

Sistemas Digitais II

1ª Repetição de 1994/95

Leia atentamente o enunciado. Seja breve nas respostas, mas justifique-as convenientemente. Por favor, use letra legível ! Quando tiver que escrever programas, deverá apresentar uma listagem com comentários que facilitem a compreensão do programa. Com os melhores desejos de boa sorte nesta repetição, utilize bem os 100 minutos de que dispõe.

I

Suponha que tem que escrever um programa para fazer a gestão de contas bancárias. As contas são guardadas numa estrutura cujos elementos têm a seguinte forma:

```
typedef struct x{
    int    num_conta,
           valor;
    char   codigo;
    struct x *seguinte;
}conta;
```

Dispõe de uma rotina que dado o número da conta, devolve um apontador para a ficha dessa conta, ou NULL se não existir esse número. A declaração dessa função é da forma:

```
extern conta *Encontra_conta(int num_conta);
```

A rotina que faz a gestão de levantamentos é a seguinte:

```
1  int Faz_levantamento( int valor, int num_conta)
2  {
3  int tmp;
4  conta *apont_conta;
5
6  apont_conta = Encontra_conta( num_conta );
7  if( apont_conta == NULL )
8  {
9      printf("Não existe a conta %d\n", num_conta );
10     getch();
11     return -1;
12 }
13 tmp = apont_conta -> valor;
14 if( tmp < valor )
15 {
16     printf("Desculpe mas a conta  %d não dispõe de %d\n", num_conta, valor );
17     getch();
18 }
19 else
20 {
21     tmp -= valor;
22     apont_conta ->valor = tmp;
23 }
24 return 0;
25 }
```

I.1) Explique o funcionamento da rotina dada, descrevendo todas as linhas relevantes.

I.2) A maior parte dos computadores que processam este tipo de informação (incluindo os nossos do SIA) são multiutilizadores, logo poderá haver mais que um utilizador a correr a rotina dada. Existe algum perigo de neste caso haver erros com esta rotina ? Se sim, explique-os.

I.3) Existem *zonas críticas* nesta rotina ? Se sim, proteja-as utilizando semáforos, ou seja introduzindo rotinas *WAIT* e *SIGNAL* onde necessário (indique entre que duas linhas induz essas rotinas).

I.4) Há alguma possibilidade de haver um deadlock nesta rotina alterada com o Wait e Signal ? Justifique.

II

II.1) A maior parte dos sistemas que usa memória virtual tem a memória paginada e não segmentada. Porquê ? Se assim é, há alguma vantagem em a ter segmentada ? Quais ?

II.2) Explique, dando exemplos, porque é que não faz sentido dizer que um sistema operativo é *o melhor*.

III.3) Qual a função de um *dispatcher* ? O que é que é necessário guardar num sistema como o UNIX quando um processo deixa de estar a executar, e passa para a lista de espera.

III

Certamente conhece bem o simulador PCRobots, e em particular as rotinas *movement*(int velocidade, int direção) que põe o robot em movimento, e *getxy*(int *x, int *y) que verifica a sua posição.

Suponha agora que queria pôr o robot a patrulhar uma dada zona, e para isso escreve um ficheiro de texto com os waypoints desse percurso. O ficheiro terá em cada linha dois inteiros, separados por um espaço, que são as coordenadas x e y do waypoint. O ficheiro pode ter um número qualquer de waypoints (sempre pelo menos 1), e termina com um waypoint fictício de coordenadas 0,0.

Escreva um programa que lê o ficheiro de dados, fecha-o, e depois põe o robot a percorrer de forma cíclica esses waypoints.

IV

(esta questão tem pouca cotação, faça-a em último lugar)

O recente afundamento do N.R.P. S.Miguel, e todo o infeliz tratamento que a operação teve na comunicação social, veio lembrar a necessidade de haver ferramentas que permitam estudar o comportamento de navios em situações não normais (forte caimento, adorno, etc).

Escreva um programa que calcule o deslocamento em função da altura da água. Para tal, dispõe de um ficheiro com a boca do navio para todas alturas (da quilha até aos 10m de 10 em 10 cm) e para todas as balizas (que estão separadas de 1m). O ficheiro tem uma linha para cada baliza, e os valores das bocas são número reais, em metros. Para as integrações utilize a regra dos trapézios ($\text{área} = \text{largura} * (\text{altura1} + \text{altura2}) / 2$).

