Arquitetura de Computadores 2018/19

Ficha 1

Tópicos: *Representação de dados, tipos e dimensões. Introdução ao C.*

Exercícios sobre bases numéricas e representação de dados

Observação: *O objetivo desta secção é recordar as bases numéricas e a representação binária.*

1. Considere a representação binária dos valores presentes na memória de um computador com palavras de 7 bits. Complete a tabela interpretando esses bits como:

- **números sem sinal**, apresentando o valor em base 10 (decimal) e em base 16 (hexadecimal);
- **números com sinal** em complemento para 2;
- **caracteres** na norma ASCII.

binário	s/sinal dec.	s/sinal hex	c/sinal dec	car ASCII
010 1010				
100 0110				
100 0011				
101 0100				
000 1010				
010 0001				
000 1000				

2. Apresente a representação binária (em complemento para dois) dos valores decimais apresentados na tabela.

decimal	5 bit	8 bit	12 bit
5			
-3			
-16			
15			
35			
260			

3. Para o caso da representação de números em palavras de 7 bits indique:

a) Quantos valores diferentes se podem representar em cada palavra, para números em binário, em hexadecimal e em decimal, com ou sem sinal?
b) Quantos caracteres diferentes serão no máximo possíveis representar?
c) Quais são o maior e o menor números que se podem representar sem sinal (dê a resposta em base 10 e em base 2)?
d) Quais são o maior e o menor números representáveis com sinal em complemento para 2 (dê a resposta em base 10 e em base 2)?

4. Crie um programa em Java que escreva quantos bytes são usados por cada tipo: `short`, `int`, `long`, `float` e `double`. Escreva também os maiores e menores valores representáveis por cada um. *Sugestão: veja as constantes nas classes Short, Integer, Long, Float e Double. (http://docs.oracle.com/javase/8/docs/api/)*

5. a) Crie um programa em C que imprime o tamanho em bytes de cada um dos tipos seguintes: `char`, `short`, `int`, `unsigned int`, `long`, `unsigned long`, `unsigned long long`, `float`, `double`. Para isso, utilize:

- "`sizeof (T)`", operador que devolve o tamanho ocupado por *T* em bytes (o valor devolvido é um `unsigned long`);
- "`printf("tamanho %lu\n", N)`", função que imprime "*`tamanho N`*" mudando de linha de seguida, onde *N* é um *unsigned long*† que será representado em base dez.
- *Nota: ao contrário do Java,* os valores para programas em C dependem da arquitetura onde o programa é executado (32 bits ou 64 bits).

b) Complete o programa anterior para escrever também o valor das constantes seguintes, definidas em `limits.h`: `SHRT_MIN`, `SHRT_MAX`, `INT_MIN`, `INT_MAX`, `UINT_MAX`, `LONG_MIN`, `LONG_MAX`, `ULONG_MAX`, `LLONG_MIN`, `LLONG_MAX`, `ULLONG_MAX`. Note que no *`printf`* deve indicar na formatação o tipo de dados que quer escrever e respetiva representação de acordo com o tipo da constante. Assim, deve usar `%u` para `unsigned int`, `%ld` para `long int`, etc. (veja o manual no terminal com o comando "`man 3 printf`", ou documentação online sobre o C). Justifique os valores dessas constantes com as dimensões antes obtidas para os tipos de dados respetivos.

6. Considere a linguagem Java e os operadores binários **&** , **|** , **~** , **>>** , **>>>** e **<<**. Crie um programa em Java que, para uma variável `byte b`, escreve no ecrã o resultado de cada uma das expressões a seguir indicadas.

a) Colocar o bit 1 de `b` a 1, mantendo os outros inalterados;
b) Colocar os 4 bits mais significativos de `b` a 0, mantendo os outros inalterados;
c) Colocar o bit 2 de `b` a 0, mantendo os outros inalterados;
d) Determinar se o bit 0 de `b` é 0 ou 1;
e) Multiplicar `b` por 16 (sem usar os operadores de multiplicação nem de soma);
f) Dividir `b` por 4 (divisão inteira, sem usar os operadores de divisão nem de subtração);
g) Multiplicar `b` por 12 (sem usar o operador de multiplicação);

Teste, escrevendo o resultado para `b` com os valores: `127`, `0x86`, `0b00110011`. Para tal escreva em base dez e em binário o valor inicial de `b` e o resultado de cada expressão. Justifique os resultados. *Sugestão: Veja o método Java `Integer.toBinaryString()`.*

7. Repita o exercício anterior na linguagem C. Declare `b` como `unsigned char`. Para testar, escreva os valores em decimal e em hexadecimal (dado que não tem uma função para escrever um valor em binário).

8. Admita agora que em Java não existe o método `toBinaryString`. Escreva um método (em Java) que permita escrever a representação binária de um valor do tipo `int`:

```
void printBin (int val)
```

9. Repita o exercício anterior na linguagem C.

---

† Em arquiteturas de 32 bits é provável que o operador `sizeof` devolva um `unsigned int` (se assim for use `%u` no `printf`)

Mais informação

O seguinte endereço indica uma página Web com informação detalhada sobre comandos Linux e a shell (CLI): **http://linuxcommand.org/**

Sobre a linguagem de programação C:

> Kernighan; Dennis M. Ritchie (March 1988). The C Programming Language (2nd ed.). Englewood Cliffs, NJ: Prentice Hall. ISBN 0-13-110362-8.

e outros (por exemplo): **http://publications.gbdirect.co.uk/c_book/**

http://www.cprogramming.com/tutorial/c-tutorial.html