

Lista de Exercícios 3: Arquivos (organização e índices)

*Professor: Moacir Pereira Ponti Jr.**PAE: Paulo Henrique Ribeiro Gabriel*

1 Organização de Arquivos

1. Faça um programa em C que identifique palíndromos em arquivos. Assuma que a frase não deverá ser carregada inteiramente na memória. Para tal, deverá utilizar o PUTC, que carrega caracter a caracter na memória principal. O objetivo é treinar o deslocamento de ponteiros em memória secundária.

- (a) Implementação com dois ponteiros do tipo FILE
- (b) Implementação com 1 ponteiro do FILE

Exemplo de arquivo:

```
-----  
Arquivo.txt  
-----  
arara  
polia  
aaaapaaaa  
-----
```

2. Explique o que é fragmentação de campos e registros. Quando e por que ela ocorre?
3. Se a separação entre registros e campos é feita por delimitadores, quais as restrições para a escolha desses delimitadores. Descreva uma situação que exemplifique sua resposta.
4. Aplique compressão de dados no seguinte trecho de texto para obtenção do código de Huffman “Fdgfggsdgaasggsagsdfg”.
 - (a) Construa a árvore de Huffman
 - (b) Calcule o número de *bytes* economizados
5. Quando se faz a alocação de registros de tamanho variável tendo em vista a existência de espaços vazios, existem três abordagens. Indique quais são e explique suas vantagens e desvantagens.
6. Faça uma rotina em C que implemente o **Keysort**, tendo em vista um registro de tamanho fixo com campos de tamanho fixo, mostrado abaixo.

```
struct Aluno {
    char nusp[7];
    char nome[20];
    char sobrenome[20];
}
```

A partir da estrutura, deverão ser lidas as chaves e deve ser feita a ordenação. Para identificação das chaves ordenadas utilize o RRN. Assuma que o computador em questão possui apenas 100 *bytes* de memória RAM.

7. A princípio, não é possível fazer busca binária em um arquivo de dados com registros de tamanho variável. Por que a indexação do arquivo torna a busca binária possível? Com um arquivo com registros de tamanho fixo é possível fazer busca binária. Isso significa que indexação não é necessária para arquivos de registros de tamanho fixo?

2 Indexação

1. É vantajoso manter um arquivo separado para armazenar apenas as chaves e os *byte-offsets*, ou RRNs, dos registros no arquivo de dados? Como isto afeta a inserção de um novo registro?
2. Por que título não é usado como chave primária no arquivo de dados descrito neste capítulo? Se título fosse usado como chave secundária, que problemas deveriam ser considerados na escolha de uma forma canônica para os títulos?
3. Qual o propósito em deixar um indicador de desatualizado no cabeçalho de um índice? Em um ambiente de multiprogramação, este indicador poderia ser encontrado setado por um programa por que outro programa está em processo de reindexação. Como o primeiro programa deveria responder a esta situação?
4. O que é uma lista invertida? Quando ela é útil? Como ela é mantida em memória secundária? Esquematize o conteúdo de um índice secundário organizado como lista invertida para um arquivo de dados hipotético.
5. O método e temporização do *binding* afeta dois importantes atributos de um sistema de arquivos: velocidade e flexibilidade. Discuta a relevância desses atributos, e o efeito do binding sobre o tempo em cada um dos atributos acima, para um sistema de informação de um hospital projetado para prover informação sobre os pacientes pelas chaves Nome do paciente, Código do paciente, Localização, Medicação, Médico, e Doença.