

## Sistemas Digitais II

### 2ª Repetição de 1998/99

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. Tem 110 min para completar a repetição. Boa sorte !

#### I

Como decerto saberá o Departamento de Máquinas (ou se quisermos ser rigorosos o DFEN-MEC) está a organizar um seminário sobre “manutenção preditiva”. O nosso departamento vai apresentar um artigo sobre como fazer a análise dos parâmetros medidos. Isso supõe que há um sistema capaz de medir certos parâmetros sobre o funcionamento da máquina, como sejam a velocidade angular de veios, a temperatura, as vibrações que o sistema está a sofrer, etc. Para que seja possível uma política de manutenção preditiva eficaz, é necessário que esses parâmetros estejam a ser medidos continuamente. Para que os parâmetros sejam medidos continuamente é necessário dispôr de um sistema de medição e recolha de dados que seja barato.

Imagine que lhe pediam para propôr o desenho de um sistema baseado num microprocessador (ou microcontrolador) para fazer a aquisição de dados. O sistema deverá ter a capacidade para monitorar 2 sinais analógicos (com uma resolução de 8 bits), e 8 sinais digitais (por hipótese já com níveis TTL). Para além disso, deverá poder actuar sobre 4 linhas de controlo digitais, e ter uma ligação série para um PC.

- 1) Se tiver que escolher entre usar como microprocessador um 8085, 8051, 8086, 386EX, ou Pentium, qual escolheria ? Explique a razão da sua escolha, discutindo sucintamente as vantagens e desvantagens de cada uma das 5 opções.
- 2) Admita que alguém decidiu estudar a hipótese de usar o 8086.
  - 2.1) Em que modo (máximo ou mínimo) usaria o 8086 ? Explique a diferença entre estes dois modos de operação, explicando as vantagens e desvantagens de cada um deles.
  - 2.2) Os valores lidos pelos sensores serão guardados em vectores, que serão posteriormente processados. Uma das operações mais usadas nesse processamento é o cálculo da distância entre dois vectores. Dados dois vectores  $A = [A_1, A_2, .. A_n]^T$  e  $B = [B_1, B_2, .. B_n]^T$ , a distância euclideana entre eles,  $D$  define-se como  $D = \sqrt{(\sum_i (A_i - B_i)^2)}$ . Pretende-se escrever uma rotina para calcular QUADRADO da distância euclideana entre dois vectores que, por razões de eficiência, deverá ser escrita em assembler. A rotina deverá receber em SI e DI os endereços base dos vectores, e em AX o número de elementos dos vectores (ambos os vectores têm a mesma dimensão!). Cada elemento do vector é um byte em notação de complemento para dois (pode ser negativo ou positivo). A rotina deverá devolver em AX a distância entre os vectores, que poderá ser qualquer valor entre 0 e 64000 (ou seja é menor que 64K ). Escreva essa rotina, apresentando um fluxograma da mesma, e comente convenientemente o código.
  - 2.3) Como alguns colegas seus tiveram dificuldade em converter caracteres ASCII de dígitos numéricos em números, houve alguém que lhe deu a rotina abaixo indicada em “C”. Compile essa rotina para assembler do 8086, assumindo que quer um caracter como um inteiro ocupam 1 byte. Explique como é que a sua rotina recebe e devolve os parâmetros. Se quiser ter a cotação máxima, passe os parâmetros no stack, e durante a rotina não os tire de lá...

```
int Char2int( char ch)
{
    int x;

    if( ch > '9' || ch < '0' )
        x = 255;    /* Codigo de erro correspondente a dig.não num.*/
    else
        x = ch - '0';
    return x;
}
```

- 3) Admita agora que outra pessoa decide que é melhor usar o 8051 (ou uma das suas variantes).
- 3.1) Admitindo que o todo o código do sistema não ocupa mais de 3500 bytes, e que irá necessitar de guardar aproximadamente o mesmo volume de dados, diga que variante do 8051 usaria, e porquê.
  - 3.2) Escreva o logograma detalhado do sistema, usando a variante do 8051 escolhida acima.

## II

Admita agora que já dispõe de um compilador de “C” para o seu sistema, e que está a escrever um pequeno sistema operativo para o mesmo. Até agora já definiu o seguinte:

- Existe um tipo de estrutura chamado PROC que é usada para guardar descritores de processos.
- Um dos campos das estruturas PROC é o campo NEXT que é um apontador para PROC (para o descritor seguinte). Outro dos campos é um inteiro chamado BLOCKED\_ON, que é zero se o processo não estiver bloqueado em nenhum semáforo, e caso esteja, contém o número do semáforo onde está bloqueado.
- Não existir no máximo 255 semáforos, cada um identificado com um número (1 a 255)
- Existe uma variável que é um vector de inteiros chamado SEMAPHORE, com 255 elementos. Cada um desses inteiros é 0 se o semáforo a que corresponde (i.e. o semáforo com identificador igual ao índice do vector) estiver fechado (ocupado) e 1 em caso contrário.
- Existe uma variável chamada EXECUTING que é um apontador para o descritor do processo que irá ser executado a seguir.
- Existe uma variável chamada EXECUTABLES que é um apontador para a lista de descritores de processos que não estão bloqueados.
- Existe uma variável chamada BLOCKED que é um apontador para a lista dos descritores de processos que estão bloqueados.

Escreva em “C” uma rotina WAIT e uma rotina SIGNAL para esse sistema. Essas rotinas deverão ter como parâmetros um inteiro, que identifica o semáforo sobre o qual estão a operar, e um apontador para o descritor do processo que está a chamar a rotina. As rotinas deverão alterar as listas dos descritores (EXECUTING, EXECUTABLES e BLOCKED) conforme necessário !

Boa Sorte !

