

# Gabarito Nota 10

UVV, \_\_\_\_\_

Student

Quiz

Date/Notes

| Student ID |   |   |   |   | A B C D E |   |   |   |   | A B C D E |    |   |   |   | A B C D E |   |    |   |   | A B C D E |   |   |    |   |   |   |   |   |
|------------|---|---|---|---|-----------|---|---|---|---|-----------|----|---|---|---|-----------|---|----|---|---|-----------|---|---|----|---|---|---|---|---|
| 0          | 1 | 0 | 0 | 0 | 1         | ○ | ○ | ○ | ○ | ●         | 16 | ○ | ○ | ○ | ○         | ● | 31 | ● | ○ | ○         | ○ | ○ | 46 | ○ | ○ | ○ | ○ | ○ |
| 1          | ○ | ○ | ○ | ○ | 2         | ○ | ○ | ○ | ○ | ○         | 17 | ○ | ○ | ○ | ○         | ○ | 32 | ○ | ○ | ○         | ○ | ○ | 47 | ○ | ○ | ○ | ○ | ○ |
| 2          | ○ | ○ | ○ | ○ | 3         | ○ | ○ | ○ | ○ | ○         | 18 | ○ | ○ | ○ | ○         | ○ | 33 | ○ | ○ | ○         | ○ | ○ | 48 | ○ | ○ | ○ | ○ | ○ |
| 3          | ○ | ○ | ○ | ○ | 4         | ○ | ○ | ○ | ○ | ○         | 19 | ○ | ○ | ○ | ○         | ○ | 34 | ○ | ○ | ○         | ○ | ○ | 49 | ○ | ○ | ○ | ○ | ○ |
| 4          | ○ | ○ | ○ | ○ | 5         | ○ | ○ | ○ | ○ | ○         | 20 | ○ | ○ | ○ | ○         | ○ | 35 | ○ | ○ | ○         | ○ | ○ | 50 | ○ | ○ | ○ | ○ | ○ |
| 5          | ○ | ○ | ○ | ○ | 6         | ○ | ○ | ○ | ○ | ○         | 21 | ○ | ○ | ○ | ○         | ○ | 36 | ○ | ○ | ○         | ○ | ○ | 51 | ○ | ○ | ○ | ○ | ○ |
| 6          | ○ | ○ | ○ | ○ | 7         | ○ | ○ | ○ | ○ | ○         | 22 | ○ | ○ | ○ | ○         | ○ | 37 | ○ | ○ | ○         | ○ | ○ | 52 | ○ | ○ | ○ | ○ | ○ |
| 7          | ○ | ○ | ○ | ○ | 8         | ○ | ○ | ○ | ○ | ○         | 23 | ○ | ○ | ○ | ○         | ○ | 38 | ○ | ○ | ○         | ○ | ○ | 53 | ○ | ○ | ○ | ○ | ○ |
| 8          | ○ | ○ | ○ | ○ | 9         | ○ | ○ | ○ | ○ | ○         | 24 | ○ | ○ | ○ | ○         | ○ | 39 | ○ | ○ | ○         | ○ | ○ | 54 | ○ | ○ | ○ | ○ | ○ |
| 9          | ○ | ○ | ○ | ○ | 10        | ○ | ○ | ○ | ○ | ○         | 25 | ○ | ○ | ○ | ○         | ○ | 40 | ○ | ○ | ○         | ○ | ○ | 55 | ○ | ○ | ○ | ○ | ○ |
|            |   |   |   |   | 11        | ○ | ○ | ○ | ○ | ○         | 26 | ○ | ○ | ○ | ○         | ○ | 41 | ○ | ○ | ○         | ○ | ○ | 56 | ○ | ○ | ○ | ○ | ○ |
|            |   |   |   |   | 12        | ○ | ○ | ○ | ○ | ○         | 27 | ○ | ○ | ○ | ○         | ○ | 42 | ○ | ○ | ○         | ○ | ○ | 57 | ○ | ○ | ○ | ○ | ○ |
|            |   |   |   |   | 13        | ○ | ○ | ○ | ○ | ○         | 28 | ○ | ○ | ○ | ○         | ○ | 43 | ○ | ○ | ○         | ○ | ○ | 58 | ○ | ○ | ○ | ○ | ○ |
|            |   |   |   |   | 14        | ○ | ○ | ○ | ○ | ○         | 29 | ○ | ○ | ○ | ○         | ○ | 44 | ○ | ○ | ○         | ○ | ○ | 59 | ○ | ○ | ○ | ○ | ○ |
|            |   |   |   |   | 15        | ○ | ○ | ○ | ○ | ○         | 30 | ○ | ○ | ○ | ○         | ○ | 45 | ○ | ○ | ○         | ○ | ○ | 60 | ○ | ○ | ○ | ○ | ○ |



QuickKey

Free blank answer forms at  
www.quickkeyapp.com



10,0

O → gabarito

## 1 Introdução

1. Os computadores levaram a uma terceira revolução na história da civilização, a **revolução da informação**, que se juntou às revoluções agrícola e industrial. Podem ser considerados frutos dessa revolução:
  - A. O uso de computadores em automóveis
  - B. Telefone celulares e *smartphones*
  - C. Projeto Genoma Humano
  - D. *World Wide Web* e ferramentas de busca
  - **(E)** Todas as respostas acima
2. Em relação às classes tradicionais de aplicação de computadores, avalie as seguintes afirmações:
  - I. Os **computadores pessoais** (*Personal Computer* — PC) enfatizam a entrega de boa performance para usuários individuais, em custo relativamente baixo, usualmente executando *softwares* aplicativos de produtividade e entretenimento.
  - II. Os **servidores** enfatizam a realização de grandes cargas de trabalho (por exemplo, uma única aplicação científica ou de engenharia complexa, ou diversos pequenos *jobs* em um servidor de internet), com capacidade de *input*, *output*, processamento e armazenamento iguais ou equivalentes às de um computador pessoal.
  - III. Os **servidores** abrangem a maior variedade de custo e capacidade. Na extremidade inferior podem ser pouco mais do que um *desktop* sem monitor e teclado, para trabalhos que exigem pouca carga (por exemplo, um servidor de arquivos de um departamento empresarial); na extremidade superior podem ser supercomputadores voltados para tarefas computacionais extremamente complexas e difíceis.

Marque a alternativa correta:

- ~~A. Apenas a afirmação I é verdadeira~~
  - ~~B. Apenas a afirmação II é verdadeira~~
  - C. As afirmações I e II são verdadeiras
  - (D)** As afirmações I e III são verdadeiras
  - ~~E. As afirmações II e III são verdadeiras~~
3. Em relação aos **computadores embarcados**, assinale a alternativa incorreta:
    - ~~A. Constituem a maior classe de computadores e abrangem a maior gama de aplicações e desempenho~~
    - **(B)** Como possuem enorme variabilidade, não podem ser utilizados para dispositivos de **Internet das Coisas** (*Internet of Things* — IoT)
    - ~~C. São projetados para executar um aplicativo ou conjunto de aplicativos relacionados que normalmente são integrados ao *hardware* e entregues como um sistema único~~
    - D. Estão presentes nas chamadas *smart TV*, televisões “inteligentes” com capacidade de acesso à internet
    - ~~E. Apesar do grande número de computadores embarcados, a maioria dos usuários nem percebe que está usando um computador~~

4. Em relação aos **computadores embarcados**, avalie as seguintes afirmações:

- I. Computadores embarcados são projetados para rodar aplicações que necessitam de grande performance e poder de processamento, mas com limitações no custo ou consumo de energia **F**
- II. Apesar do baixo custo, os computadores embarcados podem ter baixa tolerância à falhas, ou seja, podem falhar com frequência sem causar grandes prejuízos aos consumidores **F**
- III. A confiabilidade de um sistema com computador embarcado pode ser obtida através da simplicidade (fazer uma única coisa bem feita) ou de redundância. **V**

Marque a alternativa correta:

- ☒ A. Existem duas afirmações falsas
  - B. Existem duas afirmações verdadeiras
  - C. Apenas a afirmação I é verdadeira
  - D. Apenas a afirmação III é falsa
  - E. Todas as afirmações são falsas
5. Quando queremos indicar diferentes quantidade de *bits* ou de *bytes* (conjunto de 8 *bits*), podemos nos expressar usando o sistema decimal ou o sistema binário, conforme ilustrado na tabela abaixo (note que em cada coluna da tabela está faltando uma informação, indicado por "?"):

| Termo Decimal | Sigla | Valor     | Termo Binário | Sigla | Valor     |
|---------------|-------|-----------|---------------|-------|-----------|
| kilobyte      | KB    | $10^3$    | kibibyte      | KiB   | $2^{10}$  |
| megabyte      | MB    | $10^6$    | mebibyte      | MiB   | $2^{20}$  |
| gibabyte      | GB    | $10^9$    | gibibyte      | GiB   | $2^{30}$  |
| terabyte      | TB    | $10^{12}$ | tebibyte      | TiB   | $2^{40}$  |
| petabyte      | PB    | $10^{15}$ | pebibyte      | PiB   | $2^{50}$  |
| ?             | EB    | $10^{18}$ | exbibyte      | EiB   | $2^{60}$  |
| zettabyte     | ZB    | $10^{21}$ | zebibyte      | ZiB   | $2^{70}$  |
| yottabyte     | YB    | $10^{24}$ | yobibyte      | YiB   | $2^{80}$  |
| ronnabyte     | RB    | $10^{27}$ | robibyte      | RiB   | $2^{90}$  |
| queccabyte    | ?     | $10^{30}$ | quebibyte     | QiB   | $2^{100}$ |

Completam a tabela acima, na ordem das colunas da esquerda para a direita, os termos:

- ~~A. exabyte — BQ —  $10^9$  — yottabibyte — MiB —  $2^{60}$~~
  - ~~B. exabyte — QB —  $10^{10}$  — yobibyte — QiB —  $2^{66}$~~
  - ☒ C. exabyte — QB —  $10^9$  — yobibyte — MiB —  $2^{60}$
  - ~~D. exibyte — QB —  $10^9$  — yottabyte — MiB —  $2^{60}$~~
  - ~~E. exabibyte — BQ —  $10^{90}$  — yottabibyte — MB —  $2^6$~~
6. Um provedor de internet anuncia uma conexão de “50 megas” de velocidade. O que isso quer dizer? (considere “B” como byte, e “b” como bit)
- A. 50 MB por segundo
  - B. 50 Mib por segundo
  - C. 50 KiB por segundo
  - ☒ D. 50 Mb por segundo
  - E. 50 Kb por segundo

7. Com a evolução da computação, atualmente, estamos saindo da "era do PC" e entrando na "era pós-PC", da qual são representantes importantes os **dispositivos móveis pessoais**, a **computação em nuvem** e o uso de **software como serviço**. Esta afirmação é:

A. Falsa ☒ B. Verdadeira C. Impossível saber

8. A performance de um programa depende da combinação da efetividade dos algoritmos utilizados no programa, dos sistemas de software utilizados para criar e traduzir o programa em instruções de máquina, e da efetividade do computador em executar essas instruções, que pode incluir operações de *input/output* (I/O). Considere os seguintes componentes de *hardware* ou *software*:

A. Algoritmo

~~B. Linguagem de programação, compilador e arquitetura~~

C. Processador e sistema de memória

D. Sistema de I/O (*hardware* e sistema operacional)

Considere como os componentes acima afetam a performance dos computadores e faça uma correspondência dos itens A, B, C e D com as afirmações a seguir:

~~(B)~~ Determina o número de instruções para cada sentença do código fonte.

A ☒ (D) Determina o quão rápido as operações de I/O podem ser executadas

D ( ) Determina o número de sentenças do código fonte e o número de operações de I/O executadas

~~(C)~~ Determina o quão rápido as instruções podem ser executadas

Assinale a alternativa que contém, de cima para baixo, a ordem correta das letras A, B, C e D acima:

☒ A. Ordem: B A D C

~~B. Ordem: D C A B~~

C. Ordem: B C D A

☒ D. Ordem: B D A C

~~E. Nenhuma das alternativas anteriores~~

## 2 As 7 grandes idéias na arquitetura de computadores

9. Uma das técnicas mais importantes na arquitetura dos computadores é a habilidade de esconder os detalhes de baixo nível e oferecer um modelo mais simples em níveis mais altos. Essa idéia é conhecida por:

A. Tornar o caso comum rápido

B. Paralelismo

C. *Pipelining*

☒ D. Abstração

E. Predição



10. Os computadores não precisam ser apenas rápidos, eles precisam ser confiáveis. Já que qualquer dispositivo físico falha, que idéia garante a confiabilidade?
- A. Abstração
  - ☒ B. Redundância
  - C. Hierarquia de memórias
  - D. Paralelismo
  - E. Predição
11. Idéia que nos permitiu obter maior performance através da realização de operações simultâneas:
- A. Abstração
  - ☒ B. Paralelismo
  - C. Hierarquia de memórias
  - D. Tornar o caso comum rápido
  - E. Predição
12. Considere um processador. Pode ser mais rápido, em média, fazer com que o processador “chute” o que ele precisa fazer e comece a trabalhar do que esperar e ter certeza dos próximos passos a serem executados (desde que, é óbvio, os mecanismos para a recuperação de um chute errado não sejam muito custosos e de que o chute inicial seja razoavelmente correto). Essa idéia é a:
- A. Abstração
  - ☒ B. Predição
  - C. Hierarquia de memórias
  - D. Tornar o caso comum rápido
  - E. *Pipelining*
13. Os programas são executados através de processos na memória do computador e, idealmente, gostaríamos de ter uma quantidade enorme de memória, sempre com a maior velocidade possível. Entretanto, quanto mais rápida mais cara uma memória é, e foi preciso alcançar um balanço entre tamanho e custo. A idéia desse balanceamento entre velocidade, custo e tamanho é a:
- A. Abstração
  - B. Predição
  - ☒ C. Hierarquia de memórias
  - D. Tornar o caso comum rápido
  - E. *Pipelining*
14. Ao tentarmos melhorar a performance de um processador geralmente devemos nos focar nos casos mais simples e não nos mais complexos e poucos comuns. Isso é conhecido por:
- A. Paralelismo
  - B. Predição
  - C. Hierarquia de memórias
  - ☒ D. Tornar o caso comum rápido
  - E. Redundância

15. Um tipo especial de paralelismo, já grandemente utilizado nos processadores de hoje em dia ao nível de instruções é:
- A. Abstração
  - B. Predição
  - C. Hierarquia de memórias
  - D. Tornar o caso comum rápido
  - ☒ E. *Pipelining*

### 3 Abaixo de seu programa

16. A figura a seguir ilustra algumas camadas de *software* e o *hardware* que dá suporte à execução dos programas:



Os *softwares* que utilizamos no dia a dia são chamados de **aplicações** e formam a camada mais externa na figura. Entre essa camada externa e o *hardware* existe uma camada de *softwares* de sistemas. Existem vários tipos de *softwares* de sistema, mas os dois tipos que são centrais aos computadores de hoje em dia são:

- A. Sistema operacional e *assembly*
  - B. Compiladores e sistemas embarcados
  - C. *Assembly* e compiladores
  - D. Sistema operacional e sistemas embarcados
  - ☒ E. Sistema operacional e compiladores
17. Dentre as funções de um sistema operacional incluem-se, exceto:
- A. Lidar com operações de entrada
  - B. Lidar com operações de saída
  - ☒ C. Compilar código fonte de programas
  - D. Alocar armazenamento e memória
  - E. Compartilhamento seguro dos recursos computacionais entre os diversos programas e aplicativos em execução simultânea

18. Sequências de *bits* que formam comandos que os computadores entendem e obedecem são chamadas de:

- A. Programas
- ☒ B. Instruções
- C. Dígitos binários
- D. Palavra
- E. *Assembly*

19. Um determinado endereço da memória de um computador contém o seguinte número binário: 1001010100101110. Analise as afirmações a seguir:

- I. A informação armazenada é um dado armazenado que corresponde a um número inteiro (cujo valor decimal é 38190).
- II. A informação armazenada pode ser uma instrução a ser executada pelo computador.
- III. A capacidade de usarmos números binários para representar e armazenar tanto dados quanto instruções é um dos fundamentos da computação. ✓

Marque a alternativa correta:

- ☒ A. Existem duas afirmações verdadeiras
- B. Todas as afirmações são falsas
- C. Apenas a afirmação II é verdadeira
- D. Apenas a afirmação III é verdadeira
- E. Todas as afirmações são verdadeiras

20. Qual a diferença entre *assembly* e *assembler*?

- ~~A. Apesar de diferentes esses termos representam exatamente a mesma coisa, a linguagem que os computadores conseguem entender e executar~~
- ~~B. *Assembly* é a linguagem de máquina que um computador entende, no formato binário (0 e 1); o *Assembler* é um programa que traduz a versão das instruções em linguagem simbólica para o código binário~~
- ~~C. *Assembler* é uma linguagem simbólica que representa as as instruções que uma máquina consegue entender; *Assembly* é um programa que traduz a versão das instruções em linguagem simbólica para uma versão em código binário~~
- ☒ D. *Assembly* é uma linguagem simbólica que representa as as instruções que uma máquina consegue entender; *Assembler* é um programa que traduz a versão das instruções em linguagem simbólica para uma versão em código binário
- ~~E. Nenhuma das respostas acima~~

21. Considere o seguinte programa na linguagem C, cujo código fonte está salvo no arquivo "ola.c":

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      printf("%s", "Olá, mundo!\n");
6      return 0;
7  }
```

Se você abrir um terminal e digitar o comando a seguir, o que você está fazendo?

\$ gcc -std=c17 -Wall -S -o ola.s (ola.i)

- ☒ A. Estou solicitando ao compilador para pegar o arquivo pré-processado (ola.i) e gerar um arquivo com o código em *assembly* (ola.s)
  - B. Estou solicitando ao compilador para pegar o arquivo pré-processado (ola.s) e gerar um arquivo com o código em *assembly* (ola.i)
  - C. Estou solicitando ao *assembler* para pegar o arquivo pré-processado (ola.i) e gerar um arquivo com o código em *assembly* (ola.s)
  - D. Estou solicitando ao *assembler* para pegar o arquivo pré-processado (ola.s) e gerar um arquivo com o código em *assembly* (ola.i)
  - E. Nenhuma das respostas acima
22. O processo de compilação e execução de um programa cujo código fonte está na linguagem C envolve várias etapas, listadas abaixo (não estão em ordem!):
- 5 1. O *loader* carrega o programa para a memória
  - 2 2. O compilador traduz um arquivo pré-processado em um outro arquivo, contendo instruções simbólicas que representam as instruções que o computador entende e executa
  - 3 3. O *linker* faz a ligação entre o arquivo objeto do programa e os demais arquivos objeto das bibliotecas incluídas no programa
  - 4 4. O *assembler* traduz as instruções em linguagem simbólica para uma versão binária, gerando um arquivo de objeto
  - 1 5. O compilador faz o pré-processamento do código fonte e gera um arquivo pré-processado (após tratar as diretivas de compilação)

A ordem correta das etapas de compilação é:

- ~~A. 2 - 4 - 5 - 3 - 1~~
- ☒ B. 5 - 2 - 4 - 3 - 1
- ~~C. 1 - 2 - 4 - 3 - 5~~
- ~~D. 5 - 2 - 3 - 4 - 1~~
- ~~E. 5 - 4 - 2 - 1 - 3~~

5 - 2 - 4 - 3 - 1

23. Entre os *softwares* e o *hardware* existe uma **interface** de tradução responsável pelo funcionamento do computador: seu programa solicita alguma ação e essa interface faz a tradução do código de seu programa para a linguagem que o computador entender. O que é essa interface?

- A. O código fonte do programa
- B. As instruções binárias que o processador entende
- ☒ C. A arquitetura do conjunto de instruções
- D. O compilador
- E. A linguagem *assembly*

Assembly



24. São características das linguagens de alto nível, exceto:

- A. Facilmente compreensível
- B. Grande produtividade para o programador
- C. Alta portabilidade
- D. Facilidade de manutenção
- ☒ E. Interface entre o *software* e o *hardware*

25. Observe o seguinte arquivo:

```
1          .LC0:
2          .string "Olá\303\241, mundo!"
3          main:
4              addi    sp, sp, -16
5              sd      ra, 8(sp)
6              sd      s0, 0(sp)
7              addi    s0, sp, 16
8              lui     a5, %hi(.LC0)
9              addi    a0, a5, %lo(.LC0)
10             call    puts
11             li      a5, 0
12             mv      a0, a5
13             ld      ra, 8(sp)
14             ld      s0, 0(sp)
15             addi    sp, sp, 16
16             jr      ra
```

Podemos afirmar que:

- A. Esse arquivo foi criado pelo *assembler*, mas ocorreu um problema durante a compilação pois a palavra "Olá" foi traduzida incorretamente para "Olá\303\241" (linha 2)
  - B. Esse arquivo é o código-fonte pré-processado de um programa que tenta imprimir "Olá, mundo!", mas ainda precisa ser compilado para ser transformado em código *assembly*
  - C. Existem 13 instruções nesse código como, por exemplo, a instrução ".string" que imprime um texto no terminal
  - D. Esse arquivo é um exemplo do arquivo produzido quando a etapa de "ligação" do processo de compilação é executado.
  - ☒ E. Instruções como *sd*, *lui* e *mv* fazem parte da arquitetura desse processador
26. Você está gerando o código *assembly* de um determinado programa em C para diferentes arquiteturas de conjunto de instruções. Assinale a alternativa correta:
- A. O código *assembly* será idêntico, pois corresponde ao mesmo programa
  - B. Não é possível gerar *assembly* para diferentes arquiteturas, somente a x86
  - C. Será necessário reescrever o código fonte para que o *assembly* possa ser gerado corretamente para cada arquitetura
  - D. Não haverá diferença se a arquitetura for de 32 ou de 64 bits
  - ☒ E. O código *assembly* será diferente pois, mesmo que o programa seja o mesmo, cada arquitetura tem um conjunto diferente de instruções

27. Você está discutindo com seu colega de turma a respeito das diferenças entre a linguagem *assembly* e a linguagem de máquina. Seu colega afirma, apesar de você não ter certeza, de que *assembly* é a linguagem formada pelas representações simbólicas das instruções (comandos) que um computador consegue entender e executar, e que a linguagem de máquina nada mais é do que a versão binária dessas instruções. Você:

- ☒ A. Concorda com seu colega
- B. Não concorda com seu colega pois *assembly* é a linguagem formada pelo conjunto de instruções, e não por representações simbólicas
- C. Não concorda com seu colega pois linguagem de máquina já é a representação em *assembly* do código de um programa
- D. Não concorda com seu colega pois a linguagem de máquina é que é a representação simbólica das instruções, e *assembly* é a versão binária dessas instruções
- E. Não concorda com seu colega pois a representação simbólica das instruções é o *assembler*, e o código de máquina é o *assembly*

#### 4 Os componentes clássicos

28. Os cinco componentes clássicos do computador são:

- A. Entrada, saída, processador, linguagem de máquina e memória
- B. *Control* e *datapath* (que juntos formam o processador), memória, software e compilador.
- C. Disco, processador, memória, compilador e programas
- D. *Control* e *datapath* (que juntos formam o processador), interface, compilador e memória
- ☒ E. *Control* e *datapath* (que juntos formam o processador), memória, entrada e saída.

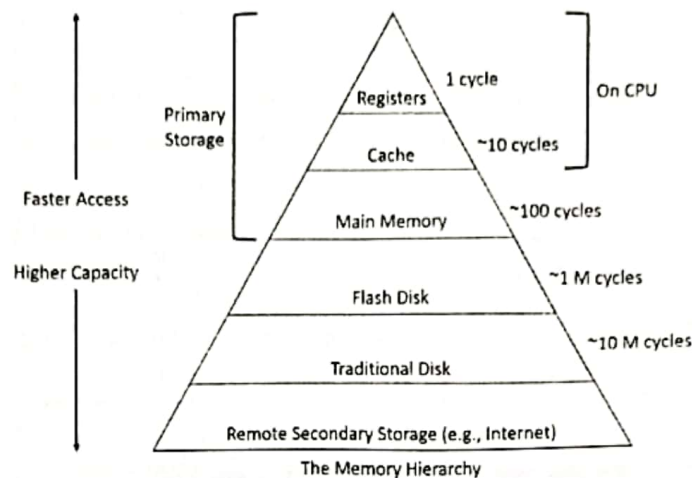
29. Considere as seguintes descrições:

1. Um mecanismo através do qual o computador é alimentado com dados e informações
2. O componente do processador que realiza operações aritméticas
3. Área de armazenamento na qual os programas são mantidos quando estão em execução, e que contém os dados necessários para esses mesmos programas
4. Um mecanismo através do qual o computador retorna o resultado de algum processamento para o usuário
5. O componente do processador que comanda o *datapath*, a memória e os dispositivos de entrada e saída, de acordo com as instruções do programa

As descrições anteriores que correspondem, respectivamente, ao *datapath*, ao *control*, à memória, aos dispositivos de saída e aos dispositivos de entrada são:

- ~~A. 1 — 2 — 3 — 4 — 5~~
- ~~B. 5 — 1 — 3 — 5 — 4~~
- ☒ C. 2 — 5 — 3 — 4 — 1
- ~~D. 2 — 5 — 3 — 1 — 4~~
- E. 2 — 3 — 5 — 4 — 1

30. A figura a seguir ilustra diversos componentes da **hierarquia de memória** em um computador, indicando a quantidade de ciclos de processamento necessários para o acesso à um dado armazenado nessas memórias (M representa milhões):



Com base na figura acima podemos concluir exceto:

- A. A memória *cache* é cerca de dez vezes mais lenta do que os registradores ✓
- B. A memória primária é composta pelos registradores, cache e memória principal do computador ✓
- C. O acesso ao disco rígido é cerca de cem mil vezes mais lento do que o o acesso à memória principal
- **D.** As memórias com maior capacidade são as memórias com menor tempo de acesso e, portanto, apresentam o melhor custo/benefício
- E. Os registradores e memória cache estão localizados dentro da CPU para otimizar o tempo de acesso e, assim, aumentar a performance dos processadores.

## 5 Performance

31. A métrica de performance que nos diz o tempo total exigido para o computador completar uma tarefa, incluindo acessos à memória, acesso ao disco, atividades de entrada/saída, *overhead* do sistema operacional, tempo de execução da CPU e tempo em espera, entre outros, é:
- **A.** Tempo de resposta B. Throughput C. CPI D. Período de clock E. Taxa de clock
32. A métrica que nos informa sobre a quantidade total de trabalho (tarefas) que um computador consegue realizar em um determinado período de tempo é a:
- A. Tempo de resposta • **B.** Throughput C. CPI D. Período de clock E. Taxa de clock
33. Sabe-se que uma medida simples do desempenho de um computador é dado pela sua **performance**, calculada como o inverso do tempo de resposta:

$$\text{Performance} = \frac{1}{\text{Tempo de Resposta}}$$



Se a performance de dois computadores, A e B, for comparada e obtivermos o seguinte resultado,

$$\frac{\text{Performance}_A}{\text{Performance}_B} = \frac{150}{15} = 10$$

podemos afirmar que:

- A. A performance do computador A é pior do que a do B
  - B. A performance do computador B é melhor do que a do A
  - C. A performance dos computadores A e B é equivalente
  - ☒ D. A performance do computador A é melhor do que a do B
  - E. Não é possível dizer pois o cálculo apresentado está errado
34. Considere um programa em C que faz a soma de dez milhões de números. Se executarmos esse programa e solicitarmos ao utilitário `time` do Linux para medir o tempo de execução, obteremos algo semelhante ao ilustrado na figura a seguir:

```
[abrantesast@ideapad ~]$ time ./somar 10000000
=====
Soma de 10000000 números
=====
Performance:      Runtime (ms)      MFLOPS
-----
Soma:              21.000147      952.374292

real    0m0.146s
user    0m0.126s
sys     0m0.020s
```

Com base no resultado ilustrado na figura acima, assinale a alternativa incorreta:

- A. O tempo total do programa foi de aproximadamente 21 ms
  - B. O tempo de CPU gasto no código da soma, dentro do processo do usuário, foi de 0,126 s.
  - C. O tempo de CPU gasto dentro do kernel, em benefício do programa, foi de 0,02 s.
  - ☒ D. O tempo de CPU gasto dentro do processo do usuário é influenciado por outros fatores como outros processos, acesso à memória, acesso ao disco, atividades de entrada/saída, tempo em espera e *overhead* do sistema operacional
  - E. Podemos identificar, dentro do tempo de resposta, o tempo de CPU, que é o tempo real que a CPU gasta realizando uma tarefa específica, sem incluir outros fatores como outros processos, acesso à memória, acesso ao disco, atividades de entrada/saída, tempo em espera e *overhead* do sistema operacional. Esse tempo de CPU pode ser subdividido, ainda, em tempo de CPU gasto no processo do usuário, e tempo de CPU gasto no kernel (em benefício do processo).
35. O que é a **quantidade de ciclos de clock**?
- A. É a duração (medido em alguma unidade de tempo) de cada ciclo de *clock*
  - ☒ B. É a quantidade de ciclos de *clock* que ocorre em determinado período, por exemplo, durante a execução de um programa
  - C. É o inverso do período de *clock*
  - D. É o inverso da taxa de *clock*
  - E. Nenhuma das respostas acima



36. Analise as seguintes informações da CPU de um notebook:

```
labrantesas@ideapad:~$ cat /proc/cpuinfo
processor       : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 165
model name    : Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz
stepping      : 2
microcode     : 0xf0
cpu MHz       : 2500.000
cache size    : 8192 KB
```

Pode-se afirmar que:

- A. O período de *clock* é de 2,5 GHz
- ☒ B. A taxa de *clock* é de 2,5 GHz
- C. O período de *clock* é de 10300 H
- D. A taxa de *clock* é de 10300 H
- E. O período de *clock* é de 250 picossegundos

37. Considere as seguintes métricas de desempenho de computadores:

- 1. Tempo de resposta ☐
- 2. Quantidade de ciclos de *clock*
- 3. Período de *clock*
- 4. *Throughput* ☐
- 5. Taxa de *clock*
- 6. Tempo de CPU (usuário/sistema) ☐

São métricas de interesse do usuário e do projetista:

- ☒ A. Usuário: 1, 4, 6; Projetista: 2, 3, 5
- B. Usuário: 1, 2, 6; Projetista: 4, 3, 5
- ~~C. Usuário: 1, 4, 3; Projetista: 2, 6, 5~~
- ~~D. Usuário: 2, 3, 5; Projetista: 1, 4, 6~~
- ~~E. Nenhuma das respostas acima~~

38. Existe uma relação entre o tempo de CPU e métricas relacionadas ao *clock* de um processador. Essa relação é dada pela seguinte equação:

$$\begin{aligned} \text{Tempo de CPU} &= \text{Quantidade de Ciclos de Clock} \times \text{Período do Ciclo de Clock} \\ &= \frac{\text{Quantidade de Ciclos de Clock}}{\text{Taxa de Clock} \uparrow} \end{aligned}$$

Com base na equação acima, como é possível melhorar o desempenho de um computador?

- ~~A. Aumentando a quantidade de ciclos de *clock*~~
- B. Aumentando o período do ciclo de *clock*
- ~~C. Diminuindo a taxa de *clock*~~

- D. Diminuindo a taxa ou aumentando o período de *clock*
- ☒ E. Diminuindo a quantidade de ciclos de *clock*, diminuindo o período do ciclo de *clock* ou aumentando a taxa de *clock*
39. Uma métrica importante na avaliação do desempenho de computadores é a métrica abreviada por "CPI", calculada de acordo com a fórmula abaixo. O que essa métrica nos informa?

$$CPI = \frac{\text{Quantidade de Ciclos de Clock}}{\text{Quantidade de Instruções}}$$

- ~~A.~~ O número total de ciclos de *clock* que um programa gasta para executar
- B. O número médio de instruções que um programa gasta para executar
- ☒ C. O número médio de ciclos de *clock* por instrução em um determinado programa
- ~~D.~~ O número médio de instruções por ciclos de *clock* em um determinado programa
- ~~E.~~ Nenhuma das respostas anteriores
40. A equação clássica do desempenho de computadores é dada por:

$$\begin{aligned} \text{Tempo} &= \text{Quantidade de Instruções} \times CPI \times \text{Período do Ciclo de Clock} \\ &= \frac{\text{Quantidade de Instruções}}{\text{Programa}} \times \frac{\text{Quantidade de Ciclos de Clock}}{\text{Instrução}} \times \frac{\text{Segundos}}{\text{Período do Ciclo de Clock}} \\ &= \frac{\downarrow \text{Quantidade de Instruções} \times CPI}{\text{Taxa de Clock} \uparrow} \end{aligned}$$

Analise as seguintes afirmações:

- I. Se diminuirmos a quantidade de instruções é possível melhorar o desempenho ✓
- II. Se aumentarmos a taxa de *clock* é possível melhorar o desempenho ✓
- III. Se aumentarmos o CPI o desempenho piora ✓ ✗

Assinale a alternativa correta:

- ~~A.~~ Somente a afirmativa I é verdadeira
- ~~B.~~ Somente a afirmativa II é verdadeira
- ☒ C. Somente as afirmativas I e II são verdadeiras
- ☒ D. Todas as afirmativas são verdadeiras
- ~~E.~~ Todas as afirmativas são falsas

## 6 Instruções, a linguagem dos computadores

41. Um modelo abstrato de computador, que define as instruções, os tipos de dados, os registradores, o hardware de gerenciamento e as características fundamentais da memória e o modelo de I/O é:
- A. O projeto de hardware
- ☒ B. A arquitetura do conjunto de instruções
- C. A implementação de uma arquitetura computacional
- D. O projeto eletrônico de uma CPU
- E. Nenhuma das alternativas acima

42. Computadores que têm grandes conjuntos de instruções são denominados de:  
A. Intel B. RISC ☒ C. CISC D. EPIC E. ADDI
43. Na arquitetura RISC-V de 32 bits, o processador tem quantos registradores?  
A. 16 ☒ B. 32 C. 8 D. 64 E. 128
44. Considere que o endereço base do array  $A = [24, 33, 49, -15, 50]$  está no registrador  $t0$ . Considerando que nessa arquitetura RISC-V cada inteiro ocupa 4 bytes, qual instrução carregaria o inteiro 49 para o registrador  $t1$ ?  
A. `lw t1, 4(t0)`  
B. `sw t0, 2(t1)`  
C. `lw t0, 2(t1)`  
D. `sw t1, 2(t0)`  
☒ E. Nenhuma das alternativas anteriores
45. Considere a seguinte instrução em RISC-V:

`sub x28, x0, x28`

Sabendo-se que no registrador  $x28$  existe um número inteiro, o que esta instrução está fazendo?

- ☒ A. Está invertendo o sinal do número no registrador  $x28$   
B. Está substituindo o valor armazenado em  $x28$  pelo valor  $x0$   
C. Está fazendo uma operação de substring de bits  
D. Está salvando o valor do registrador  $x28$  no endereço de memória indicado pelo registrador  $x0$   
E. Nenhuma das alternativas anteriores.