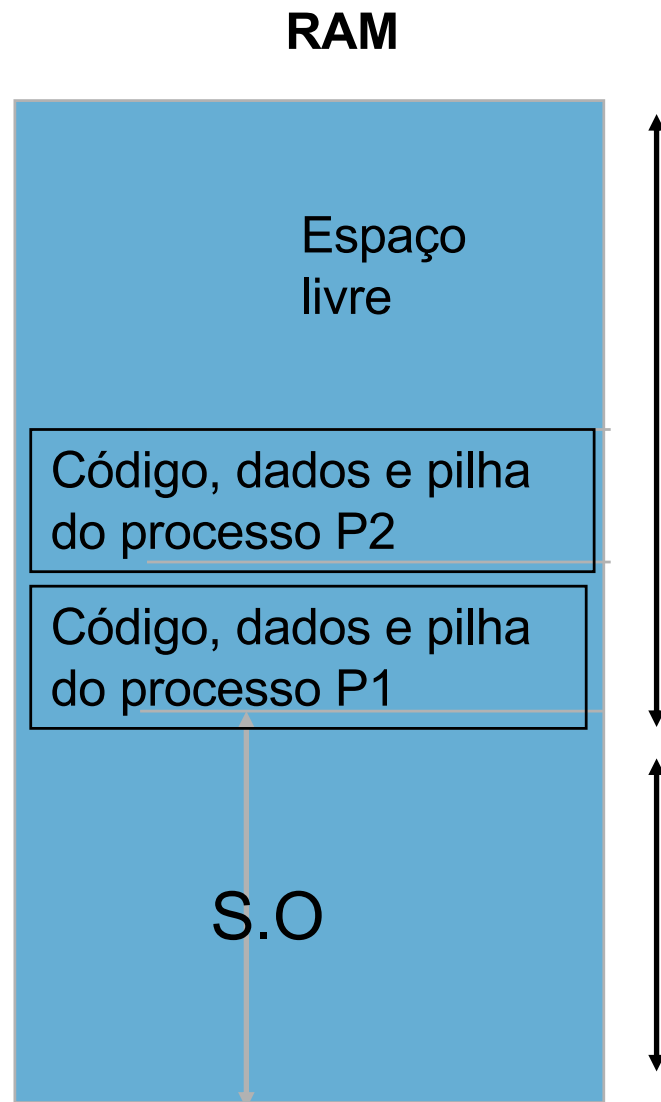
Tanenbaum,
Modern Operating
Systems

VIRTUALIZAÇÃO DO(s) CPU(s)

A repartição do CPU é feita tirando partido do facto de os periféricos serem muito lentos e os processos passarem muito à espera que transferências de informação acabem



VIRTUALIZAÇÃO DA RAM

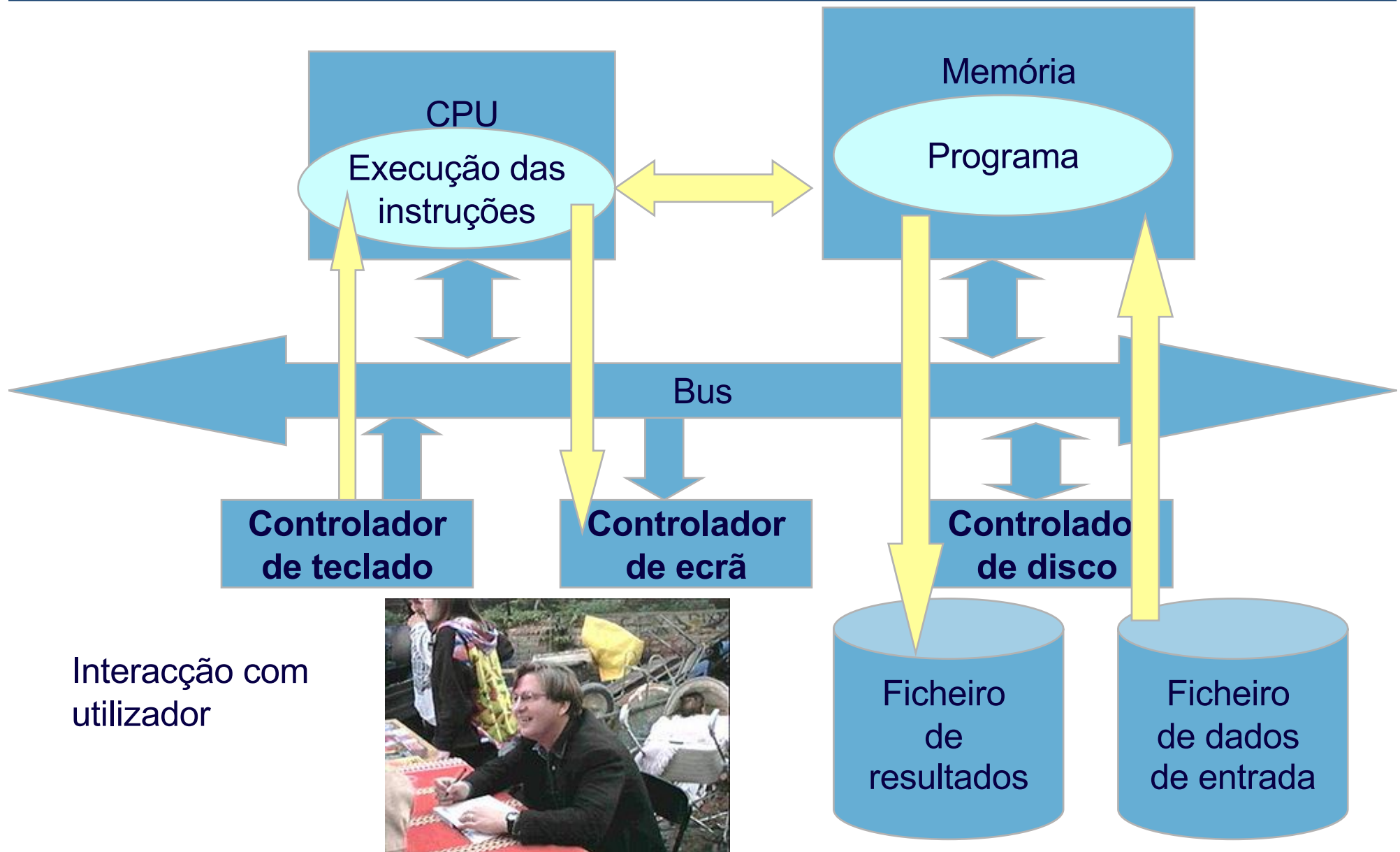


A repartição da RAM é feita espacialmente, sendo atribuídos aos processos faixas de endereços contíguos

RAM disponível para os processos. Quando um processo é lançado ocupa memória; quando termina a memória é libertada

RAM ocupada pelo SO

Execução de um programa



O que é que é preciso?

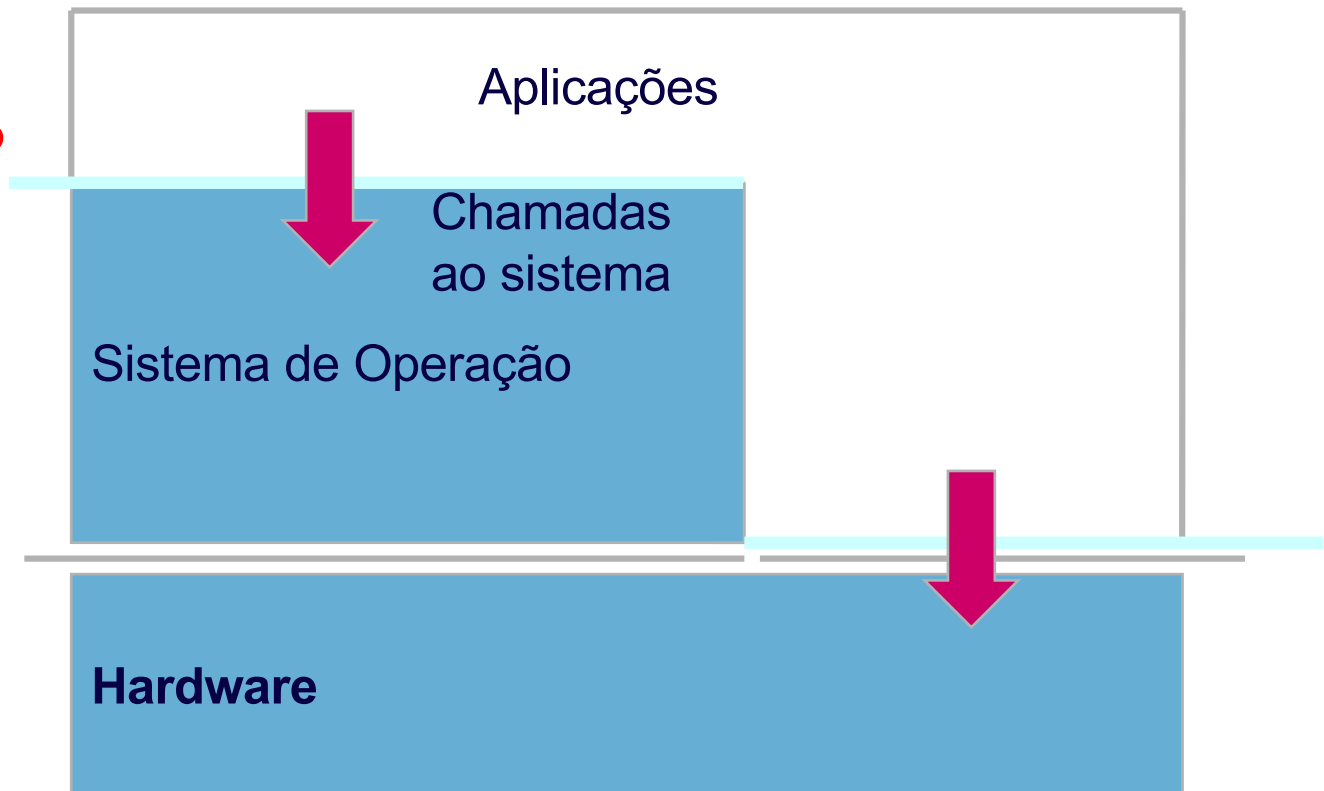
- Conjunto de código que implemente **funções de acesso aos periféricos e aos ficheiros**
 - Operações simples
 - Independentes do hardware
- **Carregador**
 - Usando as funções anteriores carrega ficheiros executáveis em memória
- **Interpretador de comandos**
 - Usando as funções anteriores lê comandos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

Programa em execução

- O S.O. *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

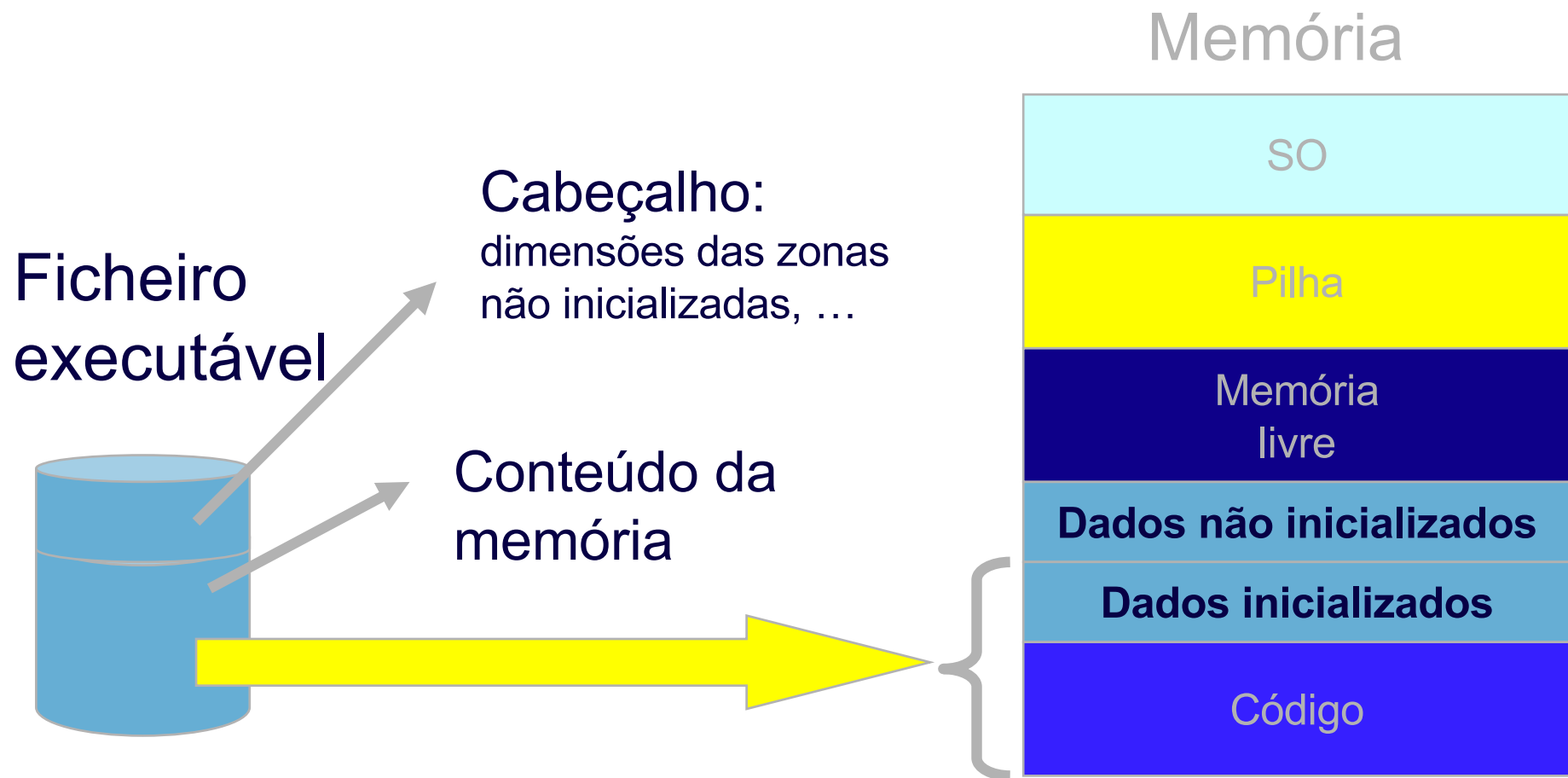
Instruções suportadas pelo sistema de operação, por exemplo as relacionadas com ficheiros

Instruções suportadas directamente pelo hardware



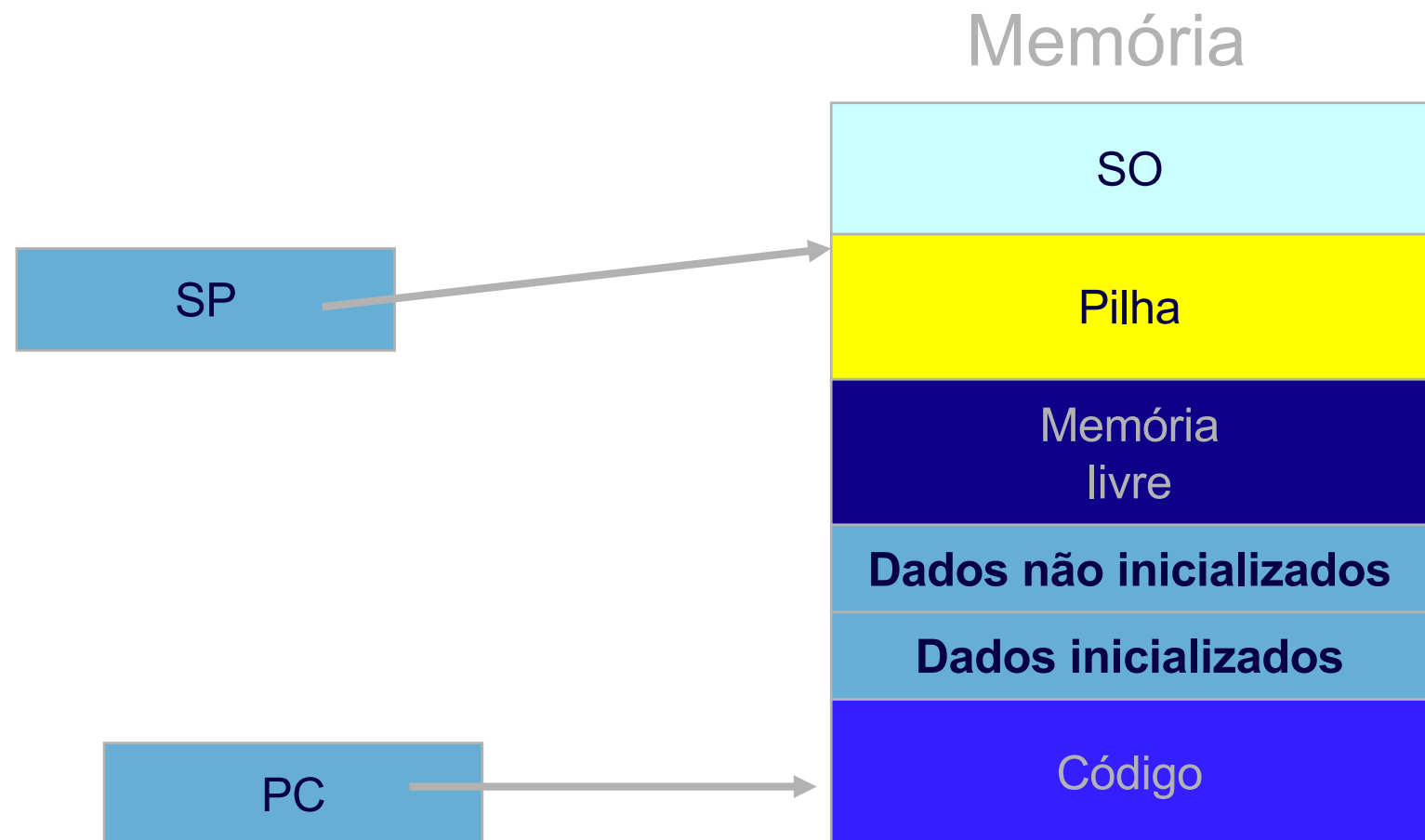
CARREGADOR (1)

- Na zona de memória livre para programas, o carregador coloca o código e os dados inicializados



CARREGADOR (2)

- O PC e o SP do CPU são inicializados de forma a que o programa acabado de carregar comece a executar

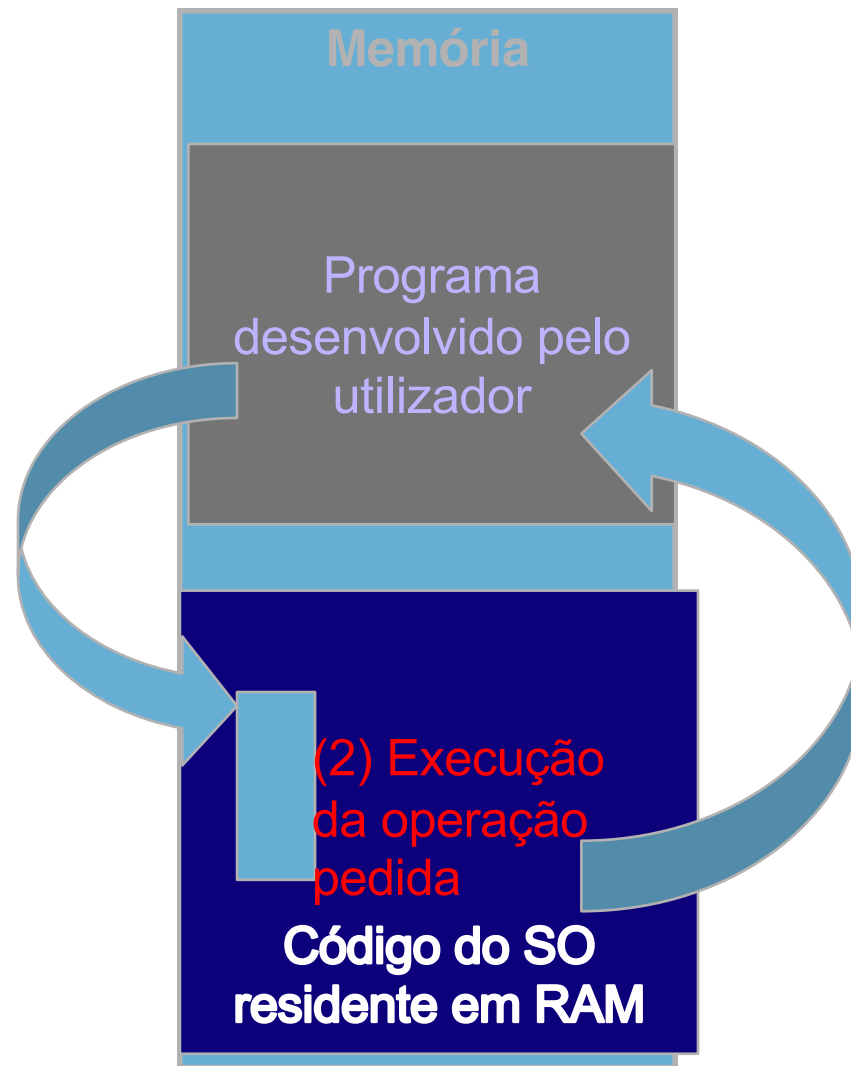


Programa em execução



E como é que se invocam os serviços do sistema operativo ?

(1) Invocação dos serviços do SO através de uma instrução máquina que salta para o código do SO. Antes guarda o valor corrente do PC (ou IP)



(3) Instrução máquina que coloca no PC (ou IP) o valor guardado em (1). O programa é retomado na instrução a seguir à invocação do SO