6897/9895 – Organização e Recuperação de Dados
Profa. Valéria

2ª Lista de Exercícios

1. Foi visto em aula um programa que lê um arquivo texto e o escreve em tela (slide 24 do arquivo "Operações em Arquivos.pdf"). Modifique esse programa para que ele leia os dados a partir da entrada padrão, em vez de um arquivo, e os escreva em um arquivo, em vez da saída padrão.

2. Faça um programa que solicite ao usuário o nome de um arquivo de texto e então calcule e apresente o seu tamanho em bytes e o número de linhas do arquivo.

3. Faça um programa que solicite ao usuário o nome de um arquivo de texto para que espaços repetidos sejam removidos. Um novo arquivo deve ser criado com o resultado da remoção.

4. Faça um programa que receba do usuário um arquivo de código em python e produza um novo arquivo contendo o mesmo código, mas com todos os comentários de linha removidos. Um comentário de linha começa com '#' em qualquer posição de uma linha e se estende até o final dela.

5. A quebra de linha em arquivos de texto muda dependendo do S.0. No DOS/Windows é utilizado o par de caracteres '\r' '\n' (decimal ASCII 13 e 10). Já no Unix/Linux, é utilizado apenas o caractere '\n'. Faça um programa que converta arquivos de texto do Windows para Linux e vice-versa. Seu programa deve receber o nome do arquivo a ser convertido e o sentido da conversão (se Windows → Linux ou Linux → Windows). O resultado da conversão deve ser gravado em um novo arquivo, preservando os dados do arquivo original. Lembre-se de abrir os arquivos em modo binário (tanto o de leitura quanto o de escrita), para que você consiga processar os caracteres de quebra de linha corretamente.

   Dica: Para receber argumentos pela linha de comando, você utilizar o módulo *sys*. Os argumentos passados estarão na lista *sys.argv* na mesma ordem em que foram digitados no terminal, sendo que o nome do script python sempre será o primeiro argumento da lista (i.e., está em *sys.argv*[0]).

6. Faça um programa que leia uma sequência de 10 números inteiros a partir da entrada padrão e, a seguir, escreva a sequência em um arquivo em formato binário. Considere que todos os inteiros deverão ser armazenados no arquivo com 4 bytes de tamanho.

7. Faça um programa que leia uma sequência de 10 valores de 4 bytes a partir do arquivo criado no exercício anterior e os imprima como na saída padrão como números inteiros.

8. Faça um programa que leia várias strings a partir da entrada padrão e as escreva em um arquivo. No arquivo, cada string deve ser precedida por um número inteiro de 2 bytes que corresponde ao seu comprimento. Assim, o arquivo resultante conterá tanto números binários quanto strings.

9. Faça um programa que abra o arquivo criado no exercício anterior e então leia e apresente as strings armazenadas nele. Lembre-se que cada string é precedida por um campo de 2 bytes que armazena o seu tamanho. Utilize a informação do tamanho para ler cada string com uma única chamada da função read.