

Projeto 2 - Laboratório de Programação

Faculdade de Ciências da Universidade do Porto

Carolina Ferreira, up201905810

João Moura, up201904881

Tiago Eusébio, up201904872

Introdução

Este projeto tem como objetivo desenvolver um sistema de entrada de texto preditivo (idêntico aos telemóveis antigos com teclados de 12 teclas) em linguagem C, utilizando um dicionário de palavras para determinar com apenas um toque por tecla qual a palavra que o utilizador pretende escrever, atribuindo automaticamente a letra (ou símbolo) respetivo das diferentes possibilidades associadas a cada tecla através de uma interface semelhante à dos telemóveis antigos com o auxílio de uma GUI, usando o GTK Toolkit.

Estrutura de dados:

Para a elaboração do dicionário, optamos por criar um array de listas com o propósito de guardar prefixos de tamanho 5 e o número de ocorrências dos mesmos. Para isso criamos a livraria “bibp2”, onde implementados um conjunto de funções, previamente declaradas no header “bibp2.h”.

- “cria_lista”: cria um header da lista e coloca um ponteiro de lista apontar para ele.
- “pesquisa”: pesquisa na lista um nó no qual a string corresponde a um prefixo específico, utilizando a função “procura_lista”.
- “procura_lista”: procura na lista um nó no qual a string corresponde a um prefixo específico, caso exista guardamos o valor de uso associado a essa palavra (num), de

seguida apagamos o nó em questão da lista adicionando logo de seguida uma mesma string do apagado e com o valor (num+1) que definira o seu número de ocorrências e consequentemente a sua prioridade nas posições da lista, caso contrário inserimos um novo nó na lista utilizando “insere_lista”.

- “insere_lista”: insere um nó na lista e ordena-o de forma decrescente pelo valor da variável “num”
- “num”: converte vários números (correspondentes aos números das teclas do t9 clicadas) num só algarismo.
- “troca”: converte os caracteres que apresentam acentos em caracteres sem pontuação.

“next”: converte a sequência de números numa sequência de letras de forma a formar a palavra que se encontra na primeira posição da lista.

- “next1”: caso a palavra que venha do parâmetro next não seja a correta esta função troca a string “returnada” em next pela que se encontre na posição a seguir da lista em questão.

Para definir o tipo “nodes” e “List” usamos duas estruturas, esta definição tem características recursivas, pois uma lista é um apontador para uma estrutura e um dos membros dessa estrutura é uma lista. Cada nó da Lista tem dois parâmetros a “List next” referente ao próximo nó da lista e a variável “info” que é do tipo “List_words” que apresenta os parametros “string” e “num”, para guardar os prefixos e o número de ocorrências do mesmo, respetivamente.

Estrutura geral do programa:

Após declaradas todas as estruturas de dados criamos o ficheiro “main.c”, onde declaramos um array de listas “dicionario”, que guarda a palavra no parâmetro “string” onde a posição no array corresponde aos números das teclas clicadas para obter o prefixo.

De seguida, lemos o ficheiro “lusíadas.txt”, que no nosso programa é chamdo de “in.txt”, caracter a caracter e fomos inserindo-os as letras num vetor “word”, até encontrar

um sinal de pontuação, nesse caso inseríamos na última posição do vetor um terminador, pois significa que teríamos chegado ao final de uma palavra, seguidamente com o auxílio da função “num” determinamos quais os valores das teclas que seriam utilizadas para formar a palavra e dessa forma obtivemos a posição onde será inserida a palavra e com a utilização da função “pesquisa”, inserimos a palavra na lista, caso não existisse, em caso contrário aumentamos o valor da variável “num” correspondente ao número de ocorrências e reordenamos.

Para a construção da interface gráfica, optamos por uma tabela com 13 botões. O primeiro botão está localizado acima da label de escrita serve para trocar entre o modo de escrita inteligente e o modo normal, para isso chamamos uma função que consiste unicamente em mudar o valor de uma variável de controlo que dependendo do seu valor certos botões irão fazer coisas diferentes, em ambos os modos, o botão 1 serve para adicionar sinais de pontuação.

No modo inteligente, os botões (2-9) escrevem o próprio valor de forma a criar uma palavra codificada que a partir dos primeiros 5 dígitos (que sejam aceitáveis) desse número obtemos a posição à qual o dicionário deve aceder, quando o botão “*” é premido (devendo ser premido logo a seguir à sequência de números), a sequência de números passará a uma palavra que se encontrará na primeira posição da lista que, provavelmente, será a correspondente (uma vez que a lista esta organizada da palavra mais usada para a menos usada), se não, basta premi-lo novamente de modo a aceder à palavra que se encontre no próximo nó da lista.

Já no modo normal, esses mesmos botões funcionarão à base de timers para poder ser o utilizador a escolher ao certo a palavra e podendo adicionar palavras novas, sendo que a palavra só será dada como terminada se premir o botão “*” e nesse momento a palavra será procurada e se não existir na lista, será introduzido um nó com essa string e número de ocorrências igual a 1, no caso de já existir irá eliminar esse nó e inserir um que se diferencia apenas por ter um número de uso maior em uma unidade do que o anteriormente eliminado.

Os botões “0” e “#” tal como o botão 1 funcionam da mesma forma servindo para escrever um espaço branco na label e apagar o ultimo caracter escrito na mesma, respetivamente.

Breve explicação de como o programa se executa:

Ao executar o programa é apresentado ao utilizador uma interface gráfica que simula o teclado dos telemóveis antigos no modo normal, para alterar este modo deve premir o botão de cima que permite mudar para modo inteligente. No modo inteligente o utilizador deve clicar nas teclas correspondentes às letras das palavras que pretende escrever e de seguida deve clicar no “*” até aparecer a palavra desejada.

No modo normal o utilizador deve carregar nas teclas até aparecer a letra desejada e consequentemente a palavra, ao carregar no “*” a palavra é adicionada ao dicionário.

Em ambos os modos, a tecla “1” tem o propósito de inserir pontuação, a tecla “#” permite apagar caracteres e a tecla “0” serve para colocar um espaço.

Conclusão

No decorrer da realização do projeto tivemos várias dificuldades e nem todas elas foram completamente superadas tal como a utilização de ficheiros para guardar o conteúdo presente nas listas de modo a voltar a ter acesso ao mesmo numa próxima utilização ou a necessidade de fechar e voltar abrir a interface para trocar o modo de escrita. Devido a estas adversidades sabemos que o nosso programa não está 100% eficaz e que muitos aspetos do mesmo deveriam estar sem imperfeições, contudo todos os elementos do grupo trabalharam para tentar atingir a máxima eficácia do programa.