

SSC0501 - Introdução à Ciência da Computação I (Teórica)

Professor responsável: *Fernando Santos Osório*
Semestre: 2011/1

E-mail: fosorio .at. icmc .dot. usp .dot. br
Web: <http://www.icmc.usp.br/~fosorio/>

TRABALHO DE PROGRAMAÇÃO (Extra-Classe) – REC

Este trabalho extra-classe deve ser enviado por e-mail ao professor com a implementação do programa descrito abaixo (programa fonte “.c”), **até a data de 26/07/2011**. Trabalhos recebidos após esta data serão desconsiderados – *O envio e aprovação do trabalho é condição necessária para aprovação na REC*. Enviar por e-mail ao professor, para os seguintes endereços (envie com cópia para ambos os endereços): fosorio@icmc.usp.br e fosorio@gmail.com

TRABALHO: Usando Arquivos HTML – Leia com atenção as instruções abaixo:

Faça um programa que possua um menu, conforme especificado, permitindo: (i) a leitura de um arquivo texto contendo uma página Web em formato HTML para a memória (vetor de linhas de texto); (ii) realize as operações pedidas de: eliminar as TAGs HTML, realizar uma estatística do conteúdo do texto (contagem de caracteres), busca e troca de palavras; (iii) salvar o conteúdo da memória (texto alterado) em um outro arquivo texto. Segue abaixo uma descrição mais detalhada de como devem ser implementadas as tarefas a serem executadas por este programa.

- Fazer um programa que possua um MENU com as seguintes opções: (exemplo de tela do menu de opções do programa)

```
>> Processamento de Arquivos HTML <<
MENU:
1 – Ler Arquivo HTML para a memória
2 – Remover TAGs HTML do Texto
3 – Procura e Substitui palavras no Texto
4 – Estatísticas do Texto
5 - Salvar o Texto em disco
6 - Sair do programa
Digite sua opcao:
```

- Descrição das Opções:

- 1) Ler o arquivo HTML: inicialmente deve ser solicitado o nome do arquivo em disco, sendo que este é um arquivo texto (arquivos HTML são arquivos texto comuns), onde o programa deve primeiramente abrir o arquivo e contar o número de linhas de texto (sem ler para a memória por enquanto). Baseado nesta estimativa inicial do número de linhas que o arquivo contém, fazer a alocação da memória, criando o vetor que irá armazenar os dados (linhas de texto) lidos do arquivo.

Observação: O tamanho máximo de uma linha de texto será de 100 caracteres.

Descritor do Arquivo: Criar uma estrutura que armazene um descritor do arquivo, contendo as seguintes informações: nome do arquivo, número de linhas, número total de caracteres, quantidade total de TAGs HTML (ver abaixo na opção 2 a definição de TAG HTML).

Exemplo da tela de leitura do arquivo:

```
>> Ler Arquivo HTML <<
Digite o Nome do Arquivo: texto.html
# Determinando tamanho do Arquivo...
# Total: 30 linhas
# Alocando memória para 30 linhas de texto
# Lendo o arquivo para memória...
# Arquivo Lido!

Pressione a tecla Enter para continuar. . .
```

- 2) Remover TAGs HTML: Considerando que uma tag HTML começa sempre por um caracter '<' e termina por um outro caracter '>', faça uma rotina que remova TODAS as TAGs HTML do texto armazenado em memória. Por exemplo, um texto como "Hello World" deverá ser transformado em "Hello World", onde as TAGs e foram removidas. Remover toda e qualquer TAG, que comece por '<' e termine por '>'. Conte quantas TAGs foram removidas e devolva este valor da quantidade de TAGs removidas como resultado da função Remove_Tags. A quantidade de TAGs removidas pode ser armazenada na estrutura do descritor do arquivo citado na opção anterior.

Exemplo da tela de Remoção de TAGs HTML:

```
>> Remover TAGs HTML <<
# Iniciando processo de remoção de TAGs...
# Total de TAGs removidas do texto: 10

Pressione a tecla Enter para continuar. . .
```

- 3) Procura e Substitui Palavras: O usuário irá informar uma palavra que deseja buscar no texto, e uma palavra que deve ser colocada em seu lugar. Esta rotina executa um procedimento similar ao "find-replace" dos editores de texto. Note que a palavra original e a palavra buscada não precisam ser do mesmo tamanho. O processo de busca e substituição deve ser repetido tantas vezes quanto o usuário desejar, ou seja, enquanto ele responder que deseja realizar mais uma substituição.

Exemplo da tela de consulta do próximo compromisso:

```
>> Procura e Substitui Palavras <<
Palavra Buscada: lucro
Substituir Por: prejuízo
# Total de Substituições Realizadas: 2
```

Deseja realizar mais uma substituição (s/n)? **s**

Palavra Buscada: **vender**

Substituir Por: **comprar**

Total de Substituições Realizadas: **1**

Deseja realizar mais uma substituição (s/n)? **n**

- 4) Estatísticas do Texto: Exibir na tela os dados relativos ao texto, que incluem os dados inicialmente coletados de nome, nro. de linhas, nro. de caracteres e nro. de tags, além disto exibir também o total de substituições de palavras realizadas até o momento. Além destas informações, exibir uma estatística da ocorrência de cada tipo de caracter presente no texto atual: Sabendo que um texto possui caracteres e que cada caracter possui um código ASCII entre 0 e 127, faça uma estatística da ocorrência de cada caracter (códigos ASCII), contando o nro. de vezes que cada um destes ocorre e depois determinando sua frequência (porcentagem) em relação ao total de caracteres do texto. Armazenar os dados estatísticos do texto em uma struct: nro. de linhas, nro. total de caracteres, frequências de ocorrência de caracteres. Exibir na tela a tabela de quantidade e frequência de ocorrência de cada caracter do texto. Por exemplo, se o texto possui 100 caracteres e o código ASCII 65 ocorre 15 vezes, deve ser exibido na tela: Código 65 – Qtde.: 15 - Freq.: 15 %;

>> Estatísticas do Texto <<

Nome do Arquivo : **texto.html**

Número de Linhas: **30**

Número de Caracteres: **924**

Número de TAGs Removidas: **10**

Número de Substituições de Palavras Realizadas: **3**

Estatísticas sobre o Texto:

Código 0 – Qtde.: 0 - Freq.: 0 %

Código 1 – Qtde.: 0 - Freq.: 0 %

Código 2 – Qtde.: 0 - Freq.: 0 %

...

Código 65 – Qtde.: 92 - Freq.: 10 %

...

Código 127 – Qtde.: 0 - Freq.: 0 %

Pressione a tecla Enter para continuar. . .

- 5) Salvar em disco: O programa deve salvar em disco um arquivo em formato texto, denominado de "saida.html", contendo o texto que estava armazenado em memória, com as devidas alterações realizadas.
- 6) Sair do programa: pedir para o usuário confirmar se deseja sair do programa, respondendo S/N (maiúsculo ou minúsculo). Se o usuário confirmar, terminar a execução do programa, se não confirmar, voltar ao menu principal.

Observações Importantes:

- Faça um programa modular com uma rotina para cada opção do menu (opções 1, 2, 3, 4 e 5).
- O programa não deve conter variáveis GLOBAIS, exceto o vetor que armazena o texto lido do arquivo em memória (vetor de texto). As demais variáveis devem ser todas locais e o programa deve passar adequadamente os parâmetros (por valor ou por referência, conforme a necessidade de cada rotina) e retornar valores também se for necessário.
- A modularidade, passagem de parâmetros, e retorno de valores serão avaliados.
- Não use comandos do tipo GO TO!

Data da Especificação: 11 de Julho de 2011

BOM TRABALHO!

FIM DO TRABALHO DA REC
