

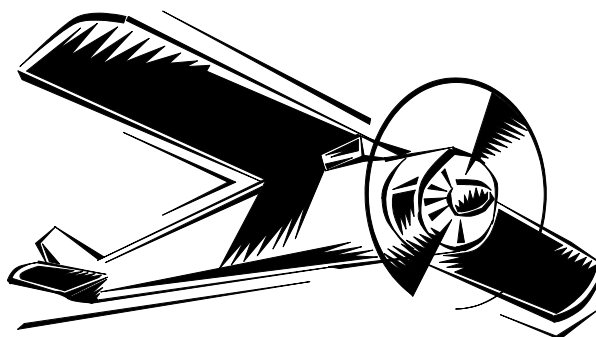
Sistemas Digitais II

2ª Repetição de 1997/98

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 90 min para completar a repetição. Boa sorte !

- 1) O 8086 usa memória segmentada. Explique em que consiste, quais são as vantagens, e quais são as desvantagens.
- 2) A ordem pela qual um mesmo conjunto de instruções máquina é executado pode ter uma influência dramática no tempo de execução (por isso é que a maior parte dos microprocessadores recentes fazem reordenação de instruções). Explique porquê.
- 3) Para que serve o prefixo LOCK nas instruções do 8086 ?
- 4) A Escola Naval está neste momento empenhada na realização do “Colóquio Vasco da Gama, os Oceanos, e o Futuro”, que visa sensibilizar os universitários portugueses para a importância do Mar. Um dos problemas com que o país se debate é a impossibilidade de vigiar permanentemente a ZEE, dados os elevadíssimos custos que acarreta. Esta turma está naturalmente empenhada em dar um contributo válido para a resolução deste problema nacional, e por isso vamos desenvolver o ATCHIM (Avião Tático de Caça, Hidrografia e Investigação Marítima). O ATCHIM será um avião autocomandado ligeiro, que pode ser lançado e recolhido por qualquer navio (até LF). O avião terá um motor principal, um altímetro, um GPS, um transmissor e receptor de rádio, actuadores dos lemes de profundidade, e de direcção, actuadores de ailerons, uma agulha de fluxo, etc. Como bons engenheiros que queremos ser, vamos dividir o problema numa série de problemas mais pequenos.

Admita que o motor principal é comandado digitalmente escrevendo num registo um número entre 0 (motor parado) e 255 (Avante toda). O Altímetro envia 16 bits em paralelo, com a altitude em metros. O GPS envia através de um porto série as coordenadas de grade do avião. As coordenadas de grade (que são cartesianas) consistem em dois números de 8 bits, precedidos pelo carácter “X” ou “Y” (também escritos com 8 bits). Admita ainda que os lemes de direcção e profundidade recebem também os comandos através de registos de 8 bits, sendo o número enviado em complemento para 2 (-127 é tudo a bombordo ou tudo para baixo, e +127 é tudo para estibordo ou para cima). Pretende-se usar um PC para controlar o avião.



4.1) Escreva o logigrama do interface entre o Bus ISA do PC e as linhas de controlo/recepção do motor, altímetro e lemes. Faça uma tabela dos endereços de I/O usados para cada dispositivo. Não se esqueça que de acordo com as especificações da IBM deverá usar apenas portos entre 300H e 31F.

4.2) Por hipótese, o sinal do GPS entra por um porto série que é implementado à custa de um 8255. Este integrado gera um IRQ4 cada vez que chega um caracter. Esse IRQ irá provocar um software interrupt 1CH, e o interrupt handler deverá então ler o endereço 280H para ler o caracter que acabou de chegar. Escreva em assembler o interrupt handler que recebe os dados do GPS e mantém no endereço absoluto 80000H e seguinte as coordenadas x e y do avião.

4.3) Escreva uma rotina em assembler que recebe em AX a altitude pretendida, e faz o avião ir até essa altitude.

- 5) “Um programa escrito para UNIX deverá correr em qualquer computador que corra esse sistema operativo”. Comente a veracidade desta afirmação, focando o problema da dependência/independência entre programa, o sistema operativo, e a plataforma onde este corre.
- 6) Imagine que é encarregado de escrever um servidor de caixas multibanco, que deverá correr em ambiente UNIX. Quando uma caixa envia uma mensagem ao servidor é necessário correr uma série de rotinas para descodificar a mensagem e actuar em conformidade. Para que não se percam mensagens é conveniente que haja um processo que está sempre à espera de novas mensagens e que lança um processo novo para tratar de cada nova mensagem que chega. Assim, cada mensagem tem direito a uma “máquina virtual”, e o servidor está sempre pronto a receber mais mensagens, por muito que elas demorem a processar. Para fazer a sincronização entre o sistema de recepção de mensagens e o servidor, vamos usar semáforos: o servidor bloqueia-se no semáforo **GetMsg**, e o sistema de recepção liberta esse semáforo sempre que chega uma mensagem. O servidor deverá então chamar a rotina **ProcMsg** que irá ler a mensagem so sistema de recepção, e processá-la. Entretanto o servidor deverá já ter voltado a ficar à espera da mensagem seguinte. Escreva um programa em C com chamadas ao sistema operativo UNIX que implemente este servidor.
- 7) Ontem à noite estava a trabalhar no meu computador (um PC com 16Mb de Ram a correr Windows95) e precisei de usar matrizes que ocupam 42Mb de memória. É possível manipular essas matrizes nesse computador, se o programa em questão tentar carregar as matrizes para memória ? Explique a sua resposta, indicando que mecanismos é que podem ou não ser usados, e quais as vantagens/desvantagens das diversas soluções possíveis.

