

# Sistemas de Computação 16/2

notação de testes  $\rightarrow 8.5$   
(907.)

TPC (10%)

recursos da função

29 / março 1 / julho  
1  $\rightarrow$  10 valores  
10 valores

$\rightarrow$  Níveis de um computador:

$\downarrow$   
só é considerado co-putador se / distinção do que não é um  
 $\rightarrow$  data-dependent instruction sequence co-putador:  
 $\rightarrow$  data-independent data selection.

$\rightarrow$  depende do dado a tempo de execução.

$\rightarrow$  dados independentes

if (x == 1)

...

else

...

depende dos dados.

x = 3  $\Rightarrow$  depende dos dados.  
 $\rightarrow$  resolve o "else".

depende de auto.

dados. (ex: quando o

valor 7, ao escrever

7, vai depender do

dado inicializado "7".)

$\rightarrow$  é uma calculadora se não tiver 1 ou nenhum destes dois pios.

$\rightarrow$  programa: sequência de instruções que descrevem a tarefa a ser desempenhada.

ex: projeto: computadores que foi programado para uma tarefa específica para gerar um

programa em Haskell e transformado em ferramenta para gerar o código gerado.

configurar piladas

o PC não

programa o código logo.



→ transformar o programa numa linguagem em formato que o PC entende.

→ máquina que opera e executa apps

level 6

User

problem-oriented language → program  
translation (compiler)

assembly language → parecido com o  
translation (assembler)

operating system machine → peça de software que a máquina  
usa para gerir a memória etc...

instruction set architecture (ISA)

direct execution or interpretation (microprogram)

micro architecture

hardware

digital logic

o que  
se vai  
passar  
+ detalhes

0/1  
ou  
on/off

utilizam

cadeias de código binário

001 → cadeia de bits  
(cadeia de bits)

(cadeia instrução)

2<sup>1</sup> bit)

0

→ cadeia de

bits que o problema executa

Fronteira dos níveis 4 e 3.

O porque do código binário? A eletrificação flui por elétrons; e + fácil  
distinguir 2 níveis (0 e 1) 2 níveis.

↳ pensar em porta (ou seja aberta/fechada ⇒ não é meia aberta...)