

## Introdução à Programação

Prova 1 - Prof. Marcus Ramos - 31/08/2010

1. (0,5 ponto) A substituição das válvulas pelos transistores, e posteriormente desses pelos circuitos integrados, trouxe quais vantagens para os computadores e os seus usuários?

Essas evoluções tecnológicas tornaram os computadores menores, mais leves, mais econômicos (com gasto menor de energia), mais confiáveis (com ocorrência de um número menor de falhas de hardware) e mais baratos. Para os usuários, o computador se tornou mais acessível e parte da vida cotidiana, contribuindo na realização de tarefas até então inéditas.

2. (0,5 ponto) Que tipo de linguagens eram usadas para programar os primeiros computadores?

Linguagens numéricas, que eram programadas através de painéis com chaves ou diretamente através da manipulação do hardware, como por exemplo fios, válvulas e circuitos.

3. (1 ponto) Descreva as principais características das:
  - a. Linguagens de programação de alto-nível;

Elas são mais legíveis (por serem mais parecidas com a linguagem natural), mais seguras para se trabalhar (pois possuem uma grande quantidade de regras que devem ser observadas pelo programador), permitem uma maior produtividade do programador (que consegue escrever mais linhas de programa no mesmo intervalo de tempo) e geram programas que podem ser executados em múltiplas máquinas com maior facilidade.

- b. Linguagens de programação de baixo-nível.

São menos legíveis, menos seguras, conferem menor produtividade ao programador e não tem portabilidade. No entanto, elas conferem um maior poder e flexibilidade ao programador, que tem mais liberdade para construir o seu programa como quiser, ou quando esse necessita acessar o hardware diretamente.

4. (1 ponto) Descreva como ocorre a execução de um programa escrito numa linguagem de programação de alto-nível, através de:
  - a. Compilação;

O programa-fonte é traduzido pelo compilador para um programa-objeto, o qual é então executado pelo hardware, produzindo os resultados desejados.

- b. Interpretação.

O interpretador lê as instruções do programa-fonte e as executa diretamente, sem gerar um programa-objeto equivalente.

5. (0,5 ponto) Uma vez executado um programa, e obtidos os resultados da sua execução, pode ser necessário refazer uma ou mais das etapas anteriores. Que etapas são essas e como elas se relacionam umas com as outras?

Essas etapas iniciam com o perfeito entendimento do problema a ser resolvido, prosseguem com proposição de solução genérica, do algoritmo elaborado para refletir essa solução de forma mais sistemática possível, do programa-fonte que reflete o algoritmo e do programa-objeto que é

executado pelo hardware diretamente. Em caso de falhas, pode ser necessário rever e refazer uma ou mais dessas etapas.

6. (1 ponto) Descreva os elementos que compõem o Modelo de von Neumann e descreva como ocorre a execução de um programa nesse modelo.

O Modelo de von Neumann é composto por (i) dispositivo de entrada, usado para transferir dados do meio externo para a memória; (ii) dispositivo de saída, usado para transferir dados da memória para o meio externo; (iii) memória, onde são armazenados os valores que são manipulados pelo programa e também o próprio programa; e (iv) unidade central de processamento, onde são decodificadas e executadas as instruções do programa e também são realizadas todas as operações de transformação de valores.

Um programa é executado sequencialmente. As instruções são transferidas da memória para a unidade central de processamento, uma de cada vez. Após a sua decodificação, as operações demandadas por cada instrução são executadas e a próxima instrução é lida, e assim por diante até atingir o final do programa.

7. (1 ponto) Qual é a diferença que existe entre um endereço de memória e um valor armazenado na memória do computador?

Um endereço de memória é a identificação numérica de uma posição de memória onde um certo valor (também numérico) pode ser armazenado. Uma mesma posição pode conter um entre uma coleção pré-determinada de valores possíveis, mas em cada instante ela armazena um único valor desse conjunto. Se um novo valor é gravado, o valor anterior é perdido.

8. (0,5 ponto) Por que motivo o uso compartilhado da memória para armazenar simultaneamente dados e o programa a ser executado facilita a programação de computadores?

Porque isso permite que o computador possa ser facilmente reconfigurado para executar outras tarefas sem que haja necessidade de mudar o hardware, o que pode ser demorado, complexo e introduzir falhas operacionais.

9. (0,5 ponto) Descreva as principais diferenças entre o Simple Computer Simulator estudado em sala de aula e a arquitetura de um computador real.

O simulador estudado é um interpretador, portanto o seu desempenho quando comparado com uma execução direta pelo hardware é bastante inferior. Além disso, um computador real possui mais posições de memória e um conjunto maior e mais rico de instruções.

10. (1 ponto) Responda:

a. Quantos bytes existem em 2 megabytes?

$$2 * 1.024 * 1.024 = 2^1 * 2^{10} * 2^{10} = 2^{21} = 2.097.152 \text{ bytes}$$

b. Quantos kilobytes existem em 4 gigabytes?

$$4 * 1.024 * 1.024 = 2^2 * 2^{10} * 2^{10} = 2^{22} = 4.194.304 \text{ bytes}$$

c. Quantos bits existem em meio megabyte?

$$0,5 * 1.024 * 1.024 * 8 = 2^{-1} * 2^{10} * 2^{10} * 2^3 = 2^{22} = 4.194.304 \text{ bits}$$

11. (0,5 ponto) Descreva a estrutura de um programa básico em C.

```
#include <stdio.h>
int main () {
... // aqui vão as declarações e os comandos
}
```

12. (1 ponto) Descreva a sintaxe e a semântica dos comandos da linguagem C:

a. `scanf`;

Sintaxe: `scanf (<"<formato>">,&<variável>)`, onde <formato> pode ser, entre outras opções, %d ou %f, e <variável> representa o nome de uma variável declarada no programa;

Semântica: a execução do programa é suspensa aguardando a digitação de um valor no teclado; quando a tecla ENTER é acionada, o valor digitado é convertido para a representação interna e armazenado na variável especificada.

b. `printf`;

Sintaxe: `printf (<"<cadeia>">,<expressões>)`, onde <cadeia> é uma cadeia qualquer de caracteres, eventualmente contendo as subcadeias %d e %f, e <expressões> representa uma série de expressões separadas por vírgula; pode-se usar qualquer quantidade de expressões, incluindo nenhuma, caso em que não se usa a vírgula depois do fecha aspas;

Semântica: os caracteres que forma <cadeia> são enviados para a tela do computador; as seqüências %d e %f são substituídas pelos valores gerados pelas expressões respectivas (usando correspondência posicional).

c. atribuição.

Sintaxe: `<variável> = <expressão>`, onde <variável> representa o nome de uma variável declarada no programa e <expressão> denota uma fórmula matemática que produz um valor;

Semântica: (i) <expressão> é avaliada e produz um valor; (ii) esse valor é copiado para <variável>, se sobrepondo ao valor anteriormente armazenado nela.

13. (1 ponto) Escreva um programa na linguagem C que receba como entrada seis algarismos (0 ou 1) correspondentes a um número binário, e escreva na saída o número decimal correspondente. Formate adequadamente a entrada e a saída e emita mensagens de orientação para o usuário.

```
#include <stdio.h>
int main () {
    int d, valor;
    printf ("Digite o 1º algarismo: ");
    scanf ("%d",&d);
    valor=d*32;
    printf ("Digite o 2º algarismo: ");
    scanf ("%d",&d);
    valor=valor+d*16;
    printf ("Digite o 3º algarismo: ");
    scanf ("%d",&d);
    valor=valor+d*8;
    printf ("Digite o 4º algarismo: ");
    scanf ("%d",&d);
    valor=valor+d*4;
    printf ("Digite o 5º algarismo: ");
    scanf ("%d",&d);
    valor=valor+d*2;
```

```
printf ("Digite o 6º algarismo: ");  
scanf ("%d",&d);  
valor=valor+d;  
printf ("O valor convertido para decimal é %d\n",valor);  
}
```