

Laboratorio de Introdução à Ciência da Computação I(SSC-102 turma 3)

Trabalho Final

Professor: Dr. Jó Ueyama (*joueyama@icmc.usp.br*)

Monitor PAE: Aurea Soriano Vargas (*asoriano@icmc.usp.br*)

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo, São Carlos

22/06/2012

1 Objetivo

Praticar as técnicas de programação aprendidas em aula através da implementação de uma aplicação simples de busca de palavras em um texto com sugestões, similar a um *mini-google*.

2 Especificações

Nesta seção serão apresentadas as especificações funcionais do programa que será desenvolvido.

2.1 Descrição Geral

O programa permitirá a busca de uma palavra em um texto, sem levar em conta maiúsculas de minúsculas. Primeiro a palavra será verificada se é correta o não em um vocabulário de referência. Se a palavra não é encontrada, o programa retornará sugestões de palavras. Para isso, vai ser usado o algoritmo de *Levenshtein*, para determinar se uma palavra é similar a outra. Depois, o programa procura a palavra no texto, e retorna a quantidade de vezes que foi encontrada e a(s) sentença(s) onde aparece.

No trabalho 1 foram implementadas algumas das funcionalidades do buscador. Neste segundo trabalho, algumas dessas funcionalidades serão modificadas y/o acrescentadas. A seguir, estas modificações e novas funcionalidades serão apropriadamente descritas.

2.2 Interface do Programa

O programa deverá apresentar uma interface que permita realizar as seguintes funcionalidades:

- Carregar o arquivo "vocabulario": O usuário deverá indicar o endereço no sistema de arquivos no qual estará o arquivo.
- Carregar o um texto na tela ou em um arquivo: O usuário terá a possibilidade de escolher se o texto será ingressado diretamente por teclado(na tela) ou se prefere indicar o endereço

no sistema de arquivos. No primeiro caso, novamente será empregada a sinal '!' para indicar final do ingresso de dados. No segundo caso, o texto deve ser lido até o final do arquivo. Ao acabar de ler o programa vai retornar o número total de sentenças e o número de palavras não repetidas.

- Palavra buscada: O usuário deverá ingressar a palavra para a busca.
 1. O programa vai procurar a palavra no vocabulário, se a palavra não é encontrada no vocabulário o programa vai retornar: "Você quis dizer:"e as 5 palavras mais similares do vocabulário (para esta parte vai ser usado o algoritmo Levenshtein).
 2. A seguir, o programa vai procurar a palavra no texto, se é encontrada retorna o número de sentenças onde foi encontrada e as sentenças, caso contrário retorna a mensagem "Sua busca - palavra - não encontrou nenhuma sentença correspondente."
- O resultado da busca deve ser reportado na tela ou em um arquivo de texto (escolhido pelo usuário), seguindo as especificações da subseção 2.3.

2.3 Detalhe do Programa

Os dados iniciais serão ingressadas na seguinte forma:

- Linha 1: home/usuario/vocabulario.txt (endereço do vocabulário)
- Linha 2: ingressar texto por teclado? ou ingressar endereço no sistema de arquivos? (pensar na melhor forma de fazer isso)
- Linha 3: (se é 1) texto. texto2.'!(final do texto) (se é 2) home/usuario/texto.txt

O programa vai retornar as estadísticas do texto da seguinte forma:

- Linha 4: Número total de sentenças do texto.
- Linha 5: Número de palavras não repetidas.

A palavra a buscar:

- Linha 6: palavra

O programa vai retorna:

- Linha 7: saída pela tela? ou em um arquivo? (pensar na melhor forma de fazer isso)(se é 1 ou 2 a saída é a mesma)
- Linha 8: Você quis dizer:palavra1, palavra2, palavra3, palavra4 ou palavra5. (se a palavra não é encontrada no vocabulário, caso contrário nada.)
- Linha 9:
 1. -num- de sentenças onde a palavra aparece: (se a palavra está no texto) 1 sentença1. 2 sentença2...
 2. "Sua busca - palavra - não encontrou nenhuma sentença correspondente."(se a palavra não está no texto)

Nota: Os números de cada linha apresentados no exemplo não aparecerão na entrada real do programa.

Terminar o programa: O usuário devera ter a possibilidade de realizar muitas buscas quantas vezes quiser sem que termine o programa, assim que precisa-se de escrever a palavra "exit!"para dar fim ao programa.

3 Indicações Gerais

- O trabalho deve ser desenvolvido em grupos de **máximo três alunos**.
- Cada grupo deve projetar e desenvolver um programa seguindo **todas as especificações** dadas.
- Deve empregar-se a **linguagem de programação C**.
- Data de apresentação: **22/06**.

4 Critérios de Avaliação

- Cumprimento de todas as especificações funcionais indicadas.
- Ordem e clareza do código fonte: modularização, comentários e indentação apropriados.
- Apresentação do relatório: Deve ser entregue no mesmo dia da apresentação do trabalho. Não serão aceitos relatórios entregados fora do horário de aula.

5 Estrutura do Relatório

O relatório a ser apresentado deve cumprir com as especificações detalhadas nesta seção. Contudo, se o grupo considera necessário, pode acrescentar seções ou subseções para melhorar a clareza do documento.

- **Capa:** código e nome da disciplina, turma, nomes completos e números USP dos integrantes do grupo.
- **Conteúdo:**
 - Resumo
 - Introdução: apresentação geral do trabalho e organização do documento.
 - Especificação da Implementação: estruturas de dados utilizadas, pseudocódigo da função *main* mostrando a sua interação com as outras funções e pseudocódigo das funções principais do programa. Toda escolha deve ser apropriadamente justificada.
 - Dificuldades Apresentadas: relatar quais foram as principais dificuldades encontradas durante a implementação do programa e as decisões tomadas pelo grupo para superá-las.
 - Conclusões
 - Referencias Bibliográficas

6 Considerações Finais

O **plagio** de programas não será tolerado. Quaisquer programas similares terão nota zero independente de qual for o original e qual for a copia.