Universidade de São Paulo – USP
Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Ciências de Computação – SCC

http://coteia.icmc.usp.br/coteia/mostra.php?ident=572

SCC-203 - Algoritmos e Estruturas de Dados II

Responsável: Prof. Gustavo Batista
gbatista@icmc.usp.br

Estagiário PAE: Victor Laguna
vlaguna@icmc.usp.br

Aula de Laboratório I – Operações sobre arquivos

O objetivo desta aula é treinar as principais funções da biblioteca da linguagem C para manipulação de arquivos. Utilize os slides de aula para verificar como são realizadas as operações de associação e abertura de arquivos, leitura e gravação e fechamento de arquivos.

Para os exercícios a seguir você pode utilizar a linha de comando para informar ao programa sobre eventuais parâmetros. Para isso, a linguagem C provê dois parâmetros padrão para a função main, chamados argc e argv. argc armazena o número de parâmetros passados por linha de comando e argv armazena strings com os parâmetros passados. O seguinte programa deve ilustrar o funcionamento, repare que o nome do programa chamado é sempre o primeiro parâmetro (argv[0]).

```
void main(int argc, char **argv)

{

    int   i;



    if( argc < 2 )

        printf("Voce não passou argumento algum.\n");

    else {

        printf( "Voce passou %d argumento(s):\n\n", argc-1 );

        for(i = 1; i < (argc); i++)

            printf("%s\n", argv[i]);

    }

}
```

Exercício 1: Faça um programa capaz de ler um texto digitado do teclado e gravá-lo em um arquivo em disco. Utilize a tecla ESC (0x1b) para terminar a leitura. Utilize um editor de textos para ler e mostrar na tela o resultado do seu programa.

Dica: Você pode utilizar a função getch() (não é ANSI C!) no Windows para realizar leitura do console sem a necessidade de pressionar enter a cada caractere.

Exercício 2: Programas que fazem *dump* de arquivos são muito populares. Esses programas basicamente mostram o conteúdo do arquivo na tela na forma de números em hexadecimal e caracteres ASCII. Por exemplo, esta é a saída do programa hexdump do Unix para um arquivo conhecido:

```
46 49 4C 45 2A 20 61 72 71 3B 0D 0A 0D 0A 69 6E   FILE* arq;....in
74 20 6D 61 69 6E 28 76 6F 69 64 29 20 7B 0D 0A   t main(void) {..
20 20 20 20 63 68 61 72 20 63 68 3B 0D 0A 0D 0A       char ch;....
20 20 20 20 69 66 20 28 28 61 72 71 20 3D 20 66       if ((arq = f
6F 70 65 6E 28 22 74 65 73 74 65 2E 74 78 74 22   open("teste.txt"
2C 20 22 72 22 29 29 20 3D 3D 20 4E 55 4C 4C 29   , "r")) == NULL)
20 7B 0D 0A 20 20 20 20 20 20 20 20 70 65 72 72    {..        perr
6F 72 28 22 45 72 72 6F 22 29 3B 0D 0A 20 20 20   or("Erro");..
20 20 20 20 20 72 65 74 75 72 6E 20 31 3B 0D 0A        return 1;..
20 20 20 20 7D 0D 0A 20 20 20 20 77 68 69 6C 65       }..    while
20 28 66 72 65 61 64 28 26 63 68 2C 20 31 2C 20    (fread(&ch, 1,
31 2C 20 61 72 71 29 20 21 3D 20 30 29 0D 0A 20   1, arq) != 0)..
20 20 20 20 20 20 20 70 72 69 6E 74 66 28 22 25          printf("%
63 22 2C 20 63 68 29 3B 0D 0A 2F 2F 20 20 20 20   c", ch);..//
69 66 20 28 66 77 72 69 74 65 28 26 63 68 2C 20   if (fwrite(&ch,
31 2C 20 31 2C 20 61 72 71 29 20 3D 3D 20 30 29   1, 1, arq) == 0)
```

Repare que os números à esquerda são os valores byte-a-byte do conteúdo do arquivo, e os valores à direita são os respectivos caracteres ASCII. Alguns caracteres ASCII não são imprimíveis (0A 0D, por exemplo, que marcam o final da linha no DOS). Esses caracteres são impressos à direita como um ponto decimal. Cada linha apresenta 16 bytes do conteúdo do arquivo.

Algumas dicas úteis:

- As máscaras %x e %X do printf imprimem um número em hexadecimal;
- isprint(c) em ctype.h retorna verdade se c é um caractere imprimível.

Exercício 3: Faça um programa similar ao comando do Unix tail –n. Esse programa imprime na saída padrão as últimas *n* linhas de um arquivo especificado pelo usuário.

Exercício 4: Melhore o programa anterior para suportar arquivos originários de diversos sistemas operacionais. Infelizmente não existe padrão comum entre os sistemas operacionais para o(s) caractere(s) utilizado(s) como marcadore(s) de final de linha. Os caracteres CR (0x0D) e LF(0x0A) são normalmente utilizados para essa tarefa, entretanto com as seguintes diferenças entre os sistemas operacionais:

- **LF**: utilizado nos sistemas operacionais Multics, Unix e Unix-like (GNU/Linux, AIX, Xenix, Mac OS X, FreeBSD, etc.), BeOS, Amiga, RISC OS, entre outros;
- **CR + LF**: utilizado nos sistemas operacionais DEC RT-11 e diversos sistemas não–Unix como CP/M, MP/M, DOS, OS/2, Microsoft Windows, Symbian OS;
- **CR**: utilizado nos sistemas operacionais das máquinas Commodore, Apple II, Mac OS até versão 9 e OS-9.

Faça com que o seu programa receba um parâmetro que especifique qual é a terminação de linha do arquivo de entrada