AGT – Algoritmos

Introdução e conceitos básicos

Prof. Allan Rodrigo Leite

Organização básica de um computador

Processador

Memória principal

Canal de comunicação

Memória secundária

Dispositivos de entrada

Dispositivos de saída

Processador

- Principal componente de um computador
- Executa sequências de operações muito simples e precisas
  - Sempre uma por vez
- Velocidade é medida em ciclos
  - Um I7 é capaz de executar cerca de 112.000.000.000 operações matemáticas por segundo

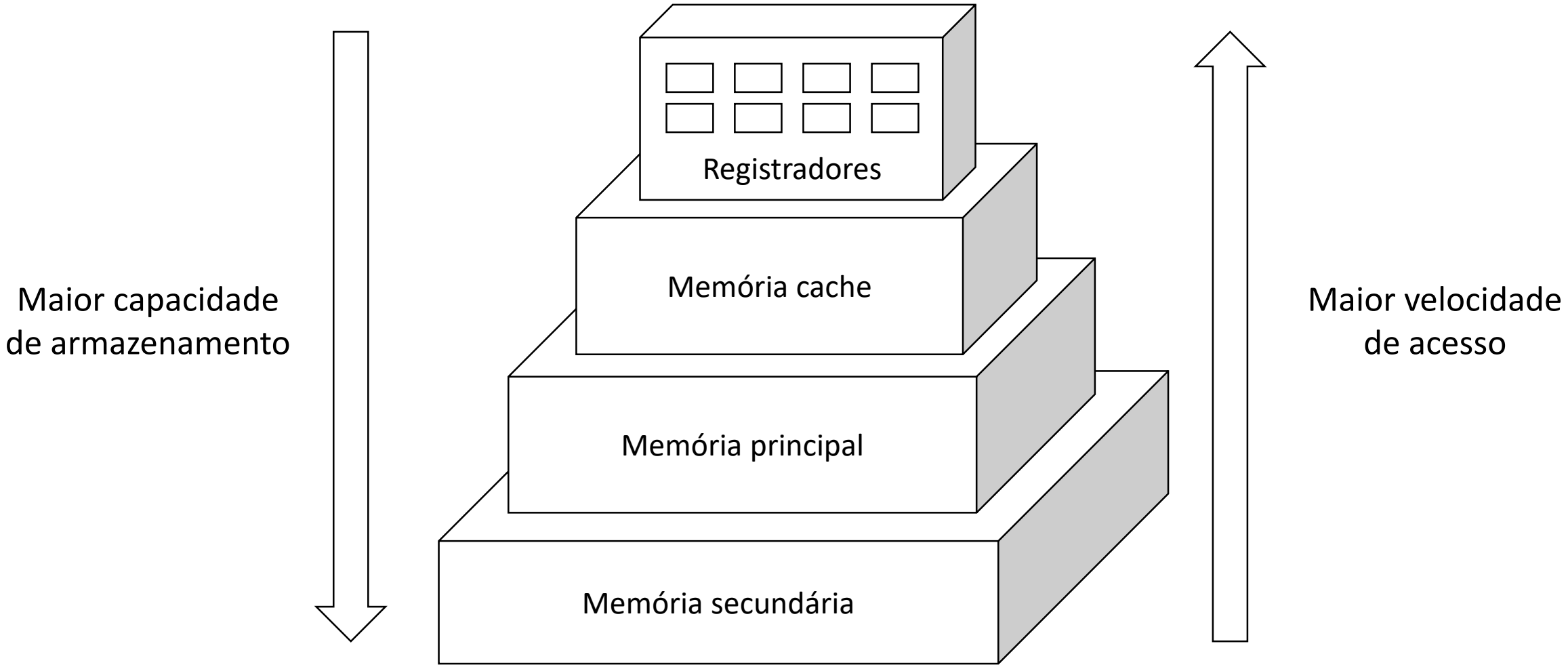
Memória principal

- Também chamada de RAM (*Random Access Memory*)
- É a unidade encarregada de armazenar os programas e dados que estão sendo processados
- Considerada um meio temporário de armazenamento de dados, pois os dados são mantidos somente durante o tempo em que o programa estiver em execução
- É na memória principal que o processador armazenados resultados de cada uma das operações que ele realiza

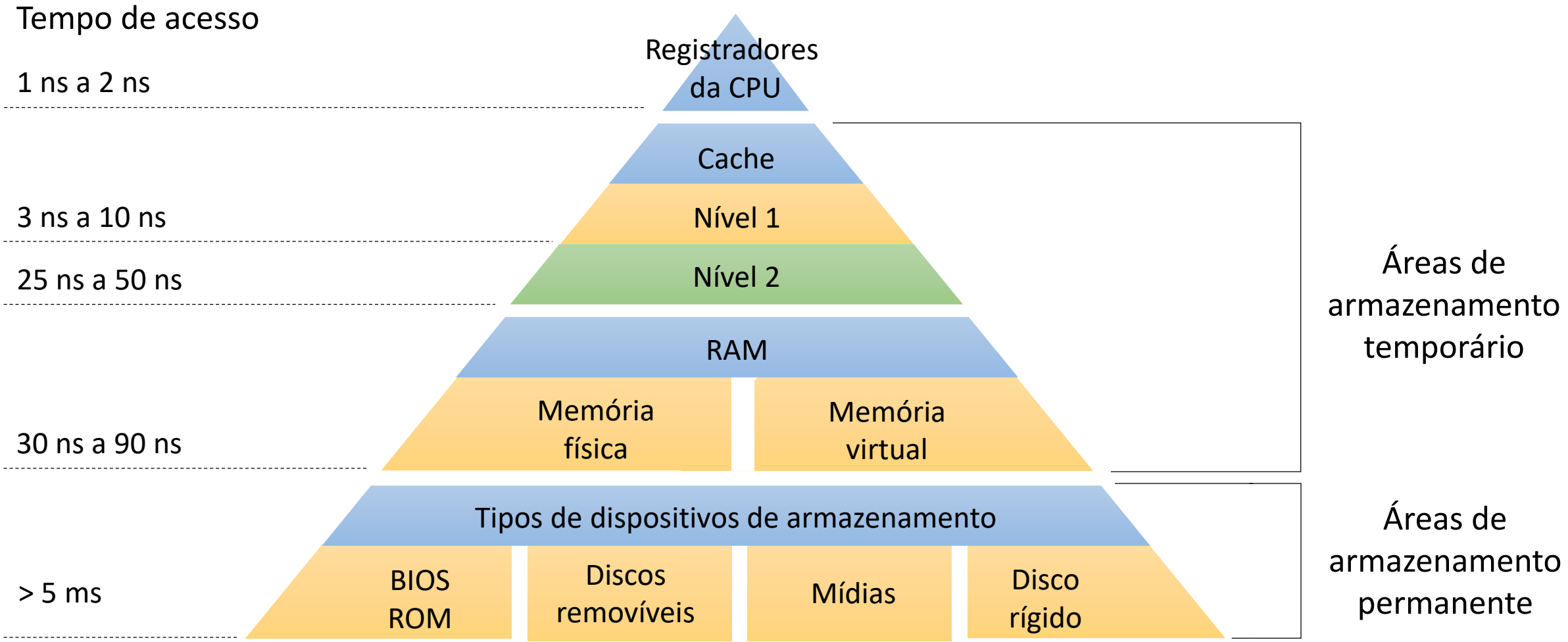
Memória secundária

- Pode ser composta por vários dispositivos capazes de armazenar grandes quantidades de dados e programas
- É um tipo de memória não-volátil, teoricamente permanente
  - Porém, em geral o acesso é mais lento
- Um programa armazenado em memória secundária precisa ser carregado na memória principal antes de ser executado

Hierarquia de memória



Hierarquia de memória



Dispositivos de entrada

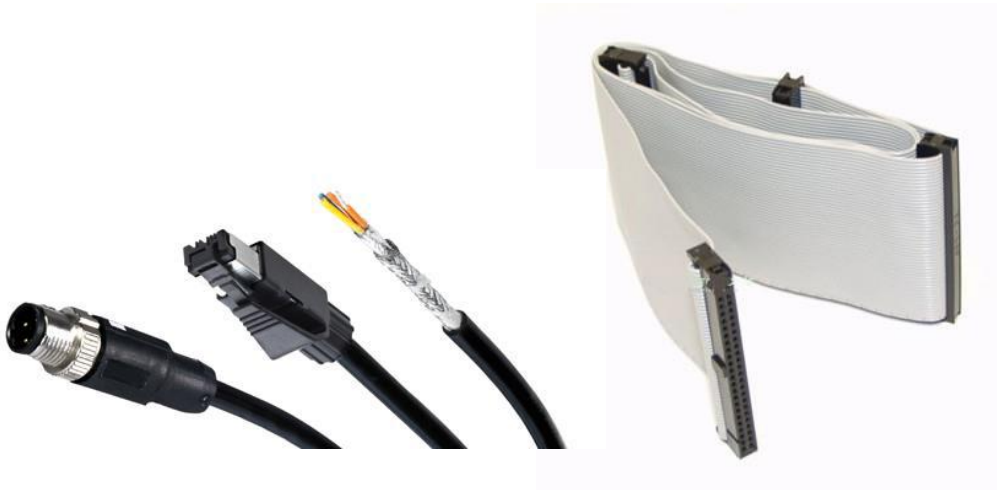
- São os recursos ou componentes que permitem que o usuário forneça dados para o computador
- Nesta disciplina, o dispositivo de entrada de dados utilizado nos programas desenvolvidos será o teclado

Dispositivos de saída

- São os recursos ou componentes que permitem que o computador forneça dados para o usuário
- Nesta disciplina, o dispositivo de saída de dados utilizado nos programas desenvolvidos será o monitor

Canal de comunicação

- A comunicação entre os dispositivos é realizada através de diferentes tipos de cabos e barramentos que, em geral, transmitem um sinal contínuo
- Dados são transmitidos utilizando apenas dois estados possíveis
  - Ausência ou presença do sinal a cada instante



Bits e bytes

- **BIT (BInary DigiT)** – Dígito binário
  - Menor unidade de informação que armazena somente um valor 0 ou 1

- **Byte (BinarY TErm)** – Termo binário
  - Conjunto de 8 bits, com o qual pode-se representar os números, as letras, os sinais de pontuação, etc.

- **Palavra (Word)**
  - É a quantidade de bits que a CPU processa por vez
  - Nos computadores atuais, são comuns palavras de 32 ou 64 bits

Bits e bytes

Unidades	Usual	Informática
Kilo (K)	10^3	2^{10} bytes
Mega (M)	10^6	2^{20} bytes
Giga (G)	10^9	2^{30} bytes
Tera (T)	10^{12}	2^{40} bytes

Qual a capacidade exata de bits que um pen-drive de 8GB possui?

$8GB = 8 * 2^{30} * 8 = 68719476736$ bits

Bits e bytes

Representação de uma memória de 1KB

Endereço	Byte							
0								
1								
2	0	1	0	0	0	0	0	1
...								
1023								

- No byte com endereço 2 está armazenado o código binário que representa o caractere **A**
- O processador acessa o conteúdo de um byte a partir do endereço deste byte

Bits e bytes

- Os computadores apenas operam com bytes e palavras

- Todo dado armazenado e processado é um conjunto de bits e bytes
  - Letras, dígitos e símbolos
  - Cores, ícones, figuras e fotos
  - Textos, músicas e vídeos

Tabela ASCII

padrões para letras, dígitos e símbolos

ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

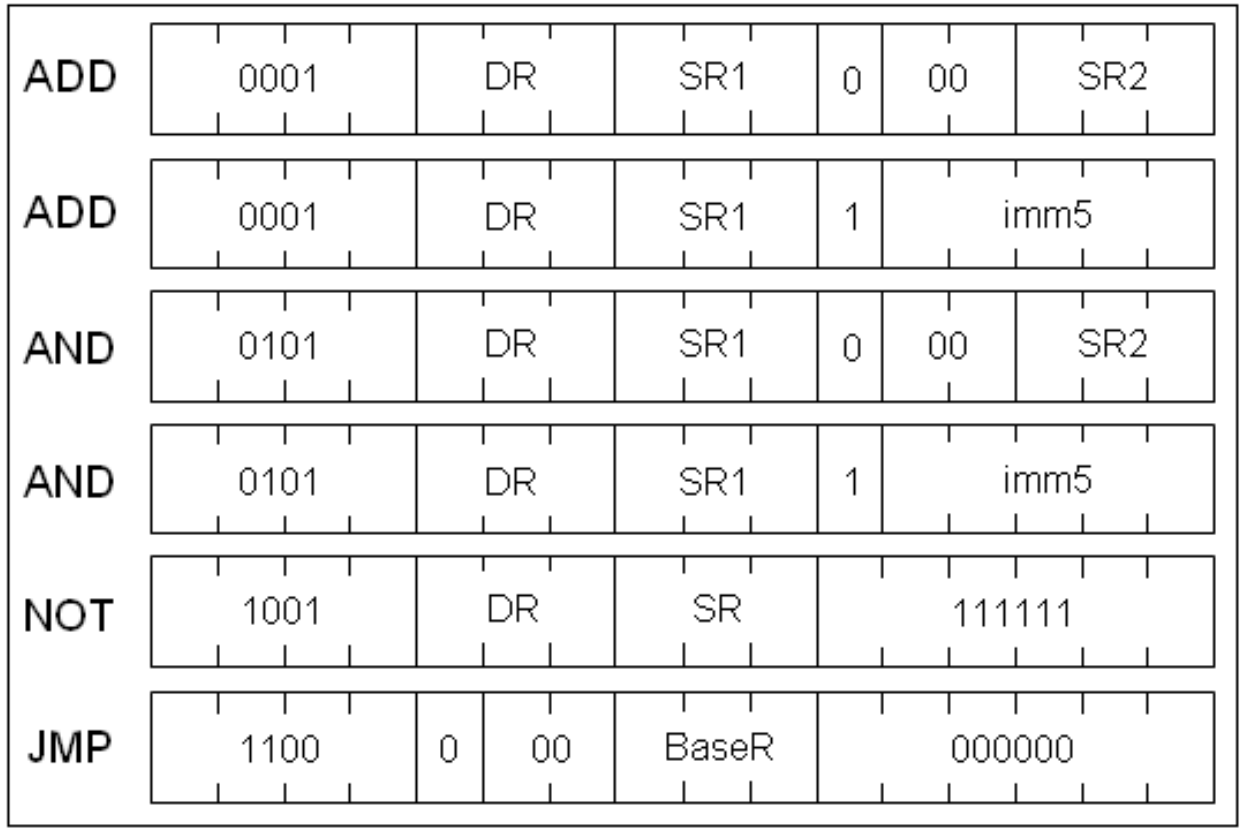
Bits, bytes e programas

- Computador é capaz de executar diferentes programas, desenvolvidos com finalidades distintas
  - Processador de texto
  - Planilha eletrônica
  - Calculadora
  - Navegador
  - Etc.

- Cada programa é executado por meio de um arquivo executável
  - Este arquivo executável contém as instruções que compõem o programa

Bits, bytes e programas

- Um arquivo executável é composto por milhares de instruções simples definidas através de sequências de 0 e 1
- A linguagem que representa a estrutura de um arquivo executável é denominada **linguagem de máquina**



Linguagens de máquina

- A l**inguagem de máquina** é a que o computador é capaz de entender e executar as instruções pré-definidas

- Por se tratar de uma sequência muito grande de 0 e 1, programar em linguagem de máquina é uma tarefa extremamente difícil

Instruções de máquina

- Operações básicas que podem ser executadas pelo hardware de forma eficiente:
  - Operações de processamento de dados
    - Operações matemáticas
    - Operações lógicas
  - Acessar variáveis ou estruturas de dados
    - Transferir dados da memória principal para o processador
    - Transferir bytes da entrada e saída para memória principal
  - Controlar a sequência (fluxo) do programa
    - Realizar desvios condicionais e incondicionais

Instruções de máquina

- Formato geral de instruções

- *Opcode* ou código de operação
  - Sequência de bits que identifica unicamente cada operação a ser realizada pelo processador

- Operandos
  - Podem ter 0, 1, 2 ou 3 campos de bits
  - Dependendo do *opcode*, determina onde estão os dados utilizados na operação
  - Registrador ou endereço de memória depende do modo de endereçamento

Linguagens de programação

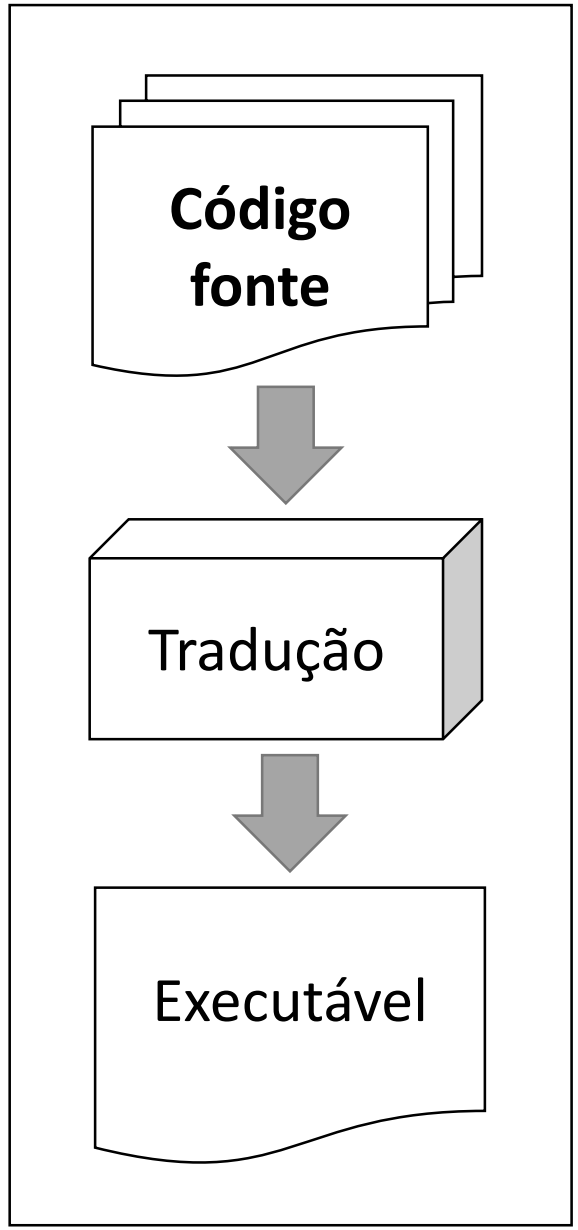
- Instruções em linguagens de programação (**linguagens de alto nível**) são escritas de forma muito mais clara e legível para o programador
  - Contudo, este tipo de linguagem a máquina não entende

- Linguagens de programação
  - É necessário traduzir o programa para a linguagem binária que o processador consiga entender
  - Em outras palavras, um programa executável

Tradutor

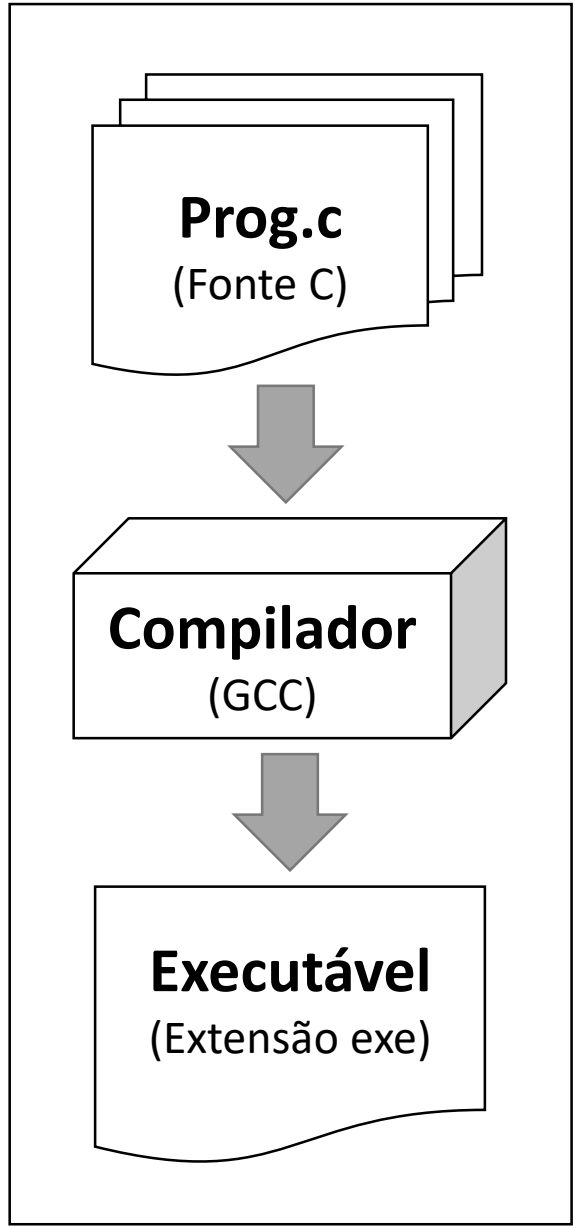
- O desenvolvedor escreve um programa em uma l**inguagem de programação**

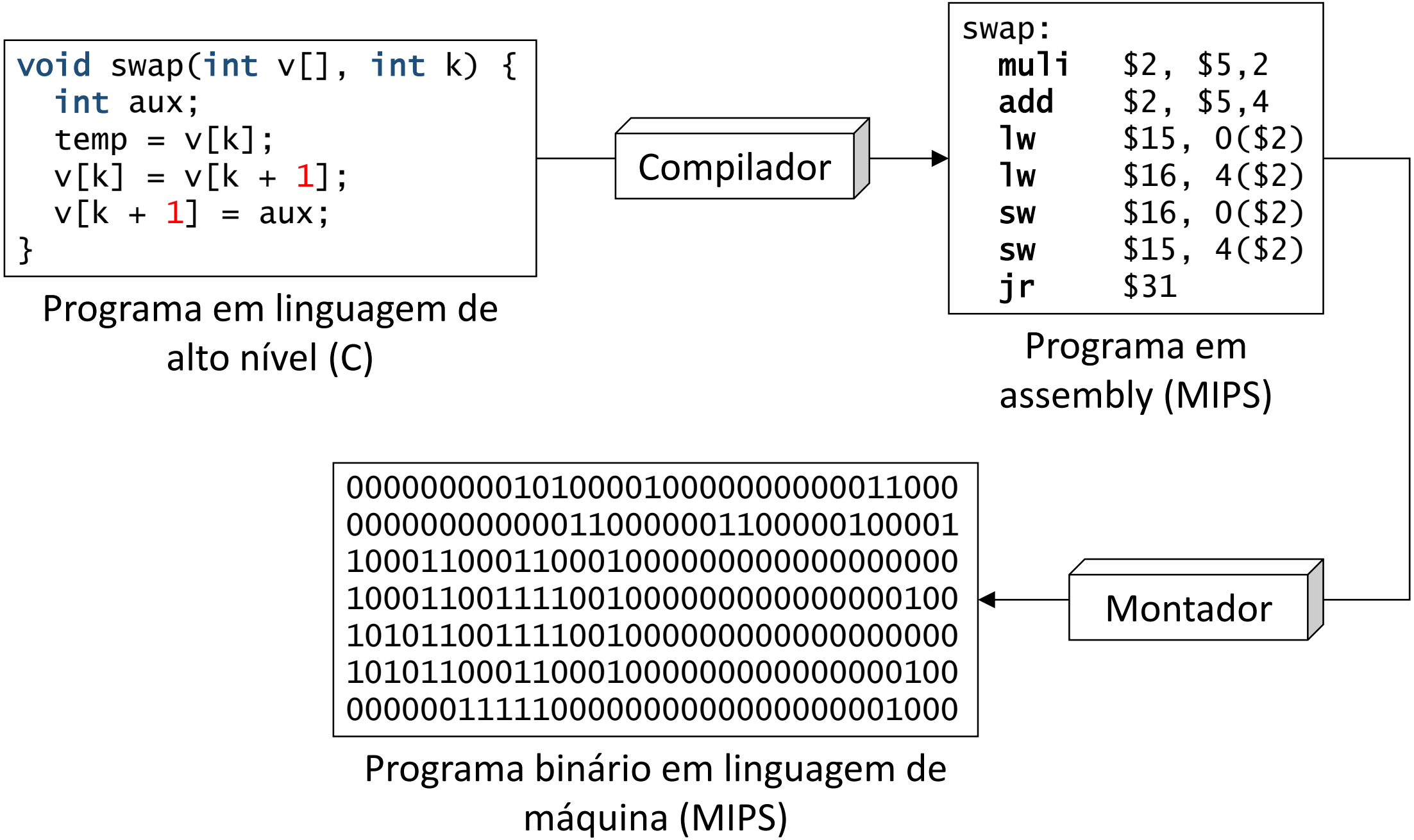
- Um programa específico é utilizado para traduzir as instruções definidas em linguagem de programação
  - Estes programas são chamados de compiladores
  - O resultado é um programa em **linguagem de montagem**



Tradutor

- Nesta disciplina
  - Linguagem de programação: C
  - Código fonte: arquivos com extensão .c
  - Tradutor: compilador C (GCC)
    - Traduzir → compilar
- Para que o processo de tradução seja possível, é necessário que o compilador consiga identificar cada instrução no código fonte
  - Assim, o programador precisa seguir uma séries de regras ao utilizar uma linguagem de programação

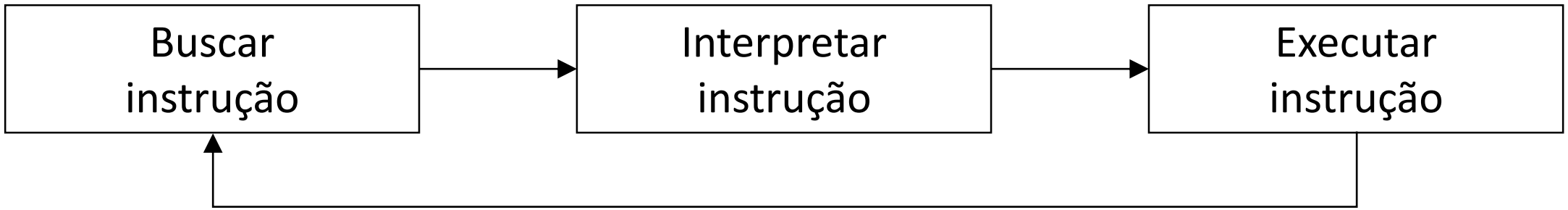



void swap(int v[], int k) {
int aux;
temp = v[k];
v[k] = v[k + 1];
v[k + 1] = aux;
}
Programa em linguagem de alto nível (C)
Compilador
swap:
muli $2, $5,2
add $2, $5,4
lw $15, 0($2)
lw $16, 4($2)
sw $16, 0($2)
sw $15, 4($2)
jr $31
Programa em assembly (MIPS)
Montador
00000000010100001000000000011000
00000000000011000000110000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
Programa binário em linguagem de máquina (MIPS)

Execução de programas

- O processador é um dispositivo que opera em ciclos
  - Cada ciclo executa uma instrução que corresponde a pequenas tarefas executadas sobre operandos

- Representação de um ciclo de operação:



Análise léxica, sintática e semântica

- Análise léxica
  - Decompõe o programa fonte em seus elementos individuais distintos
    - Comandos, operadores, variáveis
  - Verifica se estes elementos estão de acordo com as regras da linguagem

- Análise sintática
  - Conjunto de regras que definem como uma linguagem pode ser utilizada
  - O desenvolvedor precisa seguir estas regras ao escrever as instruções dos programas para que o compilador possa interpretá-las e traduzi-las

- Análise semântica
  - Cada instrução tem uma finalidade bem específica
  - A combinação de instruções tem um significado lógico
    - Uma sequência de instruções colabora de forma parcial para os objetivos do programa sejam alcançados durante a execução

Análise léxica, sintática e semântica

- Análise sintática
  - Se o desenvolvedor comete um erro de sintaxe, o compilador interrompe o processo de tradução e indica a linha do arquivo onde o erro ocorreu
    - Nestes casos o arquivo executável não é gerado
  - Exemplos de erros de sintaxe
    - Palavras com erros de grafia
    - Parênteses ou aspas que não fecham
    - Ausência de vírgulas, pontos ou ponto e vírgula

Análise léxica, sintática e semântica

- Análise semântica
  - Se o programa gerado não cumpre totalmente seus objetivos, o programa provavelmente contém erros de lógica
  - É comum um programa funcionar corretamente na maioria das vezes e apresentar um erro de lógica só em condições muito específicas
  - Erros de lógica não são apontados pelo compilador
    - Encontrá-los é uma tarefa do desenvolvedor
    - Estes erros costumam ser difíceis de encontrar

Análise léxica, sintática e semântica

Qual o erro de lógica na sequência de instruções do programa abaixo?

1. Solicite ao usuário do programa que digite o número de pontos do **time 1** (este valor será chamado de **pt1**)
2. Solicite ao usuário do programa que digite o número de pontos do **time 2** (este valor será chamado de **pt2**)

SE **pt1** > **pt2**:

Exibe “Time 1 venceu”

SENÃO

Exibe “Time 2 venceu”

Análise léxica, sintática e semântica

- Para o desenvolvimento de programas, estes dois aspectos estão sempre presentes

- Sintaxe
  - É necessário conhecer a linguagem de programação e suas regras para ser possível construir programas

- Semântica
  - É necessário conhecer a finalidade de cada instrução da linguagem e, principalmente, é necessário saber combinar estas finalidades isoladas para alcançar o objetivo do programa

Montagem

- O montador realiza a tradução de um programa em linguagem de montagem (*assembly*) para linguagem binária
  - Realiza a análise léxica, sintática e semântica do código fonte
  - Substitui códigos de operações simbólicos por valores numéricos
  - Substituir nomes simbólicos de endereços por valores numéricos
  - Converter valores de constantes para binários
  - Reservar espaço de memória para armazenamento de instruções e dados

Modelo de Von Neumann

- Ciclo de instrução
  - Busca a próxima instrução
  - Decodifica (identifica) a instrução
  - Executa a instrução
    - Executa uma sequência de suboperações equivalente do que foi solicitado
    - Exemplo: buscar os dados, realizar a função solicitada ou guardar o resultado
  - Repete o ciclo

- Instruções são interpretadas

Paradigmas de programação

- No desenvolvimento de um programa, o desenvolvedor utiliza um modo de raciocínio pouco comum em outras áreas do conhecimento

Encontrar uma estruturação que contemple instruções para resolver um determinado problema

Um paradigma de programação fornece e determina um raciocínio sobre a estruturação e execução de um programa

Algoritmos

- Algoritmo
  - Sequência de instruções que resolve um determinado problema

- Programa
  - Algoritmo escrito em uma linguagem de programação específica
    - Um algoritmo que pode ser executado em um computador

- Paradigma de programação
  - Raciocínio utilizado para criar um algoritmo

Algoritmos

- Detalhes sobre algoritmos
  - Sequência de instruções bem definidas
  - Pode receber ou gerar informações
    - Dados de entrada e saída
  - Tem um início e fim bem definidos
    - Processo finito, sempre termina
  - Cumpre um propósito específico

- Na maioria das vezes, elaborar o algoritmo para resolver um problema é o maior desafio na programação
  - Embora algoritmos já façam parte do dia-a-dia das pessoas, elaborar algoritmos de forma sistemática é uma atividade muito pouco exercitada

Algoritmos

Treino de corrida para iniciantes

1. Caminhe por 5 minutos em ritmo lento
2. Repita 5 vezes a seguinte sequência
   - Corra por 30 segundos em um ritmo em que você respire com dificuldade
   - Caminhe por 60 segundos
3. No final, caminhe por 5 minutos em ritmo lento

Pontos importantes

- A ordem das instruções é significativa
- Dificilmente existe um único algoritmo para resolver um problema

Algoritmos

Vamos elaborar um algoritmo para trocar uma lâmpada em um quarto vazio, assumindo que:

- Existe uma escada disponível
- Existe uma lâmpada nova disponível
- Está de dia

Algoritmos

Um senhor está em uma das margens de um rio com

- Uma raposa
- Um saco de milho
- Uma dúzia de galinhas

O senhor precisa atravessar e dispõe de uma canoa que suporta seu peso e uma de suas cargas

- O senhor não pode deixar a raposa sozinha com as galinhas em uma das margens
- Também não pode deixar as galinhas sozinhas com o milho

Que instruções podemos definir para o senhor atravessar o rio?

Algoritmos

Travessia do rio

1. Atravesse com as galinhas
   (se levasse a raposa, as galinhas ficariam com o milhos e se levasse o milho, a raposa ficaria com as galinhas)
2. Deixe as galinhas e retorne sozinho
3. Atravesse com a raposa
4. Deixe a raposa e retorne com as galinhas
   (as galinhas não poderiam ficar com a raposa)
5. Deixe as galinhas e atravesse com o milho
   (não adiantaria voltar com as galinhas)
6. Deixe o milho e retorne sozinho
7. Atravesse com as galinhas

Algoritmos

- Ao longo da disciplina, vamos resolver
  - Problemas matemáticos
    - Média aritmética
    - Raízes de uma equação de segundo grau
    - Operação em matrizes
  - Questões genéticas de leitura, armazenamento ou processamento de dados
    - Encontrar dados de uma pessoa
    - Atualizar o saldo de uma conta bancária
  - Puzzles
    - Resolver problemas lógicos
    - Encontrar uma sequência de ações

Formas de representação

- Formas de representação de algoritmos mais conhecidas
  - Descrição narrativa
  - Fluxograma tradicional
  - Diagrama de Chapin
  - Pseudolinguagem

Descrição narrativa

- Nesta forma de representação, os algoritmos são expressos diretamente em **linguagem natural**
  - Deve ser usada com cuidado pois o uso da linguagem natural pode dar margem a má interpretação, ambiguidades ou imprecisões

Troca de um pneu furado

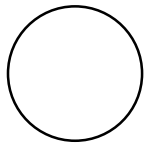
1. Afrouxe ligeiramente as porcas das rodas
2. Suspenda o carro
3. Retire as porcas e o pneu
4. Coloque o pneu reserva
5. Aperte ligeiramente as porcas
6. Abaixe o carro
7. Aperte completamente as porcas

Fluxograma

- Representação gráfica de algoritmos, onde cada forma geométrica representa uma instrução em específico
- Objetiva facilitar o entendimento das ideias e fluxos de dados contidas no algoritmo

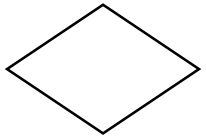
Início e
fim de fluxo

Operações de
atribuição



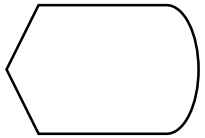
Particionamento
do diagrama

Operação de
entrada de dados



Decisão

Operação de
saída de dados
(impressora)



Operação de
saída de dados (vídeo)

Fluxo de dados

Fluxograma

Fluxograma para calcular a média de um aluno

- Desenhar o fluxograma pode ser uma tarefa difícil dependendo da complexidade do algoritmo
- Modificar ou corrigir o algoritmo, depois de desenhado o fluxograma, pode ser trabalhoso também

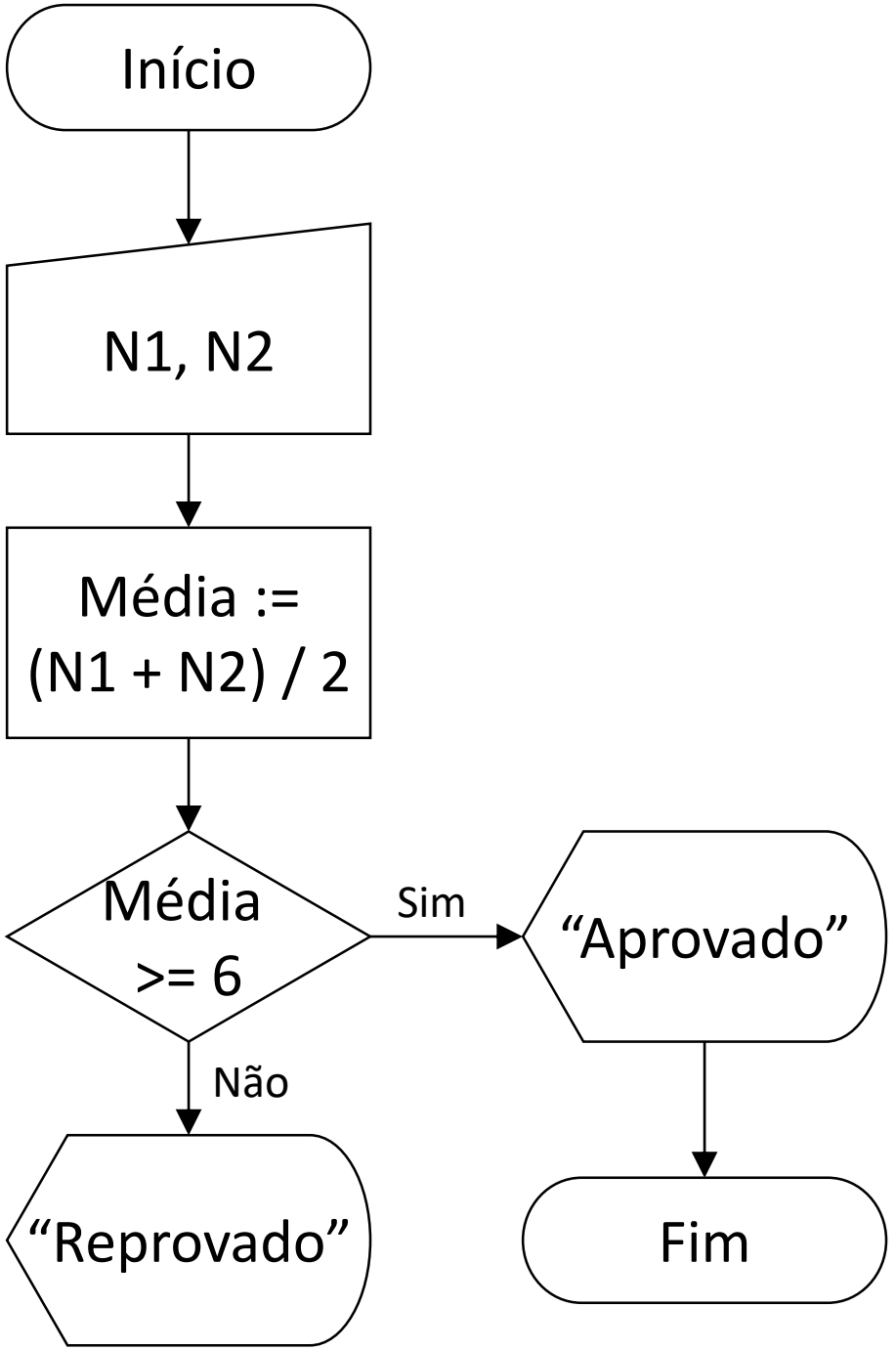
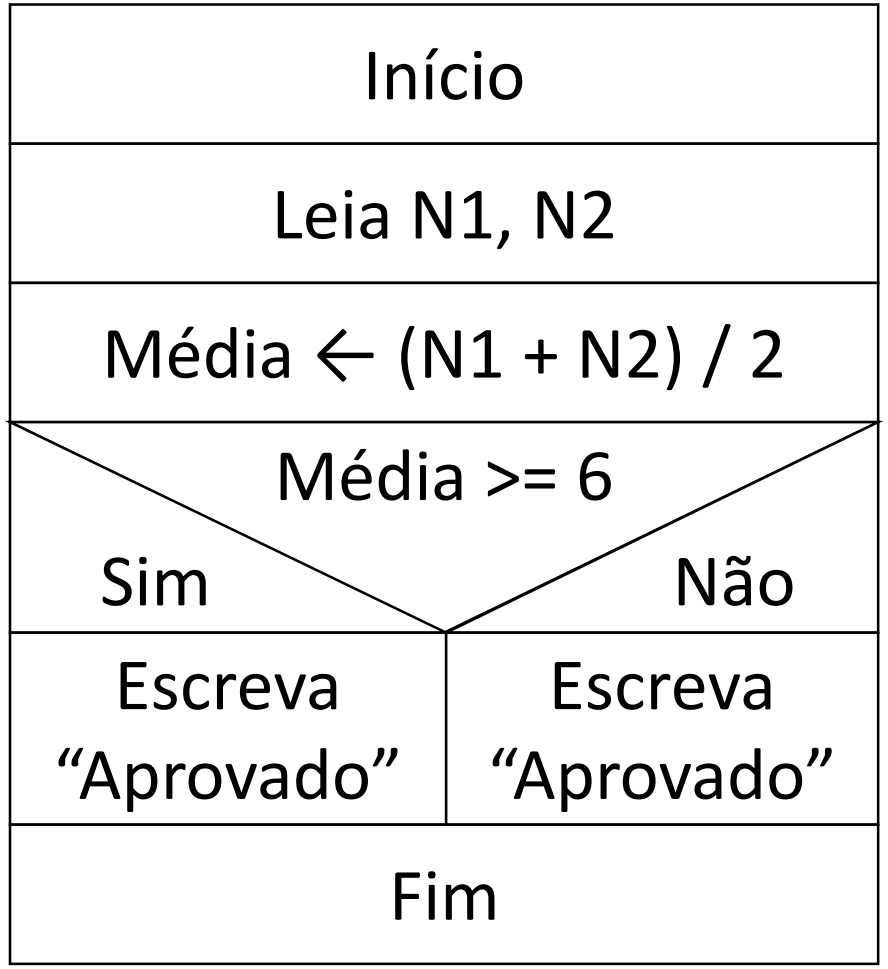


Diagrama de Chapin

- Variação do fluxograma tradicional que permite uma visão hierárquica e estruturada da lógica do programa
  - Criado por Ned Chapin

- Possui os mesmos pontos de atenção de um fluxograma tradicional
  - Representar o diagrama também pode ser difícil, especialmente quando há muitas instruções alinhadas



Pseudolinguagem

- Representação de algoritmos, também conhecida como **pseudocódigo**, **português estruturado** ou **portugol**
  - Rica em detalhes e assemelha-se à forma que os programas são escritos

- A tradução do pseudocódigo de um algoritmo para uma linguagem de programação é praticamente direta e intuitiva

Pseudolinguagem

Pseudocódigo para calculo da média de um aluno

```
principal {
    real n1, n2, media;
    leia(n1, n2);
    media ← (n1 + n2) / 2;
    se (media >= 6) {
        imprima("Aprovado");
    } senão {
        imprima("Reprovado");
    }
}
```

Exercícios

- Utilizando um jarro de 5 litros, um jarro de 3 litros e uma fonte de água, elabore um algoritmo para obter exatamente 4 litros d'água
  - Os jarros não possuem marcação de capacidade

- Você possui 8 esferas de tamanho idêntico, sendo que apenas uma delas tem peso superior às outras
  - Utilizando uma balança de dois pratos, faça um algoritmo para encontrar a esfera com peso diferente sabendo que a balança só pode ser usada duas vezes

AGT – Algoritmos

Introdução e conceitos básicos

Prof. Allan Rodrigo Leite