

Programação - Exame Especial

Eng.^a Informática; Eng.^a Informática – Pós-laboral; Eng.^a Informática - Curso Europeu

Duração: 2h30m

12/09/2018

1. Um ficheiro binário contém informação sobre a distribuição de um conjunto de documentos por várias pastas. Cada documento é caracterizado pelo seu nome, tamanho e pasta em que se encontra. Ao responder a esta questão pode assumir que não existem pastas dentro de pastas. A estrutura seguinte permite armazenar a informação de um documento:

```
struct doc{
    char nome[100];
    int tam;
    char pasta[100];
};
```

O ficheiro binário contém várias estruturas (1 por cada documento), sem nenhum ordenamento especial. Ao lado pode ver um exemplo de um diagrama representativo de um ficheiro binário em que existem 6 documentos distribuídos por 3 pastas.

Seguro 72 Docs
IRS 721 Docs
IrisCoutinho 12 Contactos
Pauta1718 234 Aulas
Casa 132 Docs
SaulBrito 13 Contactos

a) Escreva uma função em C que apresente no monitor o nome dos documentos que satisfaçam um determinado critério de pesquisa. O critério tem as seguintes componentes: nome da pasta em que se encontram os documentos e limites mínimo e máximo para o tamanho. A função recebe como parâmetros o nome de ficheiro binário e os critérios necessários para a pesquisa.

Exemplo: dado o ficheiro exemplificado em cima, se for indicada a pasta *Docs* e os limites 70 e 150, a função deveria escrever *Seguro* e *Casa*.

b) Escreva uma função em C que crie um ficheiro de texto com um resumo da informação armazenada no ficheiro binário. O ficheiro de texto deve indicar quais os nomes das pastas existentes e quantos documentos existem em cada uma. Cada uma das linhas do ficheiro de texto deve ter o seguinte formato: *nome da pasta: número de ficheiros*

O exemplo seguinte mostra o ficheiro de texto criado com base na informação do ficheiro binário indicado anteriormente:

```
Docs: 3
Aulas: 1
Contactos: 2
```

A função recebe como parâmetros o nome do ficheiro binário onde está armazenada a informação dos documentos e o nome do ficheiro de texto a criar. A ordem pela qual a informação surge no ficheiro de texto não é relevante.

2. Uma árvore binária é constituída por nós do tipo *no*:

```
typedef struct info no, *pno;
struct info{
    int num;
    pno dir, esq;
};
```

Escreva uma **função recursiva** em C que receba um ponteiro para a raiz da árvore e devolva o número de folhas que ela tem.

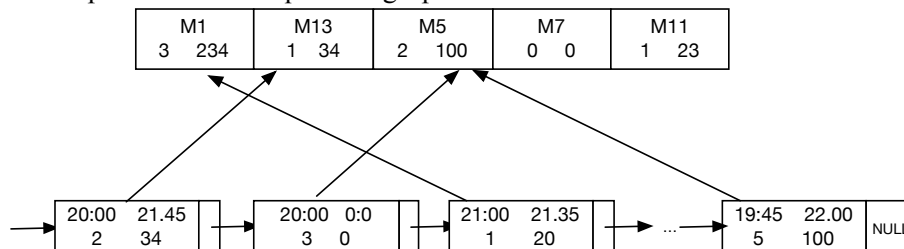
3. Um restaurante que apenas serve jantares utiliza uma estrutura dinâmica para armazenar informação sobre os grupos de clientes e a respetiva ocupação das mesas. A estrutura dinâmica tem duas componentes: um **vetor dinâmico** de mesas e uma **lista ligada simples** de grupos de pessoas que jantaram/estão a jantar no restaurante nesse dia. A estrutura dinâmica é criada com o auxílio das seguintes estruturas:

```
typedef struct tipoA mesa, *pMesa;
typedef struct tipoB grupo, *pGrupo;

struct tipoA{
    char id[20];
    int conta, valorTot;
};

struct tipoB{
    struct hora{int h, m;} in, out;
    int num, valor;
    pMesa ptr;
    pGrupo prox;
};
```

Na figura seguinte exemplifica-se a estrutura dinâmica. Existe um vetor dinâmico de estruturas do tipo *mesa* para guardar informação das mesas que fazem parte do restaurante: identificador alfanumérico, número de grupos que se sentaram nessa mesa e faturação dos jantares nessa mesa. O contador do número de grupos na mesa é atualizado assim que um novo grupo se senta, enquanto que a faturação é atualizada no final da refeição. A segunda componente é uma lista ligada simples, não ordenada, com nós do tipo *grupo*, contendo informação dos grupos que foram jantar ao restaurante nessa noite. Cada nó tem a hora de entrada, hora de saída, número de pessoas no grupo e valor pago. Na lista existem nós de grupos que ainda estão a jantar e nós de grupos que já terminaram a refeição e já libertaram a mesa. Os nós do primeiro tipo ainda não têm valor válido para a hora de saída e para o valor pago (pode assumir que têm o valor 0). Cada nó da lista tem um ponteiro para a mesa em que esse grupo está/esteve sentado.



a) Escreva uma função em C que descubra o nome da mesa em que está sentado o maior grupo no momento atual. Devem ser considerados apenas jantares que estejam a decorrer. A função recebe como parâmetros um ponteiro para o início do vetor dinâmico, a dimensão do vetor e um ponteiro para o início da lista ligada. Devolve um ponteiro para a estrutura do vetor correspondente à mesa onde está sentado o maior grupo. Caso não existam pessoas a jantar, a função devolve NULL. Se existir empate, pode adotar o critério de desempate que preferir.

b) Escreva uma função que adicione um novo grupo ao restaurante. O cabeçalho é o seguinte:

```
pMesa addGrupo(pMesa v, int *tot, pGrupo *lista, int h, int m, int
               pessoas, char *idM);
```

Recebe como parâmetros um ponteiro para o início do vetor dinâmico, o endereço de um inteiro onde está armazenada a dimensão do vetor, o endereço de um ponteiro para o início da lista ligada (i.e., um ponteiro para um ponteiro), a hora de entrada, o número de pessoas no grupo e o identificador da mesa onde se vão sentar. A função deve atualizar a estrutura dinâmica com o novo grupo. A mesa poderá não existir no vetor. Se for esse o caso, o vetor deve ser atualizado adicionando uma nova estrutura relativa a esta mesa. Se a mesa já existir, pode assumir que está livre e pronta a receber o novo grupo. A função devolve o endereço do vetor dinâmico das mesas.