



22667– Operações em arquivos

Jander Moreira*

16 de agosto de 2018

1 Introdução e objetivos

As atividades previstas visam rever conceito de programação com arquivos e a representação de informações.

2 Preparação

A preparação para esta atividade envolve as leituras indicadas:

- Folk e Zoellick (1992), Cap. 3–Secondary storage and system software:
 - 3.1: Disks (exceto 3.1.4, 3.1.5, 3.1.7 e 3.1.8)
 - 3.2: Magnetic tapes (exceto 3.2.1 a 3.2.3)
 - 3.5: A journey of a byte
- Moreira (2015) ou Deitel e Deitel (2011), Cap. 11.
- Moreira (2011), Unidade 5–Organização de dados em memória secundária
 - Até seção 5.3.3.

3 Material

Programas e dados estão disponibilizados no Drive UFSCar (<https://drive.ufscar.br/f/73c1b4d3f6/?raw=1>). Os arquivos `dados.txt` e `dados.bin` contêm a lista dos mais ricos do mundo, segundo Época Negócios¹. Constam, no arquivo, o nome, a fortuna em bilhões de dólares e a empresa/área de atuação.

4 Atividades

Cada atividade proposta pode ser resolvida individualmente ou em grupos.

Exercício 1.

Compile e execute o programa `listatxt.c`. Entenda como os dados são lidos e como o final de arquivo é tratado. ??.

Exercício 2.

Compile e execute o programa `listabin.c`. Entenda como os dados são lidos e como o final de arquivo é tratado neste caso.

Exercício 3.

Avalie as diferenças entre os arquivos `dados.txt` e `dados.bin`. Tente editá-los para verificar os efeitos na execução dos programas dos Exercícios 1 e 2.

Dica: Faça cópias dos arquivos antes de editá-los para não perder os dados originais.

Exercício 4.

Escreva um programa para, dado o número do registro, apresentar seu conteúdo, usando o arquivo `dados.txt`. Considere que a numeração dos dados se inicia em zero. Por exemplo, ao se solicitar os dados do registro 11, devem ser fornecidas as informações: Schaeffler, com US\$ 25.3 bi, da empresa Schaeffler.

Exercício 5.

Faça um programa com a mesma proposta do Exercício 4 para o arquivo `dados.bin`.

Exercício 6.

Escreva um programa para alterar o valor da fortuna do 1º registro de dados (Beate Heister & Karl Albrecht Jr, 29.8, varejo) do arquivo `dados.txt`. Modifique o valor 29.8 para 112.45.

Neste exercício, o arquivo deve ser **modificado**, sem criar um novo arquivo alterado.

Exercício 7.

Escreva um programa que faça a mesma alteração proposta no Exercício 6, agora para o arquivo `dados.bin`.

Questão 1.

Qual foi a preocupação tida com o número de operações de leitura ou escrita na solução apresentada para os Exercícios 6 e 7?



*Moreira, J. – Universidade Federal de São Carlos – Departamento de Computação – Rodovia Washington Luis, km 235 – 13565-905 - São Carlos/SP – Brasil – jander@dc.ufscar.br

¹Fonte: <https://epocanegocios.globo.com/Empreendedorismo/noticia/2018/03/dono-da-amazon-jeff-bezos-lidera-lista-de-mais-ricos-do-mundo-da-forbes.html>, visitado em 17/7/2018.

Caso o arquivo de dados fosse muito grande (com milhares ou milhões de registros), os programas seriam eficientes?

Questão 2.

É possível escrever um programa em C que modifique um único byte em um arquivo de dados. Para tanto, basta abrir o arquivo, posicionar no byte a ser modificado e solicitar a transferência desse byte ao fluxo (*stream*) de dados, fechando então o arquivo. O Programa 1 exemplifica como esse código pode ser elaborado. Porém, sabe-se que um sistema de arquivos escreve em um arquivo, em uma única operação, uma quantidade de bytes equivalente a um bloco. Assim, se o tamanho do bloco for de 4096 bytes (4KiB), mesmo que se deseje gravar um único byte, 4096 serão efetivamente transferidos da memória principal para o dispositivo. Em outras palavras, ao se gravar um único byte, todo o bloco que o contém é reescrito.

Como ocorre a alteração de um único byte dentro de um bloco se todo o bloco é reescrito? Como os dados preexistentes não são perdidos?

Programa 1 Exemplo de código em C que modifica o valor de um único byte em um arquivo. Para simplificar o código, não são feitas verificações de erro ou consistência.

```
/*
   Alteracao de um byte em um arquivo:
   A posicao e o novo valor sao
   passados na linha de comando
   Jander 2018
*/
#include <stdio.h>

int main(int argc, char *argv[]){
    FILE *arq;
    int pos;
    unsigned char b; // um byte

    sscanf(argv[1], "%d", &pos); // posicao
    sscanf(argv[2], "%hhu", &b); // valor

    arq = fopen("dat", "r+");

    fseek(arq, pos, SEEK_SET);
    fwrite(&b, sizeof b, 1, arq);

    fclose(arq);

    return 0;
}
```

Referências

DEITEL, P.; DEITEL, H. C: *Como programar*. São Paulo: Pearson, 2011.

FOLK, M.; ZOELLICK, B. *File structures*. USA: Addison-Wesley Publishing Company, Inc., 1992.

MOREIRA, J. *Armazenamento e recuperação da informação*. São Carlos: UFScar, 2011. Disponível em: http://audiovisual.uab.ufscar.br/impreso/2016/SI/SI_Jander_RecuperacaoInformacao.pdf.

MOREIRA, J. *Arquivos*. 2015. Material de apoio. Disponível em: <http://cap.dc.ufscar.br/recursos/textos/MaterialApoio.Arquivos.2015-S1.Jander.pdf>.

5 Encerramento

Use o fórum de dúvidas do AVA para postar dúvidas ou comentários que tiver sobre esta atividade.