Arquitectura de Sistemas e Computadores I

Semana 1

Miguel Barão

u évora
ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

Resumo

Introdução histórica

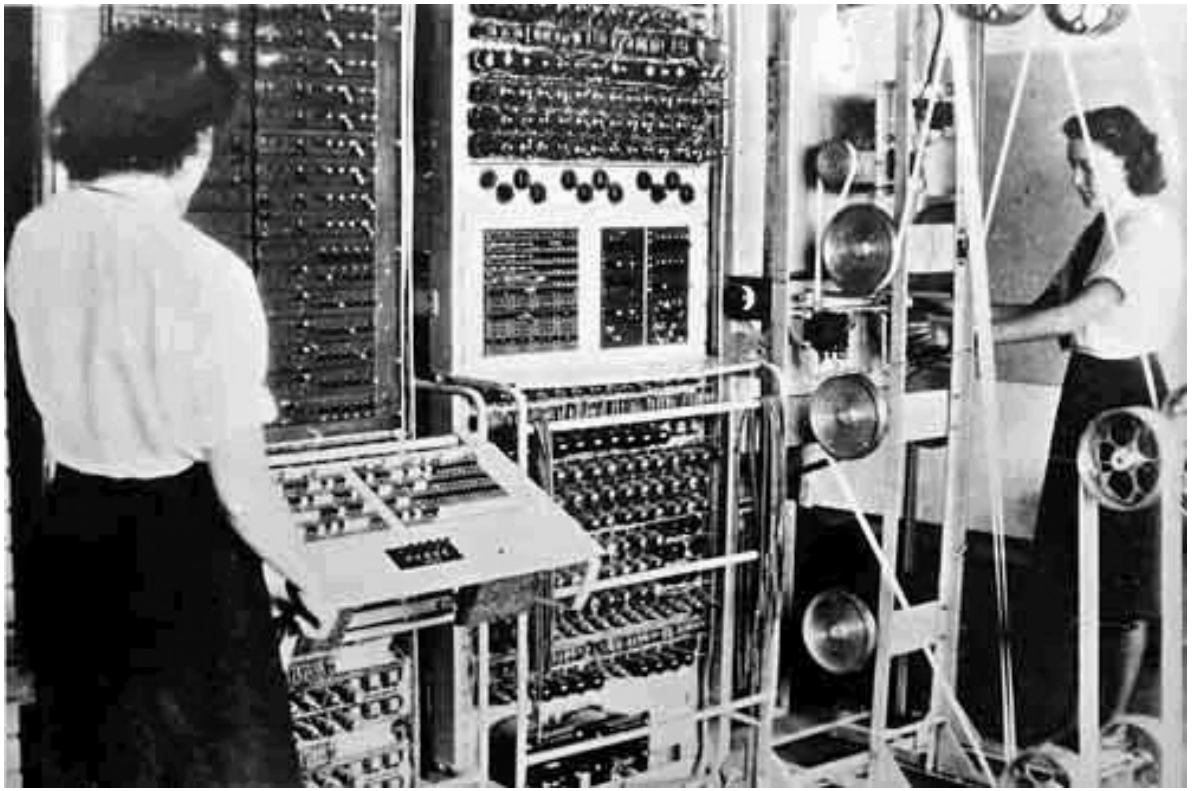
Arquitectura Von Neumann

Organização da memória

- Ordenação de Bytes (*endianness*)

Colossus

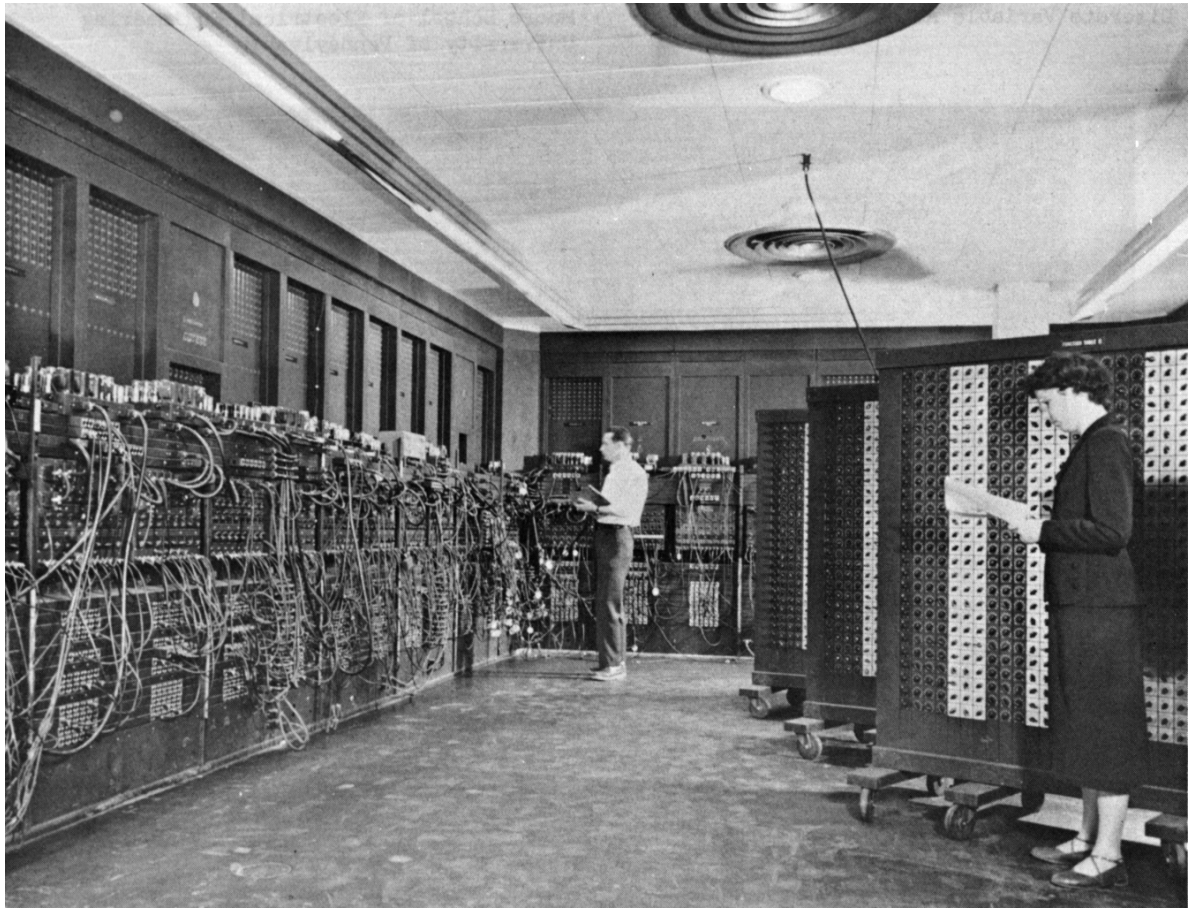
Usado pelos Britânicos para ler mensagens cifradas pelos Alemães durante a 2ª Grande Guerra, 1943–1945.



- Os primeiros computadores executavam programas fixos.
- Para correr um programa diferente era necessário modificar fisicamente um conjunto de interruptores e/ou a cablagem.
- Actualmente algumas calculadores simples também executam programas fixos.

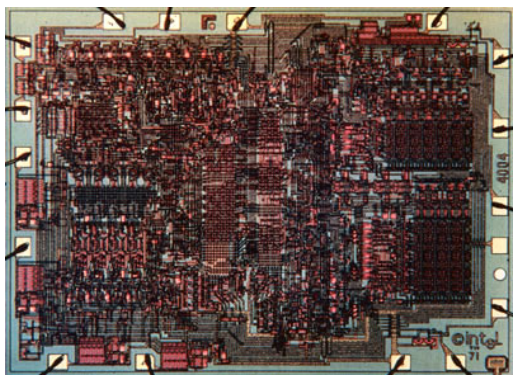
ENIAC (Electronic Numerical Integrator And Computer)

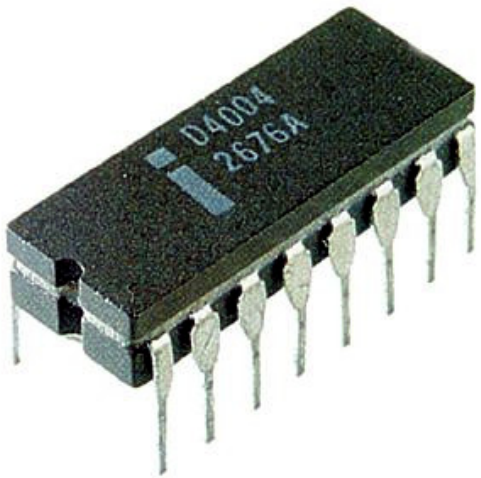
Usado pelos Estados Unidos para cálculos de balística, 1946–1955.



- Executava também programas fixos, tal como o Colossus.
- Para correr um programa diferente era necessário modificar fisicamente um conjunto de interruptores e/ou a cablagem.

Intel 4004
CPU de 4 bits desenvolvido em 1971.



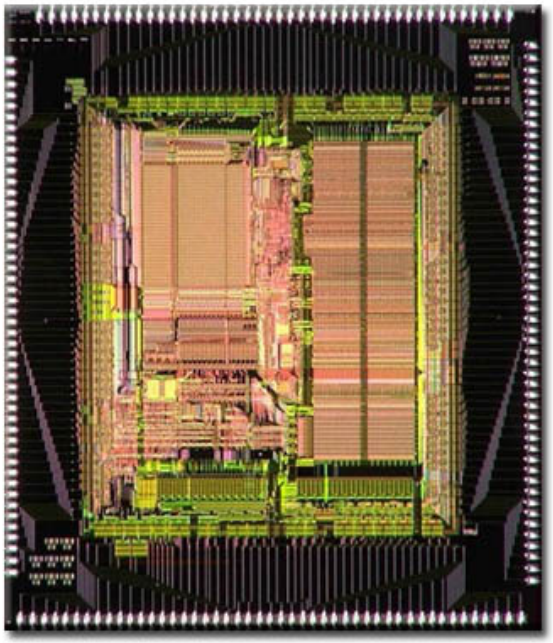


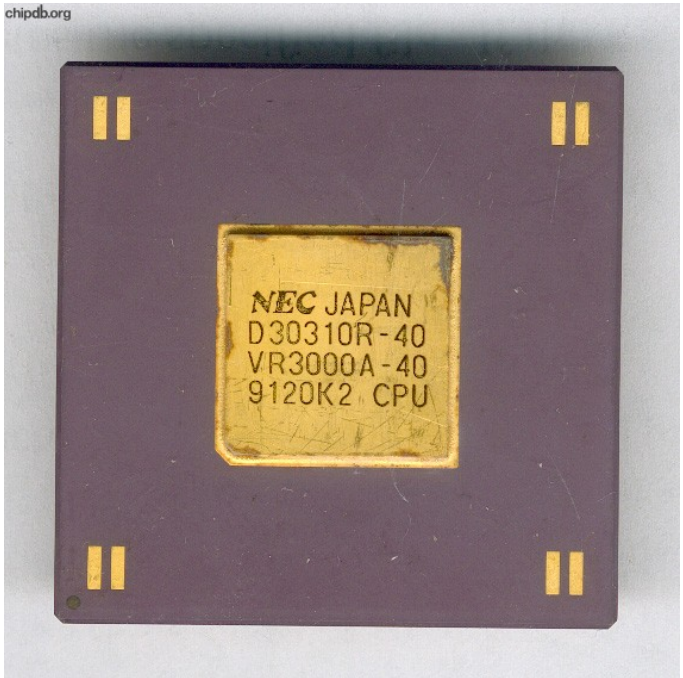
- Primeiro microprocessador comercialmente disponível.
- Ocupa uma área de $12mm^2$.
- Executa cerca de 92000 instr./s.

MOS Tech 6502
CPU de 8 bits desenvolvido em 1975.

- Processador muito barato. Custava $25 na altura.
- Deu origem à revolução dos computadores pessoais.
- Usado em:
  - Microcomputadores: BBC, Apple IIe, Atari 400/800 e Commodore 64.
  - Plataformas de jogos: Atari 2600 e Nintendo Entertainment System (NES).

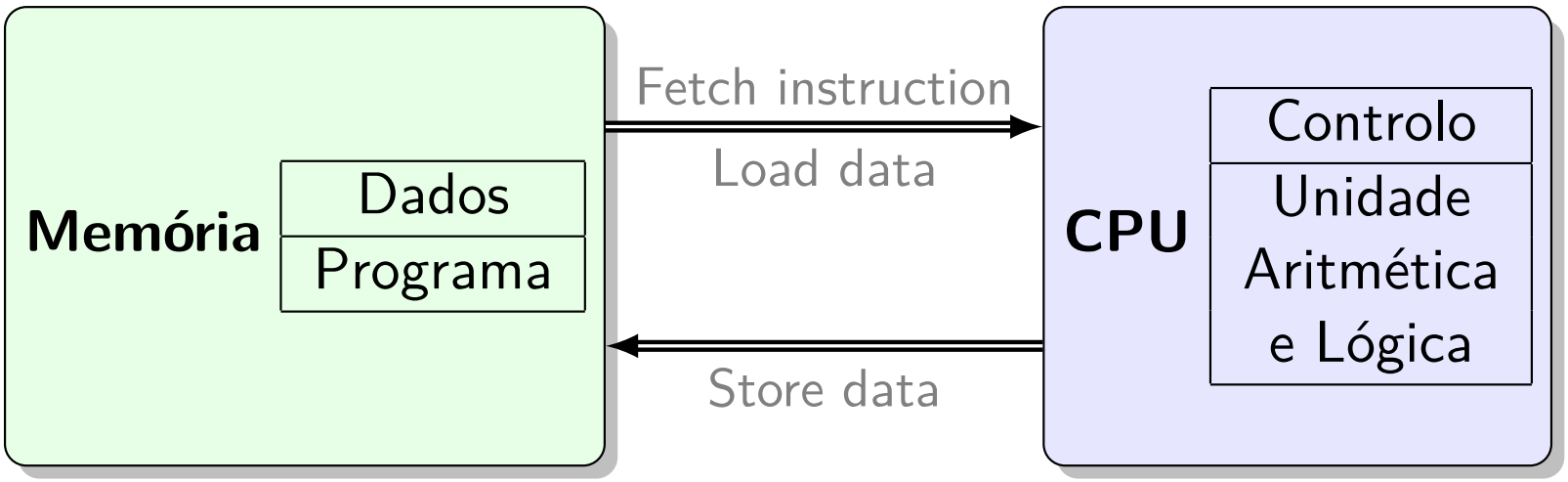
MIPS R3000
CPU de 32bits desenvolvido em 1988.





- Pioneiro na execução em pipeline.
- Processadores desta arquitectura são actualmente usados em: dispositivos de rede, consolas de jogos, impressoras, set-top boxes, televisões digitais e modems de cabo/ADSL.
- É o processador que vamos estudar em ASC1 e ASC2!

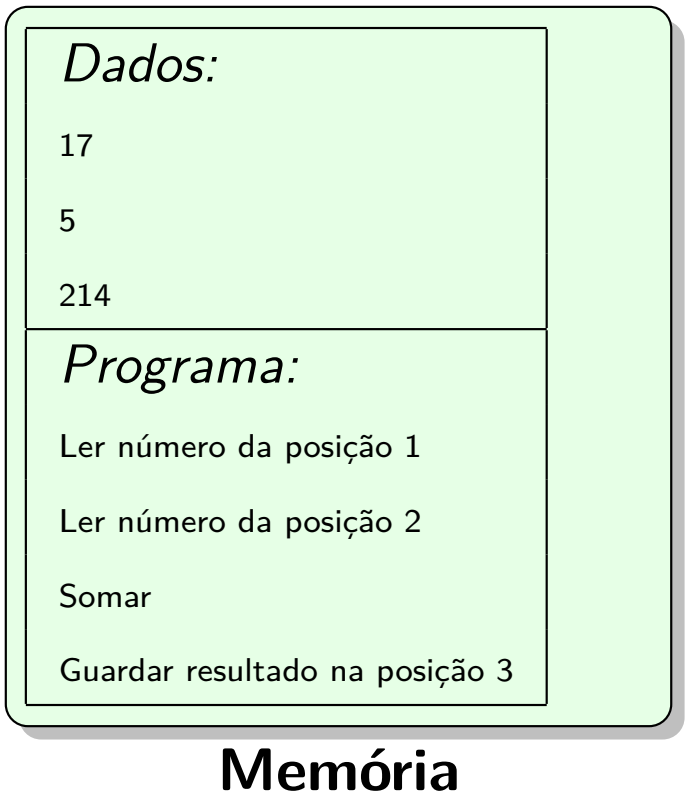
Stored Program Computers (arquitectura Von Neumann)

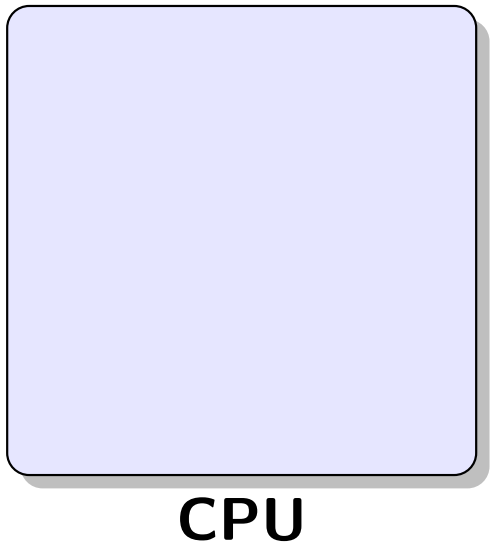


O programa e os dados são guardados conjuntamente em memória.

Exemplo de execução de um programa

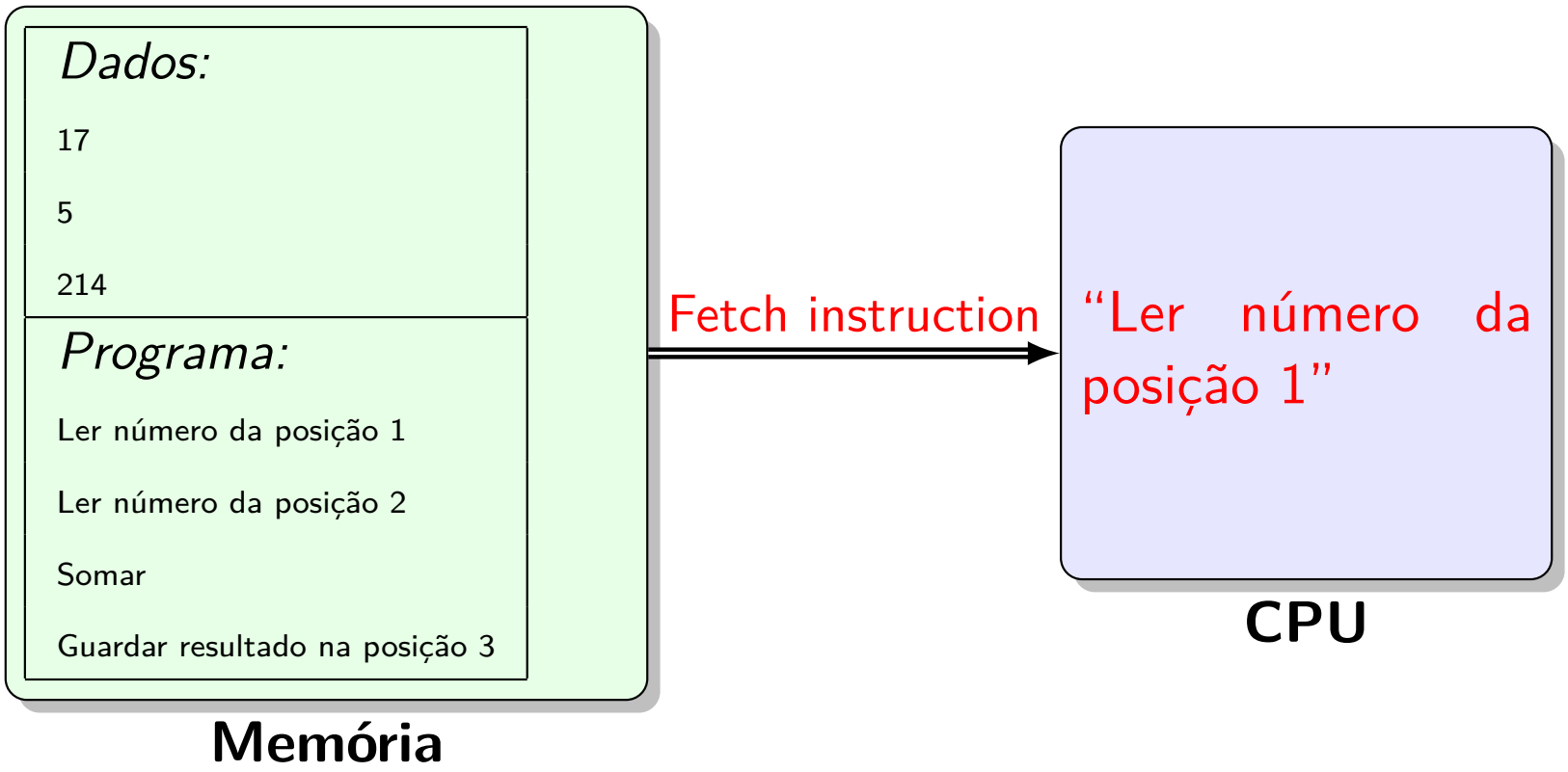
Pretende-se somar os números 17 e 5...





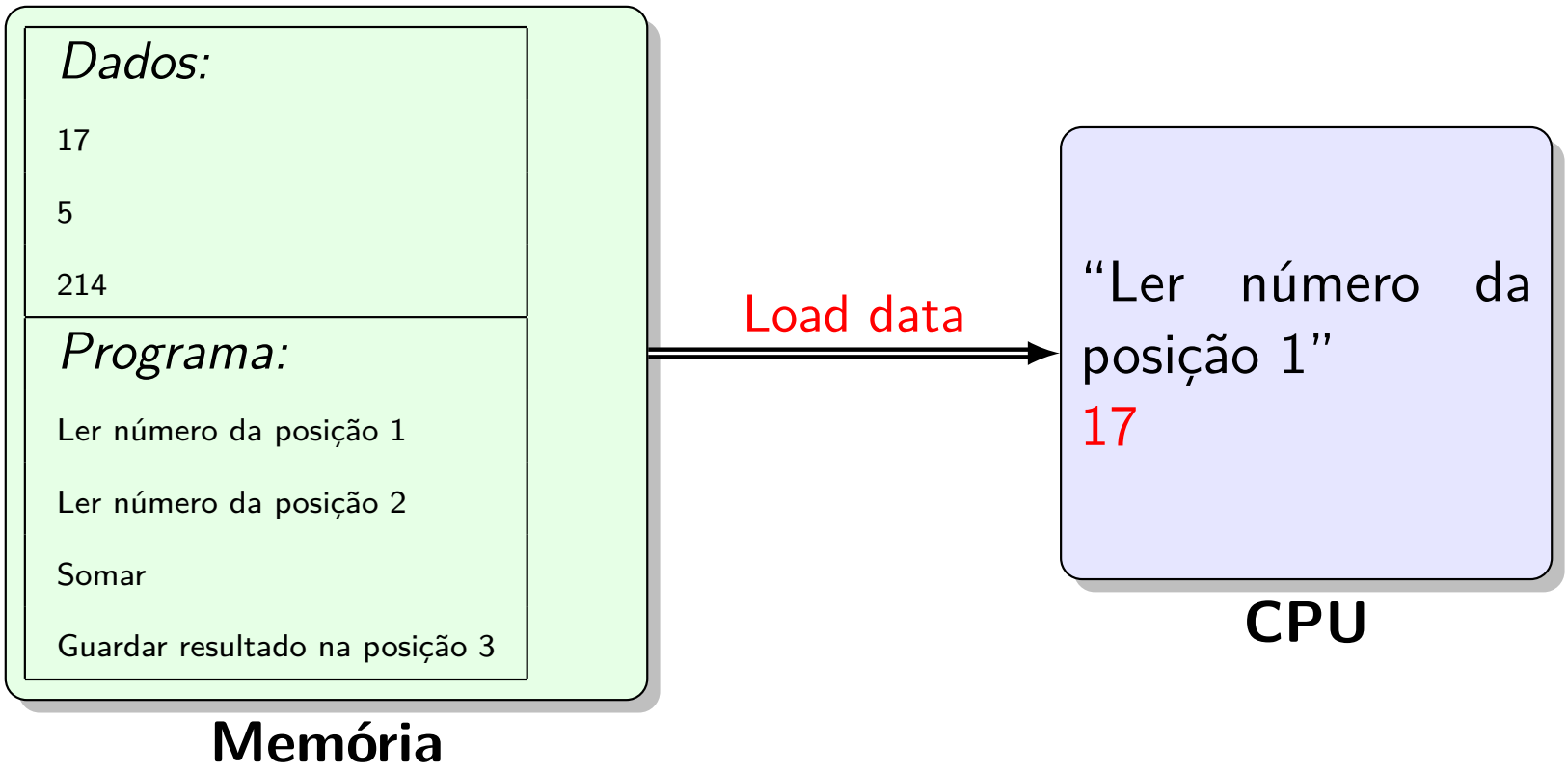
Exemplo de execução de um programa

Pretende-se somar os números 17 e 5...



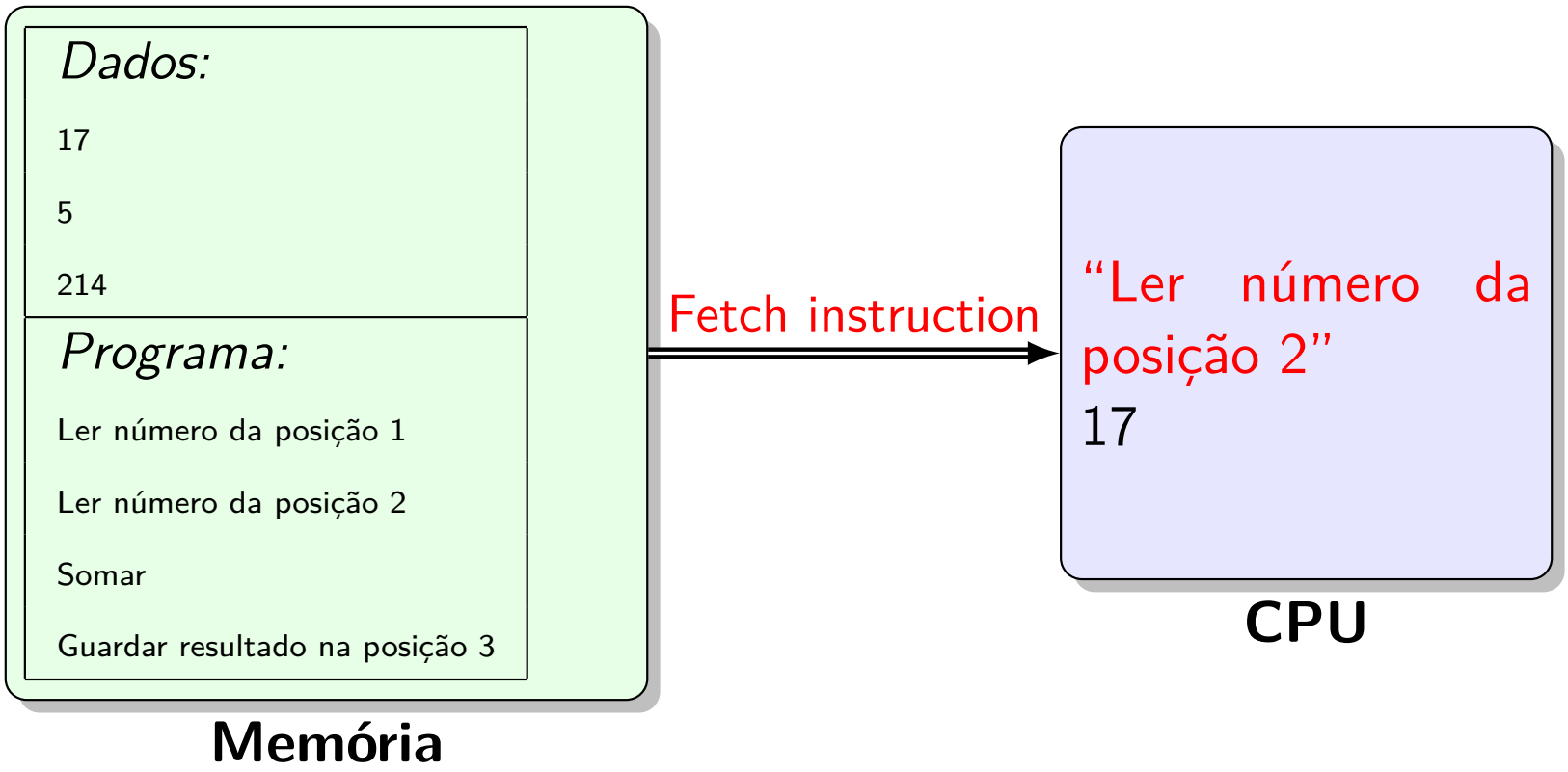
Exemplo de execução de um programa

Pretende-se somar os números 17 e 5...



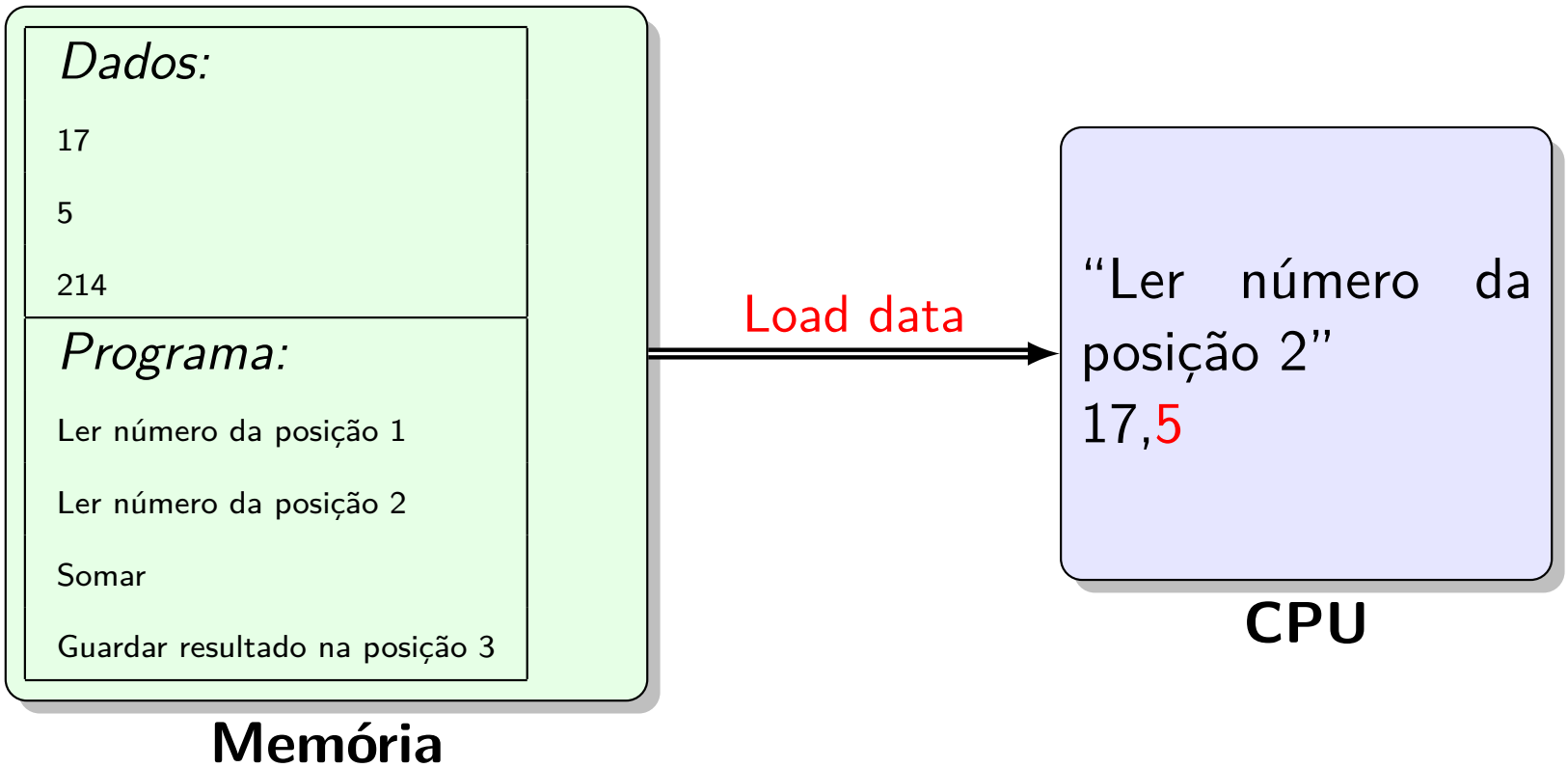
Exemplo de execução de um programa

Pretende-se somar os números 17 e 5...



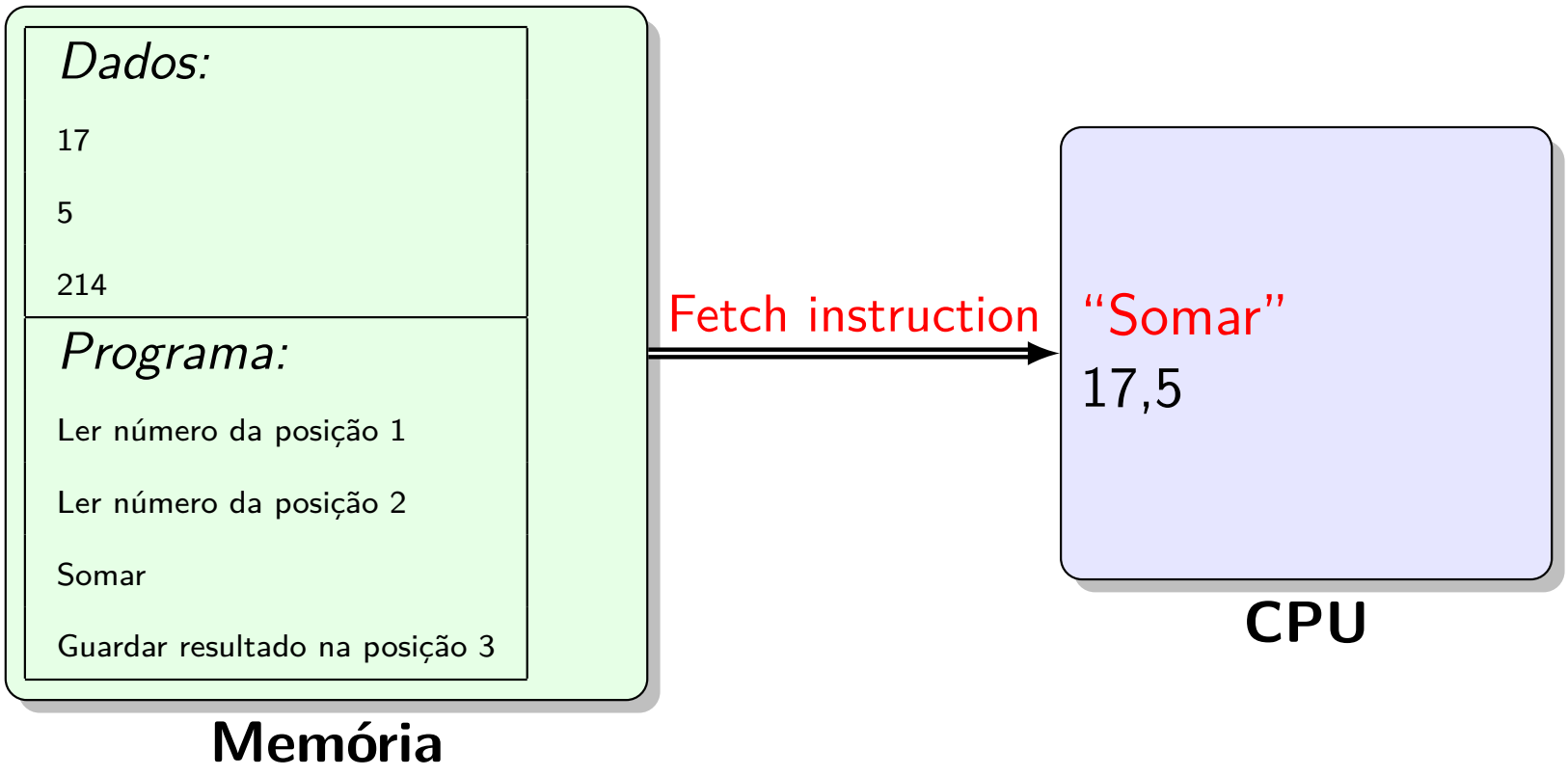
Exemplo de execução de um programa

Pretende-se somar os números 17 e 5...



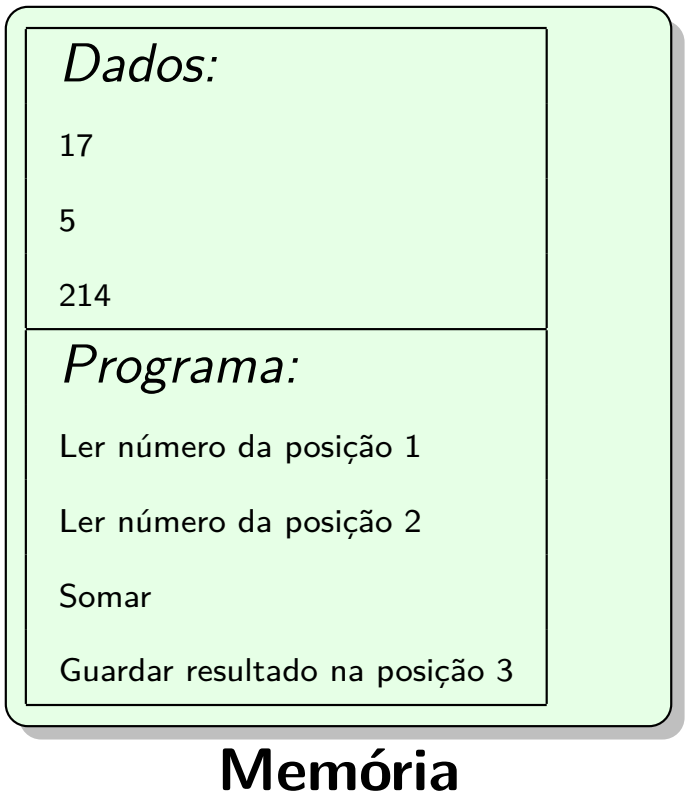
Exemplo de execução de um programa

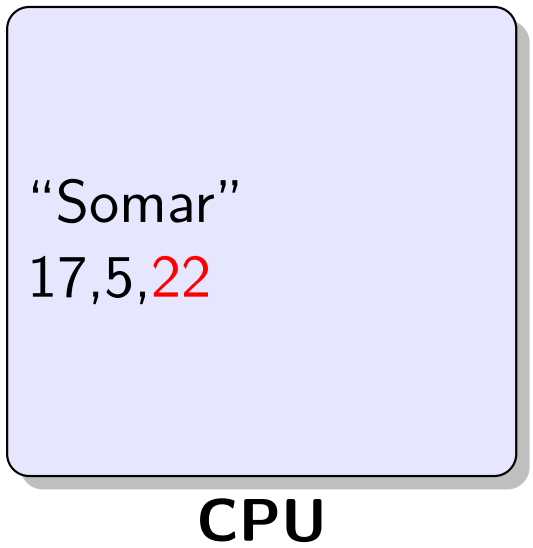
Pretende-se somar os números 17 e 5...



Exemplo de execução de um programa

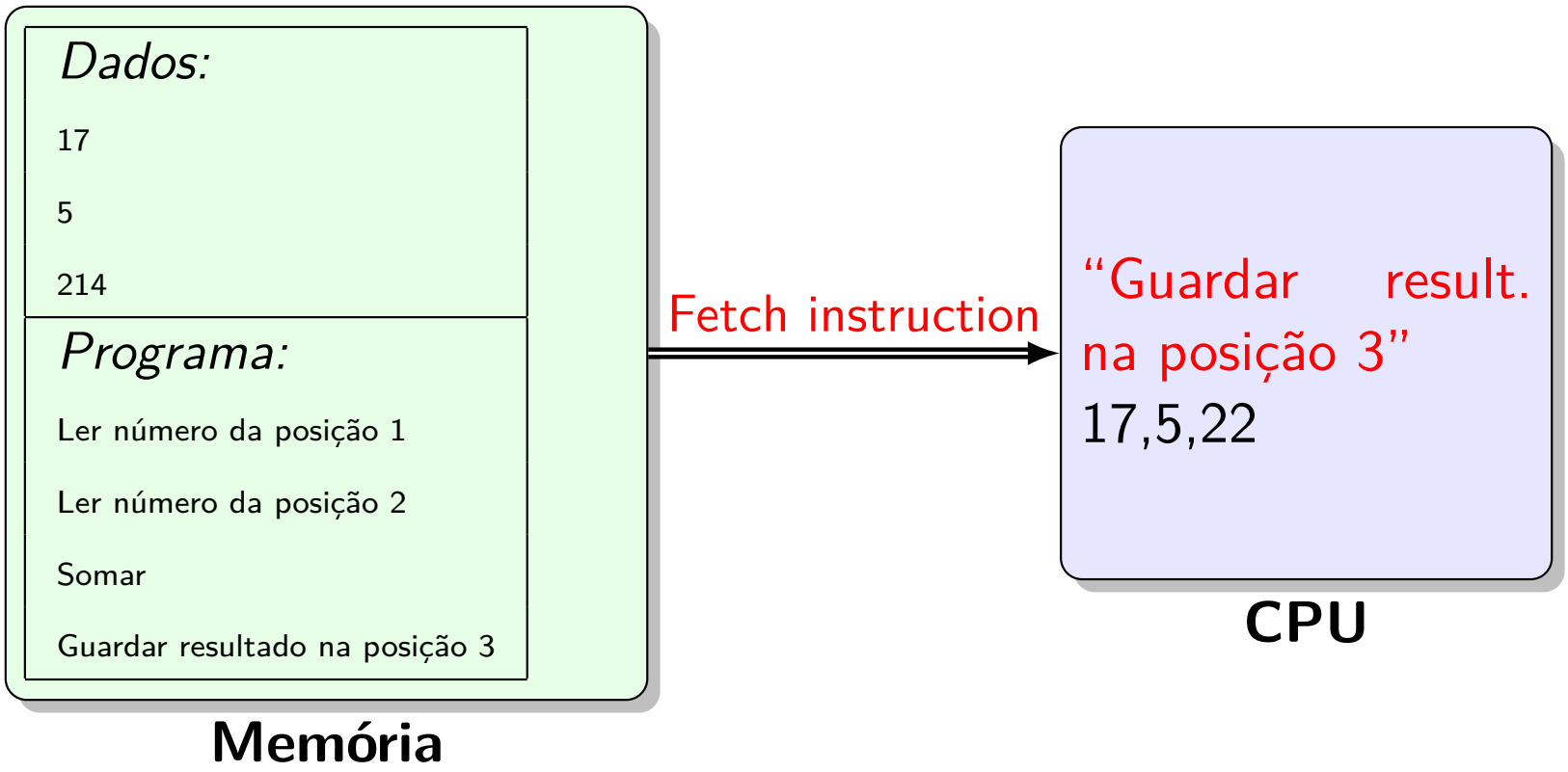
Pretende-se somar os números 17 e 5...





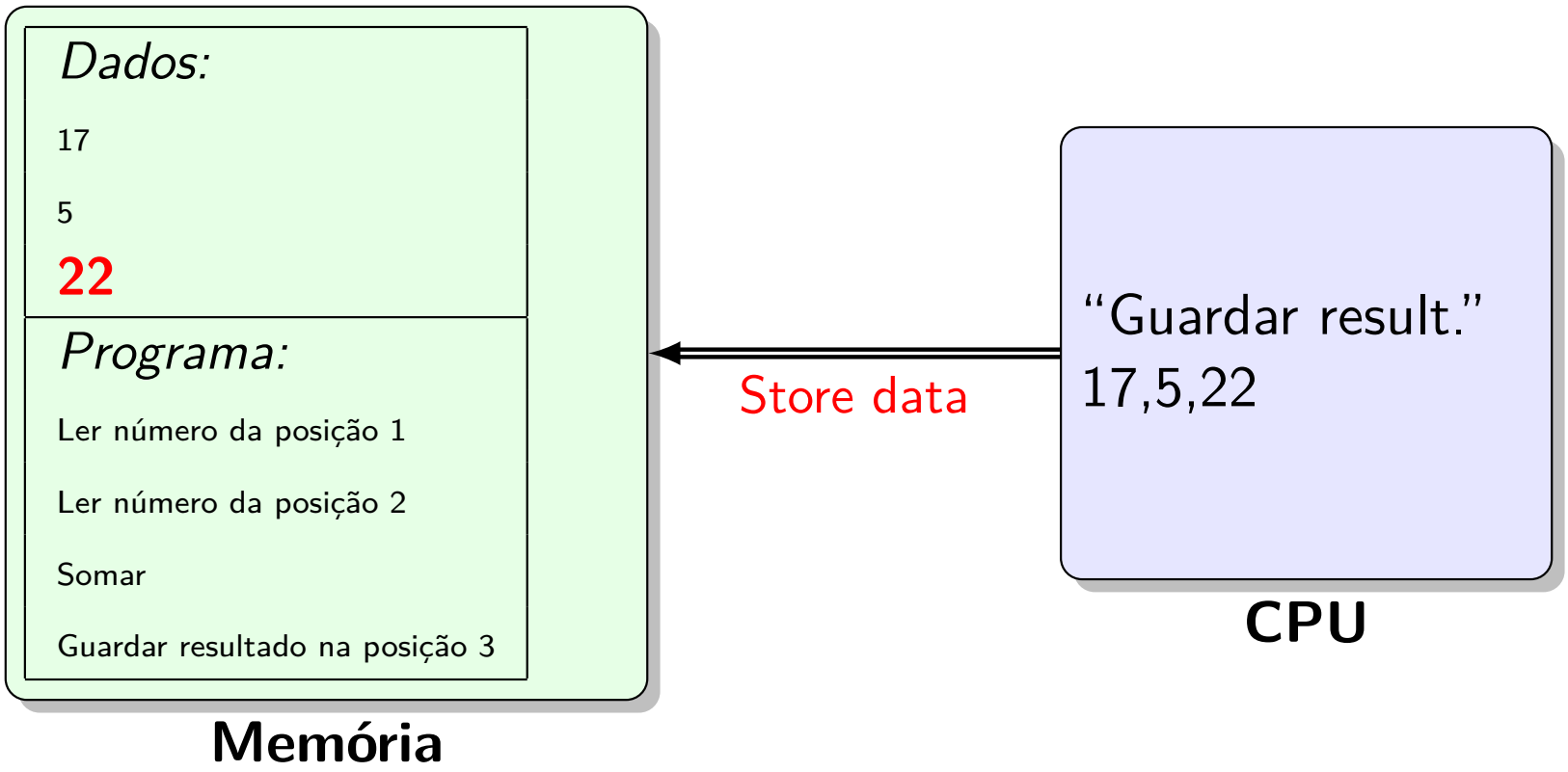
Exemplo de execução de um programa

Pretende-se somar os números 17 e 5...



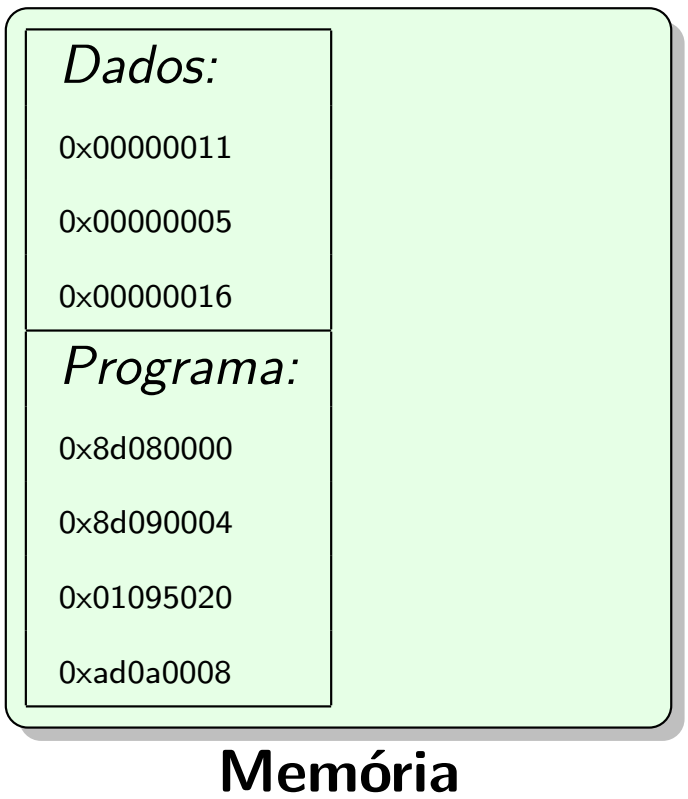
Exemplo de execução de um programa

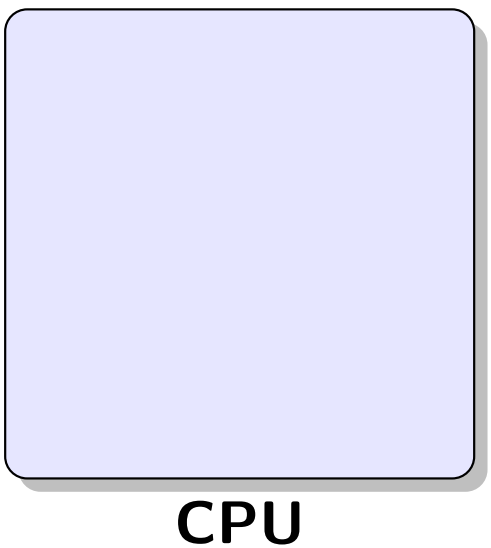
Pretende-se somar os números 17 e 5...



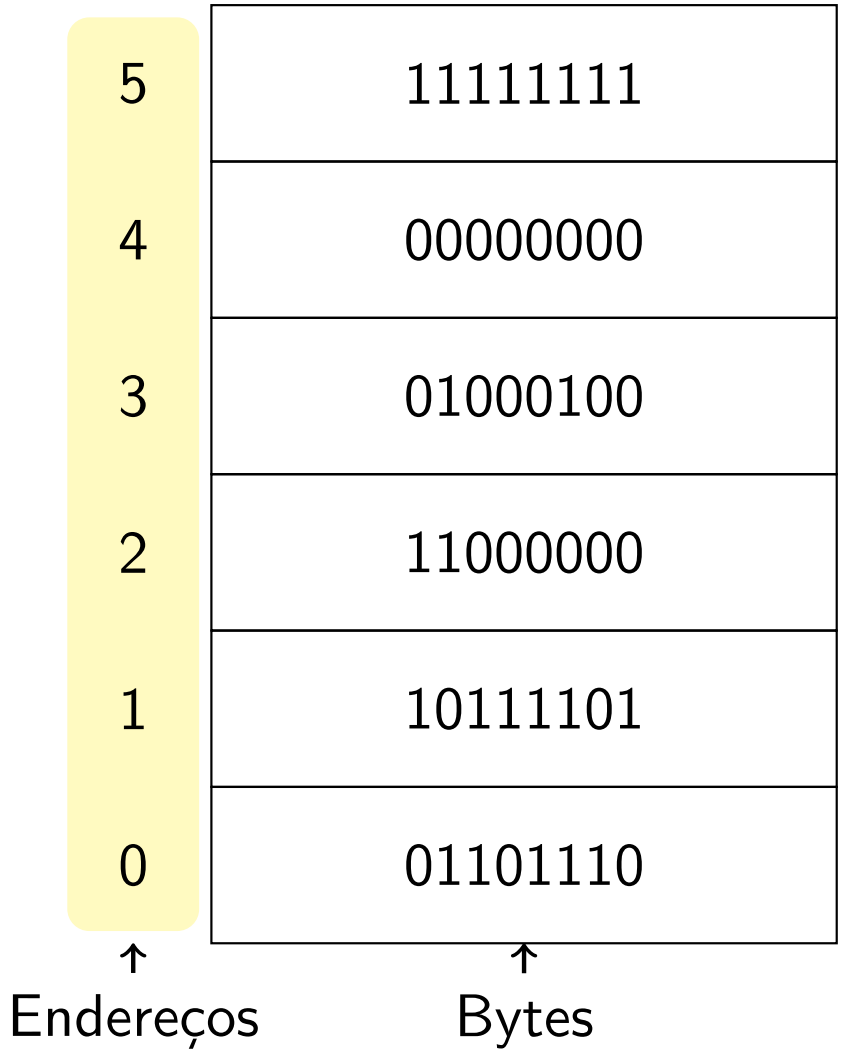
Exemplo de execução de um programa

Um programa é um conjunto de números em memória que codificam as instruções a executar $\rightarrow$ código máquina.
Na realidade, o conteúdo da memória é:





Memória: organização em Bytes



Funciona como uma tabela onde as linhas são numeradas e cada uma armazena exactamente 8 bits de informação (1 Byte).

Unidades de informação

1 bit é abreviatura de "binary unit".
Pode tomar os valores 0 ou 1. Um sistema digital funciona nesta base (base binária).

1 Byte tipicamente são 8 bits.
É a menor unidade de informação endereçável em memória.

1 word é a unidade de informação natural de uma arquitectura.
No caso da arquitectura MIPS estudada nesta disciplina 1 word = 32 bits, mas pode ter outros tamanhos noutras arquitecturas (16 bits, 64 bits, etc).

Unidades de informação

1 bit é abreviatura de "binary unit".
Pode tomar os valores 0 ou 1. Um sistema digital funciona nesta base (base binária).

1 Byte tipicamente são 8 bits.
É a menor unidade de informação endereçável em memória.

1 word é a unidade de informação natural de uma arquitectura.
No caso da arquitectura MIPS estudada nesta disciplina 1 word = 32 bits, mas pode ter outros tamanhos noutras arquitecturas (16 bits, 64 bits, etc).

(bit ← matemática, Byte ← memória, word ← processador)

Atenção

Não é possível endereçar de forma directa um bit em memória.
Um endereço apenas permite aceder a um bloco de 8 bits (1 Byte).

As operações permitidas sobre a memória são:

- Ler um Byte da memória (*load Byte*)
- Escrever um Byte na memória (*store Byte*)

Também é possível escrever vários Bytes de uma vez.
Por exemplo:

- Ler uma word da memória (*load word*)
- Escrever uma word na memória (*store word*)

Exemplo

Suponha que a memória de um computador contém a seguinte sequência de bits:

$$000001110000000011111111100000001111111\ldots$$

1. Qual o conteúdo da memória nos endereços 0, 1, 2, 3, 4 e 5?
2. Represente em decimal e hexadecimal o conteúdo desses endereços.
3. Se os Bytes representam números inteiros em complemento para 2, que números são?

Representação de números nas bases binária e hexadecimal

Para distinguir as bases de numeração usamos prefixos:

- Hexadecimais com prefixo 0x.
- Binários com prefixo 0b.
- Decimais não usam prefixo.

Exemplos

Decimal	Hexadecimal	Binário
1	0x01	0b00000001
15	0x0f	0b00001111
16	0x10	0b00010000
128	0x80	0b10000000
255	0xff	0b11111111

Representação de números nas bases binária e hexadecimal

Para distinguir as bases de numeração usamos prefixos:

- Hexadecimais com prefixo 0x.
- Binários com prefixo 0b.
- Decimais não usam prefixo.

Exemplos

Decimal	Hexadecimal	Binário
1	0x01	0b00000001
15	0x0f	0b00001111
16	0x10	0b00010000
128	0x80	0b10000000
255	0xff	0b11111111

Atenção

Num computador todos os números estão em binário. Apenas nós, os humanos, usamos as bases decimal e hexadecimal por uma questão de legibilidade.

Memória: ordenação de Bytes

Problema

Quando se pretende escrever um número com mais de um Byte para a memória surge um problema:

- Qual a ordenação dos Bytes a usar? *i.e.*, deve colocar-se primeiro os Bytes mais significativos ou menos significativos?

Problema

Quando se pretende escrever um número com mais de um Byte para a memória surge um problema:

- Qual a ordenação dos Bytes a usar? *i.e.*, deve colocar-se primeiro os Bytes mais significativos ou menos significativos?

Definição (Endianness)

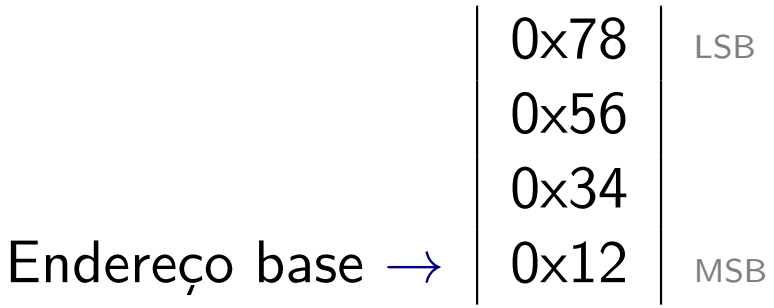
Existem duas convenções em uso:

Little endian o Byte menos significativo primeiro.

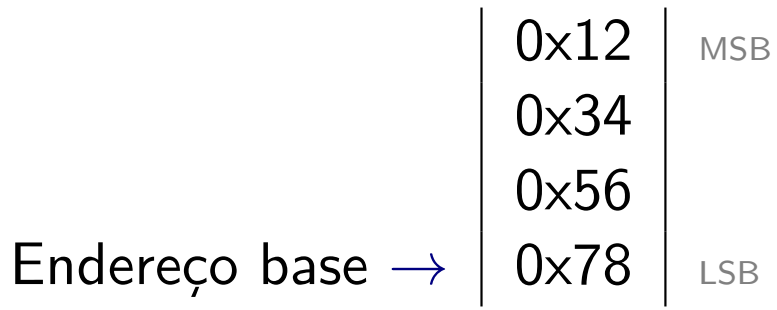
Big endian o Byte mais significativo primeiro.

Memória: ordenação de Bytes

Como escrever o número $0x12345678$ em memória?



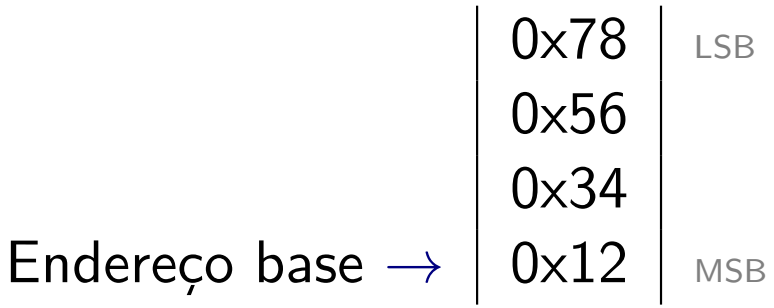
?



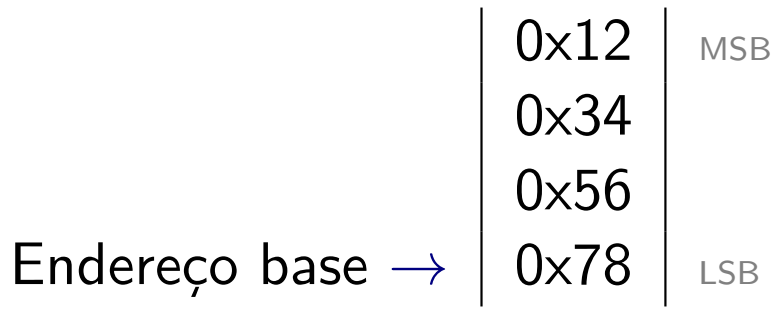
?

Memória: ordenação de Bytes

Como escrever o número $0x12345678$ em memória?



Big Endian



Little Endian

Memória: ordenação de Bytes

A definição e o tratamento correcto da ordenação de Bytes é especialmente importante quando é necessário transmitir informação entre máquinas com ordenações diferentes. Por exemplo:

- ler/escrever um ficheiro (*e.g.* música, imagem, etc).
- enviar informação pela rede.

Memória: ordenação de Bytes

A definição e o tratamento correcto da ordenação de Bytes é especialmente importante quando é necessário transmitir informação entre máquinas com ordenações diferentes. Por exemplo:

- ler/escrever um ficheiro (*e.g.* música, imagem, etc).
- enviar informação pela rede.

Arquitecturas diferentes usam ordenações diferentes:

- Little endian é usado na arquitectura Intel x86.
- Big endian é usado na arquitectura PowerPC.
- Algumas arquitecturas suportam ambos os modos. É o caso das arquitecturas MIPS e ARM. São usados os termos MIPSEL e ARMEL para realçar que a ordenação de Bytes usada é little endian.

Prefixos binários e decimais

Quando uma grandeza é muito grande podem usar-se prefixos:

SI			IEC[1]		
$10^0 = 1$	(sem prefixo)	Unidade	2^0		
$10^3 = 1000$	k	kilo	$2^{10} = 1024$	Ki	kibi
10^6	M	mega	2^{20}	Mi	mebi
10^9	G	giga	2^{30}	Gi	gibi
10^{12}	T	tera	2^{40}	Ti	tebi
10^{15}	P	peta	2^{50}	Pi	pebi
10^{18}	E	exa	2^{60}	Ei	exbi
10^{21}	Z	zetta	2^{70}	Zi	zebi
10^{24}	Y	yotta	2^{80}	Yi	yobi

Pergunta: Uma memória com 2 GiB tem quantos bits?

[1] Standard ISO/IEC 80000

Prefixos binários e decimais

Quando uma grandeza é muito grande podem usar-se prefixos:

SI			IEC[1]		
$10^0 = 1$	(sem prefixo)	Unidade	2^0		
$10^3 = 1000$	k	kilo	$2^{10} = 1024$	Ki	kibi
10^6	M	mega	2^{20}	Mi	mebi
10^9	G	giga	2^{30}	Gi	gibi
10^{12}	T	tera	2^{40}	Ti	tebi
10^{15}	P	peta	2^{50}	Pi	pebi
10^{18}	E	exa	2^{60}	Ei	exbi
10^{21}	Z	zetta	2^{70}	Zi	zebi
10^{24}	Y	yotta	2^{80}	Yi	yobi

Pergunta: Uma memória com 2 GiB tem quantos bits?

Resposta: $2\,GiB = 2 \times 2^{30} \times 8\ bits =$

[1] Standard ISO/IEC 80000

Prefixos binários e decimais

Quando uma grandeza é muito grande podem usar-se prefixos:

SI			IEC[1]		
$10^0 = 1$	(sem prefixo)	Unidade	2^0		
$10^3 = 1000$	k	kilo	$2^{10} = 1024$	Ki	kibi
10^6	M	mega	2^{20}	Mi	mebi
10^9	G	giga	2^{30}	Gi	gibi
10^{12}	T	tera	2^{40}	Ti	tebi
10^{15}	P	peta	2^{50}	Pi	pebi
10^{18}	E	exa	2^{60}	Ei	exbi
10^{21}	Z	zetta	2^{70}	Zi	zebi
10^{24}	Y	yotta	2^{80}	Yi	yobi

Pergunta: Uma memória com 2 GiB tem quantos bits?

Resposta: $2GiB = 2 \times 2^{30} \times 8\ bits = 2^1 \times 2^{30} \times 2^3\ bits =$

[1] Standard ISO/IEC 80000

Prefixos binários e decimais

Quando uma grandeza é muito grande podem usar-se prefixos:

SI			IEC[1]		
$10^0 = 1$	(sem prefixo)	Unidade	2^0		
$10^3 = 1000$	k	kilo	$2^{10} = 1024$	Ki	kibi
10^6	M	mega	2^{20}	Mi	mebi
10^9	G	giga	2^{30}	Gi	gibi
10^{12}	T	tera	2^{40}	Ti	tebi
10^{15}	P	peta	2^{50}	Pi	pebi
10^{18}	E	exa	2^{60}	Ei	exbi
10^{21}	Z	zetta	2^{70}	Zi	zebi
10^{24}	Y	yotta	2^{80}	Yi	yobi

Pergunta: Uma memória com 2 GiB tem quantos bits?

Resposta: $2GiB = 2 \times 2^{30} \times 8\ bits = 2^1 \times 2^{30} \times 2^3\ bits = 2^{34}\ bits$.

[1] Standard ISO/IEC 80000

Utilização dos prefixos binário e decimal

- A utilização do standard é relativamente recente (2008), pelo que é necessário alguma cautela.

Utilização dos prefixos binário e decimal

- A utilização do standard é relativamente recente (2008), pelo que é necessário alguma cautela.

Alguns exemplos:

Tamanhos dos ficheiros Varia com S.O. e programa usado.

Utilização dos prefixos binário e decimal

- A utilização do standard é relativamente recente (2008), pelo que é necessário alguma cautela.

Alguns exemplos:

Tamanhos dos ficheiros Varia com S.O. e programa usado.

Memória usam-se sempre prefixos binários IEC, e.g. 4 GiB.

Utilização dos prefixos binário e decimal

- A utilização do standard é relativamente recente (2008), pelo que é necessário alguma cautela.

Alguns exemplos:

Tamanhos dos ficheiros Varia com S.O. e programa usado.

Memória usam-se sempre prefixos binários IEC, e.g. 4 GiB.

Discos rígidos fabricantes usam sempre prefixos decimais, e.g. disco de 320 GB.
(mas programas diferentes apresentam o tamanho do disco usando o prefixos diferentes...)

Utilização dos prefixos binário e decimal

- A utilização do standard é relativamente recente (2008), pelo que é necessário alguma cautela.

Alguns exemplos:

Tamanhos dos ficheiros Varia com S.O. e programa usado.

Memória usam-se sempre prefixos binários IEC, e.g. 4 GiB.

Discos rígidos fabricantes usam sempre prefixos decimais, e.g. disco de 320 GB.
(mas programas diferentes apresentam o tamanho do disco usando o prefixos diferentes...)

CD usa prefixo binário, e.g. 700 MiB.

Utilização dos prefixos binário e decimal

- A utilização do standard é relativamente recente (2008), pelo que é necessário alguma cautela.

Alguns exemplos:

Tamanhos dos ficheiros Varia com S.O. e programa usado.

Memória usam-se sempre prefixos binários IEC, e.g. 4 GiB.

Discos rígidos fabricantes usam sempre prefixos decimais, e.g. disco de 320 GB.
(mas programas diferentes apresentam o tamanho do disco usando o prefixos diferentes...)

CD usa prefixo binário, e.g. 700 MiB.

DVD, Blu-ray usa prefixo decimal, e.g. 4.7 GB.

Utilização correcta

Prefixo binário para a memória, decimal para tudo o resto.

Problema

1. *Considere os números 127, 77, 1024, 0.*
   - 1.1 *Escreva-os em binário e em hexadecimal.*
   - 1.2 *Represente-os de modo a ocuparem quatro endereços de memória consecutivos.*
   - 1.3 *Repita para o simétrico de cada um dos números.*
2. *Quatro endereços de memória consecutivos contêm 0x01, 0x7f, 0xfc, 0x10, respectivamente. Sabendo que a ordenação de Bytes é Little Endian e que os quatro Bytes representam um número inteiro de 32 bits, escreva o número correspondente em hexadecimal.*
3. *Quantos bits contém um ficheiro de 4 MiB?*