

CI067 - Oficina de Computação

Exercícios # 04

1º semestre 2013

Entrada e Saída. Arquivos

PARTE I - Atividades em Laboratório

1. **(espremer)** Escreva um programa que copie sua entrada para sua saída, trocando cada cadeia de dois ou mais espaços por um único espaço.

DICA: A biblioteca padrão fornece diversas funções para leitura e escrita. Vamos utilizar **getchar()** e **putchar()** para ler e escrever um caracter de cada vez.

getchar() lê o próximo caracter da entrada padrão e o retorna como valor:

`c = getchar();`

putchar() imprime um caracter toda vez que for chamada:

`putchar(c);`

Observação: **getchar()** retorna a constante simbólica EOF (end-of-file) quando não há mais entrada. Devemos declarar a variável "c" com um tipo grande o suficiente para conter qualquer caracter e o valor EOF.

Portanto, "c" não pode ser do tipo "char", e sim do tipo "int".

(Contribuição do Prof. Egon Hilgenstieler).

- [Solução 1:](#)

2. Tome familiaridade com as funções da linguagem C para manipulação de arquivos. Para isto execute comando UNIX *man* para as seguintes funções: **fopen()**, **fclose()**, **fgetc()**, **fputc()**, **fgets()**, **fputs()**, **fprintf()**, **fscanf()**, **fread()**, **fwrite()**, **feof()**. Leia atentamente o texto para ter uma noção clara das funções existentes e como usá-las em um programa.

3. Analise e explique o que cada programa abaixo deve fazer. Depois disto, compile e execute cada um deles. Sua análise está de acordo com o comportamento observado? Troque idéias com os colegas ao seu lado e com o professor.
- [Programa 1](#);
 - [Programa 2](#);
 - [Programa 3](#).
4. **(contatexto)** Faça um programa que conte linhas, caracteres e palavras lidas da entrada padrão. Considere que uma palavra pode ser separada por um ou mais espaços, nova linha e/ou tabulação (' \t'). (*Contribuição do Prof. Egon Hilgenstieler*). —

- [Solução 1](#);

5. **(salvafrases)** Faça um programa que obtenha do usuário 5 frases, cada frase contendo no máximo 100 caracteres, e as grave em um arquivo.

- [Solução](#).

6. **(gravaint)** Fazer um programa que obtenha do usuário 20 números inteiros e os armazene em um vetor. Em seguida, o programa deve gravar o conteúdo do vetor DIRETAMENTE em um arquivo.

- [Solução](#).

7. **(prnint)** Fazer um programa que leia um arquivo produzido pelo

programa do exercício [6](#) e exiba convenientemente na tela os valores inteiros lidos do arquivo.

- [Solução](#).

Parte II - Exercícios

8. **(grava_agenda)**Fazer um programa que implemente uma agenda de endereços. Cada item da agenda possui os seguintes campos: nome (50 caracteres no máximo), endereço postal (250 caracteres no máximo), fone (DDD inclusive), endereço eletrônico (30 caracteres no máximo), cidade (30 caracteres no máximo), CEP e estado (2 caracteres).

O programa deve apresentar um diálogo em que pede no máximo 50 itens para a agenda. Caso o usuário deseje terminar a entrada de dados antes de completado os 50 itens, deve fornecer um nome vazio como entrada.

Uma vez terminada a entrada de dados, estes devem então ser gravados no arquivo **agenda.dat**.

- [Solução 1](#) - Usando `fprintf(3)`;
- [Solução 2](#) - Usando `fputs(3)`;
- [Solução 3](#) - Usando `fwrite(3)`;

9. **(ler_agenda)**Fazer um programa que leia uma agenda produzida pelo programa do exercício [8](#) e exiba adequadamente na tela os endereços lidos.

- [Solução 1](#) - Usando `fscanf(3)`;
- [Solução 2](#) - Usando `fgets(3)`;
- [Solução 3](#) - Usando `fread(3)`;

Armando Luiz Nicolini Delgado
2013-04-18