

Arquitetura de Computadores 2017/18

Ficha 1

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

Exercícios sobre bases e representação de dados

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

- Considere a representação binária dos bits 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
0101010				
1000110				
1000011				
1010100				
0001010				
0100001				
0001000				

- Apresente a representação binária (complemente para dois) dos valores decimais apresentados na tabela.

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

- Para o caso da representação de números em palavras de 7 bits indique:
 - Quantos números diferentes se podem representar em cada palavra, para números em binário, em hexadecimal e em decimal, com ou sem sinal?
 - Quantos caracteres diferentes serão no máximo possíveis representar?
 - Quais são o maior e o menor números que se podem representar sem sinal (dê a resposta em base 10 e em base 2)?
 - Quais são o maior e o menor números representáveis com sinal em complemento p/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("texto %lu\n", N)`”, função que imprime “*texto N*” mudando de linha, onde *N* é um unsigned long[†] que será representado em base dez.
 - *Nota:* em C os valores obtidos dependem da arquitetura onde executa o programa.
 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 respectiva representação de acordo com o tipo da constante. Assim, deve usar %u para unsigned int, %ld para long int, etc. (veja o manual “`man 3 printf`” ou documentação 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 *);
 - f) Dividir *b* por 4 (divisão inteira sem usar /);
 - g) Multiplicar *b* por 12 (sem usar *);
 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.toString().*

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 não existe o método toString. Escreva em Java um método 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 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>

Para ter o Linux no seu computador pode, em vez de instalar o sistema, seguir a seguinte sugestão (para Windows e para macOS):

1. instale o VirtualBox 5.2.8-121009 de www.virtualbox.org para o seu sistema.
2. obtenha a imagem de Linux já instalado fornecido pelos docentes (<https://goo.gl/hJeik2>).
3. lance o virtualbox e, no menu, escolha “import appliance...” indicando o ficheiro da imagem. Pode depois apagar este download.

Para a instalação correr bem recomenda-se que tenha no seu computador >2GB de memória e >5GB de espaço livre em disco. Para login: **user** password: **user**