

## SISTEMAS DIGITAIS II

### Exame de 1995

Leia atentamente o enunciado. RESPONDA APENAS A 3 GRUPOS. Seja breve nas respostas, mas justifique-as convenientemente. Com os melhores desejos de boa viagem, e umas bem merecidas férias,

#### I

- 1 - Qual a diferença entre "compilar" um programa e "linkar" um programa ? Alguns sistemas (como por exemplo o Boland C, Turbo C, ou sistemas UNIX) têm a opção "make". O que faz esse comando ?
- 2 - Como decerto saberá, pensa-se poder vir a re-escrever o software para um simulador tático para a E.N. Há quem defenda que tal software deverá ser escrito em Assembler, em C, em Ada, e em Basic. Diga quais seriam, *neste caso* as vantagens/desvantagens de cada uma dessas linguagens, e qual seria a sua escolha.
- 3 - Porque é que, normalmente, os programas escritos em C são compostos por vários ficheiros de código fonte. Qual a função dos ficheiros com extensão ".h" nesse conjunto de ficheiros.

#### II

Pretende-se ter um sistema automático baseado num PC, que accione a campainha do edifício do comando a determinadas horas. Deverá escrever um programa em C para realizar essa tarefa. O programa deverá ler as horas a que a campainha irá tocar de um ficheiro. O programa deverá, obviamente ser decomposto em várias rotinas independentes. Admita que alguém já escreveu as seguintes rotinas:

void	Toca_campainha(void)	- Toca a campainha durante 3 segundos
long int	Ler_hora_real(void)	- Número de segundos desde as 0:00:00
void	Ler_horas_disco(void)	- Le os dados do disco e cria uma lista de <i>fichas</i> , que pode ser acedida pelo apontador <i>prim_ficha</i> . O tipo <i>ficha</i> é definido da seguinte maneira:

```
typedef struct fichas{
    long int    hora;
    fichas     *proxima;
} fichas;
```

- 1 - Escreva uma rotina que devolva um apontador para a primeira ficha cuja hora seja superior à hora real. A rotina deverá ter o seguinte protótipo:  
extern fichas \*Encontra\_primeiro\_toque();<sup>1</sup>

2 - Escreva uma rotina que tem como parametro de entrada um apontador para uma ficha, e que faça o seguinte: 1º Esperar que a hora real seja igual ou superior à hora da ficha; 2º Accione a campainha; 3º Acabe (retorne à rotina que a chamou).A rotina deverá ter o seguinte protótipo:

```
extern void Acciona_toque(fichas *a_tocar);
```

3 - Faça uma rotina que recebe como parametro de entrada um apontador para uma ficha, e devolva um apontador para a ficha seguinte. Se não existir ficha seguinte, deverá devolver um apontador para a primeira ficha. A rotina deverá ter o seguinte protótipo:

```
extern fichas *Encontra_proximo_toque(fichas *anterior);
```

4 - Diga se o seguinte programa resolve o problema, e comente-o linha a linha.

```
typedef struct fichas{
    long int      hora;
    fichas      *proxima;
}fichas;

fichas *prim;
main()
{
    fichas *aux;

    Ler_horas_disco();
    aux=Encontra_primeiro_toque();
    do
        {
            Acciona_toque(aux);
            aux=Encontra_proximo_toque(aux);
        }while( aux != prim )
}
```

### III

Imagine que o navio onde presta serviço recebe um RPV experimental português para fazer reconhecimento além horizonte. Imagine agora que tem que o programar para:

- 1 - Receber de um teclado o numero de waypoints que devem ser atingidos
- 2 - Receber as coordenadas (lat e long) desses waypoints (qualquer núm. de waypoints)
- 3 - Esperar que o operador dê autorização para o lançamento, e lançar.
- 4 - Dirigir-se sequencialmente a cada waypoint
- 5 - Abrir um páraquedas quando atinge o último.

É fornecido um sistema de desenvolvimento que inclui:

- Um computador de bordo (no RPV) que está ligado a um monitor e teclado no navio (essa ligação desfaz-se aquando do lançamento).
- Um compilador de C para o computador de bordo.
- Uma biblioteca de rotinas linkáveis em C que incluem as seguintes funções:

- GetPos(int \*lat; int \*long)      - Devolve as coordenadas presentes (de GPS)
- FlyDir(int rumo)                    - Põe o RPV apontado para o rumo "rumo"
- OpenShoot()                         - Abre o paraquedas

- 1) Escreva o programa de controlo de vôo em C.
- 2) Suponha que a orientação do RPV é feita pondo um valor de 8 bits (igual a rumo pretendido/2) num latch que se encontra no endereço F0000 do espaço de endereçamento de memória. Escreva um implementação possível da rotina FlyDir( int rumo).
- 3) Suponha que alguém escreveu umas rotinas que anotam todos os contactos avistados pelo RPV. Essas rotinas produzem uma lista de elementos com o seguinte formato

```
typedef struct x{
    char nome[30];
    int lat,long,dia,hora,min;
    x *next;
}contacto;
```

... e que existe uma variável do tipo apontador para contacto chamada prim\_contacto.

- 3.1) O que é que faz a seguinte rotina:

```
Xpto(contacto *ap)
{
    printf("Avistei o contacto %s às %d horas", ap->nome,ap->hora);
    if( ap->next!=NULL)
        Xpto( ap->next);
}
```

- 3.2) Faça uma subrotina que apague TODOS os registos de contactos, a partir de prim\_contacto.

#### IV

- 1) A especificação de um pacote ADA não contém código executável. Assim sendo, por que é que se usam essas especificações, e não se usa apenas o corpo que, no fundo, contém TODA a informação relevante ?
- 2) Considere o seguinte pacote:

```
TASK BODY guide IS
curr_dir: integer;
BEGIN
LOOP
    SELECT
```

```

ACCEPT fly_dir(dir: IN integer) DO
    curr_dir:=dir;
    END fly_dir;
ELSE
    DELAY(1);
END SELECT;
Correct_to_dir(curr_dir);
END LOOP;
END;

```

2.1) Expelique o funcionamento dessa task.

2.2) Escreva um programa muitíssimo simples que crie esta task, e depois se sincronize através de rendez-vous com ela. Explique o funcionamento dessa sincronização.

## V

1 - Que métodos conhece para calcular, num computador digital, a transformada de Fourier de um sinal ? Discuta as vantagens e desvantagens de cada método.

2 - Imagine que, num PC, tem um conversor A/D de 8 bits mapeado para o porto de I/O 300H, e um conversor D/A mapeado para o porto de I/O 301H. Assuma que o conversor A/D tem sempre disponível o valor da conversão (não é necessário mandar fazer a conversão e esperar pelo EOC), e que o conversor D/A também não necessita de qualquer controlo. Pretende-se que o PC funcione como um processador de sinal implementando várias funções, lendo os valores de entrada através do conversor A/D e escrevendo os valores de saída no conversor D/A.

2.1 - Escreva uma rotina em assembler que implemente um amplificador sem distorção de ganho 8 ( saída = 8 x entrada ).

2.2 - Escreva outra que implemente um filtro passa baixo caracterizado por  $y(n)=0.5 x(n) + 0.25 x(n-1) + 0.25 x(n-2)$ .

2.3 - Escreva outra que implemente um diferenciador.

3 - Escreva um programa que calcule a convulsão de um sinal de entrada com uma resposta impulsiva de um sistema. A resposta impulsiva tem comprimento 5, e o sinal de entrada tem comprimento 100 (i.e. tem 100 valores correspondentes a 100 instantes de tempo).



---

<sup>1</sup>Curiosamente o formato `extern void Encontra_primeiro_toque(fichas *primeira)`; não funciona, sendo necessário usar `extern void Encontra_primeiro_toque(fichas **primeira)`; Se quiser, explique porquê.