Laboratório de Programação
CC2013

Projeto 2 - Entrada de texto Preditiva

Realizado por:
Rui Pacheco (up201906505)
Pedro Santos (up201907254)

Índice

1 Esquema com as estruturas de dados utilizadas

Primeiramente foi criada uma nova estrutura "Palavra" do tipo "palav" com o objetivo de auxiliar e facilitar o desenvolvimento do projeto, guardando assim todas as informações das palavras que são adicionadas ao dicionário. Nesta estrutura são guardadas informações como "palavra", "code", "code_original" e "ocur", sendo que as 3 primeiras são do tipo "char[]" e a última do tipo "int".

Na variável "palavra", tal como o nome indica, é guardada a palavra que é adicionada ao dicionário, já na variável "code" é armazenado o código até um máximo de 5 caracteres que irá ser usado na inserção da palavra na posição correspondente, na variável "code_original" é armazenado o código total da palavra que, caso seja menor ou igual a 5, será igual ao "code" e, por fim, na variável "ocur" é guardado o número de ocorrências da palavra no dicionário. Após isto e com o intuito de guardar toda esta informação foi criada uma estrutura "lnode" com a variável "info" do tipo "palav" onde é armazenada toda a informação anteriormente referida e possui ainda uma variável "next" do tipo "List" onde é guardado o próximo nó da lista.

Por fim, foi criada uma estrutura "List" que contém um apontador para a primeira palavra da lista em questão.

Todas as palavras são guardadas numa hash table "tab" do tipo List, ou seja, em cada posição da hash table existe uma lista que guarda todas as palavras correspondentes a essa posição da table, ressaltar ainda o facto de que apenas existem listas até à posição 99999, ou seja, palavras com tamanho máximo de 5 letras, pelo que, palavras com mais de 5 letras serão guardadas em listas com outras palavras mais pequenas e, por isto, a necessidade de uma variável "code" com o código da respetiva palavra até um tamanho máximo de 5 letras.

2 Estrutura Geral do Programa

A implementação deste algoritmo encontra-se dividida em 3 partes. Primeiramente foi criada a biblioteca "struct.h" onde são declaradas as estruturas, os seus parâmetros e todas as funções criadas com o intuito de as manipular. De seguida foi elaborado o "struct.c" onde estão implementadas todas as funções declaradas no ficheiro "struct.h". Por fim é criado o ficheiro "main.c" onde foi implementado todo o código propriamente dito, código este que utiliza funções e estruturas declaradas e implementadas nos ficheiros "struct.h" e "struct.c", anteriormente referidos, bem como estruturas e funções predefinidas da linguagem "c" utilizada para a construção deste algoritmo.

Ao iniciar o programa pela primeira vez o mesmo irá procurar um ficheiro "lusiadas.txt" que irá servir de dicionário, contudo, quando o mesmo for terminado o dicionário, já atualizado, será gravado em um ficheiro "lusiadas1.txt". Assim após a primeira execução e caso este ficheiro "lusiadas1.txt" esteja presente será este o ficheiro carregado com a finalidade de servir como dicionário, pelo que, o dicionário será sempre alterado e estará sempre atualizado conforme as preferências do utilizador, desta forma, o programa vai se adaptando e prevendo cada vez melhor as escolhas do utilizador, tornando-o mais eficiente.

Após a leitura do ficheiro as palavras são armazenadas nas hash tables de listas ligadas na posição correspondente ao seu "code", este "code" é originado com o auxílio da função "to_code" que ao receber uma palavra converte-a para o código correspondente, caso esteja a ler diretamente de um dicionário pela 1 vez, ou então utiliza o "code"que já esta presente na struct "Palavra" caso esteja a ler de um dicionário originado de uma execução anterior.

Após a gravação de todas as palavras do dicionário o utilizador tem a opção de escrever utilizando o modo inteligente, ativo por default, que lhe irá sugerir as palavras consoante as suas ocorrências aparecendo, assim, primeiro as mais utilizadas, reiterar ainda, que sempre que o utilizador utiliza uma determinada palavra as suas ocorrências são incrementadas adaptando, deste modo, o dicionário ao utilizador, ressaltar também o facto que quando o prefixo tem mais de 5 letras, como as listas só possuem um máximo de 5 letras por palavra, tal como referido anteriormente, é utilizada uma condição para confirmar se o prefixo está contido na palavra e, assim, só são apresentadas as palavras da lista que possuem este prefixo ao invés da lista ser apresentada na totalidade.

A par desta opção de escrita inteligente é também permitido ao utilizador a opção de escrita convencional permitindo assim, a inserção de novas palavras de forma manual e de pontuação.

Os diferentes modos de escrita são controlados e alternados com o pressionar do botão "DISABLE T9"

3 Estrutura dos ficheiros

No algoritmo estão presentes 3 funções "ler_ficheiro", "ler_ficheiro_gravado" e "save_changes" que se relacionam diretamente com os ficheiros com o intuito de carregar e gravar dados, respetivamente, deste modo, e caso o utilizador assim o pretenda pode carregar informações já existentes partindo do ponto onde ficou, assim, o dicionário vai se adaptando ao utilizador.

A função "save_changes" guarda os diferentes parâmetros da struct "palavra" cada um em uma linha diferente pelo que ao ler, caso seja um ficheiro originário de uma previa execução basta ler 3 strings e um inteiro e a cada 4 linhas adicionar a palavra à lista correspondente, este modo de leitura está presente na função "ler_ficheiro_gravado". Por outro lado a função "ler_ficheiro" é utilizada quando o dicionário não é originário de uma prévia execução, ou seja, lê palavra a palavra e por cada palavra gera o código correspondente e insere na lista da posição do código gerado.

4 Como executar o programa

Como já foi referido anteriormente o algoritmo encontra-se divido em 3 partes e, tendo em conta que "struct.h" é uma nova biblioteca a forma de a incluir nos ficheiros será diferente e, por isso, para executar o programa antes de correr o comum "gcc" para compilar esta nova biblioteca deverá ser gerada, para isso serão utilizados os seguintes comandos:

- gcc -Wall -c struct.c
- ar -rc libstruct.a struct.o
- ar -t libstruct.a
- gcc -Wall -o main main.c -L. -lstruct 'pkg-config –libs –cflags gtk+-3.0'

Após estes comandos a nova biblioteca já foi gerada e está pronta a ser utilizada bem como os ficheiros "struct.c" e "main.c" que já foram compilados e estão prontos a executar, para isto vamos utilizar o comando "./main" que irá iniciar a execução do algoritmo.