

Conceitos básicos

Programação I
2020.2021

Teresa Gonçalves
tcg@uevora.pt

Departamento de Informática, ECT-UÉ

Sumário

Modelo Von Neumann

Linguagem C

Valores

Variáveis

Operadores, Operandos e Expressões

Instruções

Comentários

Modelo Von Neumann

Modelo von Neumann

Modelo de um computador

Proposta em 1945

Baseada no conceito de computador de armazenamento de programas

Componentes

Unidade central de processamento (CPU, central processing unit)

Unidade de memória

Dispositivos de entrada e de saída

CPU – central processing unit

O que é?

Circuito eletrónico responsável pela execução das instruções de um programa

Também conhecido como

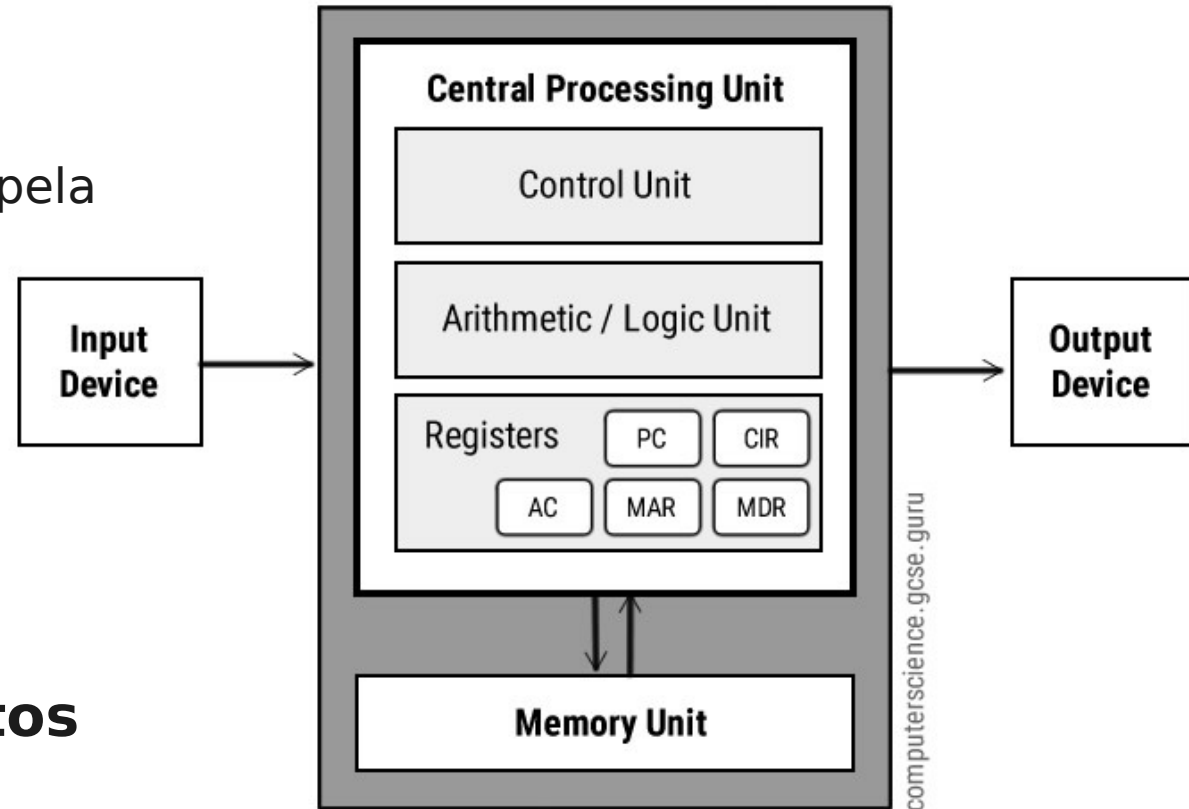
Microprocessador, processador

Contém diversos elementos

Unidade de controlo

Unidade de aritmética e lógica (ALU)

Registos



Elementos da CPU

Unidade de controlo

Controla a operação da ALU, memória e dispositivos de entrada/saída
Fornece sinais de cronometragem e de controlo necessários pelos outros componentes

Unidade de Aritmética e Lógica

Permite a execução de operações
Aritméticas (soma, subtração, etc...)
Lógicas (and, or, not, etc)

Registos

Áreas de armazenamento de alta velocidade

Unidade de memória

Memória principal ou primária (RAM)

Memória rápida e diretamente acessível pela CPU

Está dividida em unidades

Cada unidade é composta por

Endereço

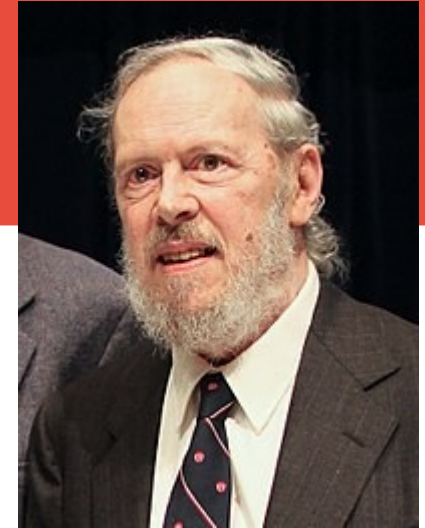
identifica univocamente cada posição de memória

Conteúdo

Valor armazenado

Linguagem C

História



Informação

Criada em 1972

Em AT&T Bell Labs

Por Dennis Ritchie

Para o desenvolvimento do SO Unix

Versões

1973: K&R

1989: ANSI C

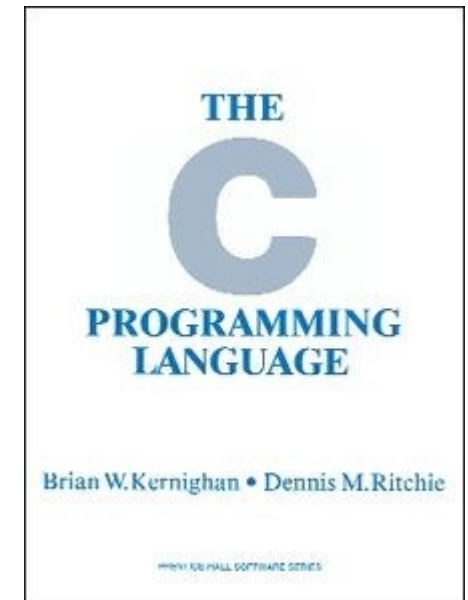
Atualmente: C18 standard revision (junho 2018)

Livro de referência

The C Programming Language

Brian Kernighan e Dennis Ritchie (1978)

serviu durante muitos anos como uma especificação informal da linguagem (K&R C)



Valores

Valor

Exemplos

1

-20

3.141519

'c'

Tem um tipo

int

float

char

Variáveis

Variável

Local em memória

Permite armazenar um valor

Referenciado por um nome

Tem um tipo

int

float

char

Nome de variável

Qualquer comprimento

Qualquer identificador válido, exceto

Palavras reservadas

Dígito no início

Case sensitive

Deve

ser **sugestivo**

porque facilita a leitura e compreensão do código

usar-se sempre o mesmo raciocínio na sua definição

Declaração

Antes de utilizar a variável é preciso declará-la

Declaração de variável

Indicar tipo e nome

Exemplos

int num

float pi

char letra

Atribuição

variavel = expressão

Instrução que associa um valor (direita) à variável (esquerda)

Exemplos

num = 5

pi = 3.1415926535897931

letra = 'u'

Operadores, Operandos e Expressões

Operadores, operandos e expressões

Operador

Símbolo que representa uma computação

Operando

Valor/variável à qual são aplicados os operadores

Expressão

Combinação de valores, variáveis e operadores

Operadores aritméticos

- + soma
- subtração
- / divisão
- * multiplicação
- % resto da divisão inteira

Precedência

Regras que definem a ordem de avaliação das expressões

Parêntesis	↑	<i>Maior precedência</i>
Multiplicação e divisão		
Soma e subtração	↓	<i>Menor precedência</i>

Operadores relacionais

==	igual
!=	diferente
<	menor que
<=	menor que ou igual
>	maior que
>=	maior que ou igual

Operadores lógicos

&& e

|| ou

! não (negação)

Precedência

!

&&

||

↑ *Maior precedência*

↓ *Menor precedência*

Expressão

Uma expressão com operadores numéricos é do tipo dos seus operandos

Exemplos

$3+5*6$

$3.16/2.0$

$a-5$

Programa e instruções

Instrução e programa

Instrução

Uma unidade de código que é executada

As instruções terminam com ;

Exemplos

$a = 3;$

$y = x * a;$

Programa

Uma sequência de instruções

A execução do programa começa pela 1ª instrução e continua com a 2ª, etc

Comentários

`/*` início de comentário

`*/` fim de comentário

São anotações em linguagem natural
ajudam a entender o código fonte

Escrita de dados

printf

Escrita (formatada) de dados no ecrã

Exemplos

```
printf( "%d %f", x, y );
```

Escreve no ecrã o valor de x seguido de y; x é inteiro, y é real

```
printf( "número: %d", num );
```

Escreve no ecrã o texto "número: " seguido do valor da variável num

%d é um especificador de formato

Especificadores de formato

%d **int**

%f **float**

%c **carácter**

Leitura de dados

scanf

Leitura (formatada) de dados do teclado

Exemplo

```
scanf( "%d %f", &inteiro, &real )
```

Lê um inteiro e um valor real do teclado e coloca-o nas variáveis inteiro e real, respectivamente

& é um operador que indica o endereço/posição de memória da variável; nesta posição é guardado o valor introduzido

Lembrete

Variável

Posição de memória; permite armazenar um valor
Identificada por um nome

Atribuição

`variavel = expressao`

A expressão pode conter valores, operadores e/ou variáveis
nome da variável

À esquerda do = refere a posição de memória (L-value)

À direita do = refere o valor (R-value)

Programa

```
int main() {
```

Declaração de variáveis

Ler dados

Processar dados

Escrever resultados

```
    return 0;
```

```
}
```

Conversão dolares → euros

Dado um valor em dolares (usd), mostrar o correspondente em euros (eur).

Considere 1 usd = 0.85 eur

Pedir o valor em dolares e guardar na variável

Calcular o valor correspondente em euros

Mostrar o valor calculado

Conversão dólares → euros

```
int main() {
```

```
float usd;
```

```
float eur;
```

declarar

```
printf( "Qual o valor em usd? ");
```

```
scanf( "%f", &usd );
```

ler dados

```
eur = usd * 0.85;
```

processar

```
printf( "%f", eur );
```

*escrever
resultados*

```
return 0;
```

```
}
```

Troca de valores

Trocar o valor de duas variáveis x e y

Pedir os valores ao utilizador e colocá-los nas variáveis x e y

Trocar os valores das variáveis

Mostrar que os valores estão trocados

Troca de valores

Considere que x tem o valor 3.5 e y tem o valor -0.7

```
int main() {  
    float x, y, aux;  
    printf( "Introduza os valores para x e y. Considere valores reais." );  
    scanf( "%f %f", &x, &y );  
    aux = x;  
    x = y;  
    y = aux;  
    printf( "x=%d y=%d", x, y );  
    return 0;  
}
```

