

Entrada de Texto Preditiva - T9



Laboratório de Programação

Grupo 24:

Murilo Marotta Reis Rosa (201900689)

Tiago Jorge Moutinho Gonçalves (201905179)

JUNHO 2021

PORTO

Índice:

Introdução	3
Estruturas de dados utilizados	4
Estrutura geral do programa	6
Breve explicação de como o programa de executa	5

Introdução

Texto preditivo é uma tecnologia de entrada onde uma tecla ou botão representa muitas letras e/ou números, como em teclados numéricos dos telemóveis antigos.

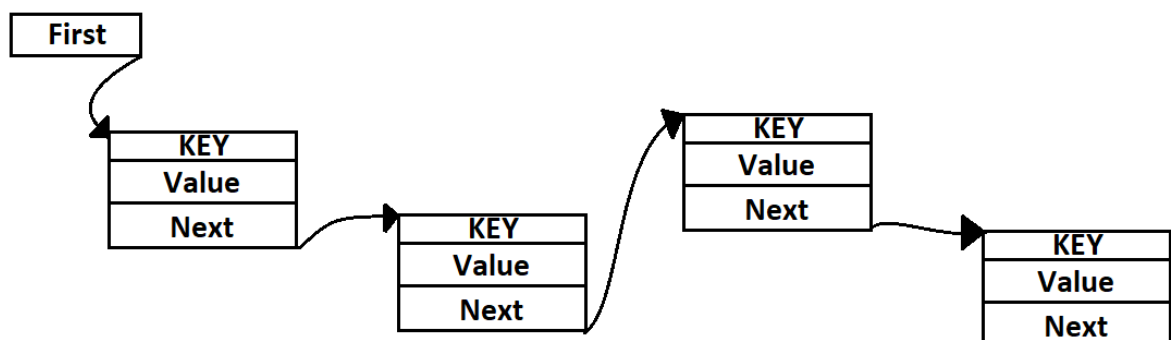
Um exemplo de tal tecnologia é a tecnologia T9 que significa “texto em nove teclas”.

O objetivo do T9 é facilitar a escrita de textos utilizando nove teclas numéricas, cada uma correspondendo a diferentes letras que tal tecla permite escrever. Por exemplo a tecla 2, permite escrever “a”, “b” e “c”. Sem a tecnologia de texto preditivo T9 seriam necessários 1, 2 ou 3 toques consecutivos na tecla, respectivamente. Com T9, pretende-se utilizar um algoritmo inteligente e preditivo para acelerar a introdução de texto através destes teclados como um dicionário de palavras para determinar qual a palavra que o utilizador pretende escrever pelo seu dicionário de palavras e frequência de utilização, sendo possível escolher e adicionar novas palavras ao mesmo dicionário. Para tal, utilizaremos a tecla 0 para ver as possibilidades e a tecla 1 para envio das palavras.

Estruturas de dados utilizados

Neste Trabalho optamos por usar diferentes tipos de estruturas de dados como String, StringT9, Dictionary, Linked List Map, Binary Search Tree Map e Hash Map.

- As estruturas String e StringT9, são muito semelhantes na medida em que ambas têm como atributo um “value” e um “length”, sendo que o “length” é o tamanho e o “value”, no caso da String, é a palavra em formato ASCII a que corresponde e, na StringT9, é a sequência de inteiros que corresponde a cada letra dessa palavra. Por exemplo para a palavra “inverno” teríamos a (String)->value = “inverno” e String->length = 7 e (StringT9)->value = “4683766” e StringT9->Length = 7. Além disso, o atributo size em String nos indica se temos espaço extra para adicionarmos mais caracteres, se necessário.
- A estrutura Linked List Map é constituída por vários nós, que possuem um atributo “value” e uma “key”. Além disso, cada nó aponta para o endereço de memória do nó seguinte e anterior. Deste modo, não estamos limitados a um certo tamanho, uma vez que são estruturas dinâmicas em que podemos adicionar e eliminar células facilmente. Adicionalmente, os nós estão organizados de forma decrescente pela sua “key”.



- A Binary Search Tree Map consiste numa Arvore Binaria Ordenada contendo Pares (Mapping). Assim, cada nó da árvore tem atribuído a si um “value” e uma “key”. A Arvore é organizada pelo valor da “key”, sendo num dado nó a

```
graph TD; Root[Root] --> K1[Key]; K1 --> K2[Key]; K1 --> K3[Key]; K2 --> K4[Key]; K2 --> K5[Key]; K3 --> K6[Key];
```

The diagram illustrates a B+ tree structure. The Root node points to a Key node, which points to two Key nodes. The left Key node points to two Key nodes, and the right Key node points to one Key node. Each node contains a Key, a Value, and pointers to Left and Right nodes.

-
- String
- StringT9
- Dictionary
- Small
- big Words
- HashMap
- BSTreeMap
- Head
- IC
- Str
- Linked List Map

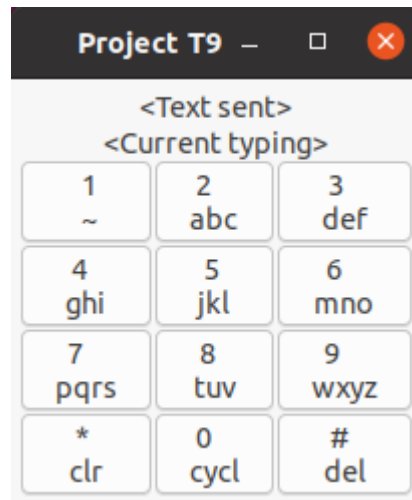
Estrutura geral do programa

O programa é constituído por 1 ficheiro 'Main.c' e demais ficheiros '.h' em que são implementadas as estruturas de dados, sendo as assinaturas das funções e structs declarados no ficheiro 'util.h'. No Main.c, temos as funções que inicializam o programa, criando a interface gráfica usando GTK+ e preenchendo o Dictionary com as palavras guardadas em ficheiro de texto, e implementando as demais funções necessárias para interação do usuário.

No ficheiro 'LinkedListMap.h', implementamos a Linked List Map usando valores int para as "key" e String para "value". Essa estrutura é usada no "value" da Binary Search Tree Map, implementada no ficheiro 'BSTreeMap.h', e sendo a "key" uma StringT9. E no ficheiro 'HashMap.h', usamos também um int como "key" e uma Binary Search Tree Map como "value". E por fim, a estrutura Dictionary definida em 'Dictionary.h' contém um Hash Map e uma Binary Search Tree Map.

Breve explicação de como o programa de executa

Quando executado o programa, este lê todas as palavras do ficheiro 'dictionary.txt' e insere elas em um Dictionary. Em seguida é criada uma interface, através da biblioteca do GTK+, contendo 12 botões e duas caixas de texto, sendo uma para a palavra que estamos a escrever e a outra para o texto já escrito.



Após a criação da interface, o programa aguarda que o utilizador introduza a sequência de números correspondentes a palavra que pretende escrever. Se o usuário quiser limpar o campo de palavra a ser escrita, pode apertar a tecla ' * ', ou se quiser apenas deletar o último número pode pressionar a tecla ' # '. Quando tiver escrito a palavra que pretende o usuário deve carregar na tecla '0'. Assim, enviaremos essa StringT9 para pesquisar o Dictionary e receberemos uma Linked List Map, a que iremos iterar para mostrar as opções de tradução no atual dicionário. Quando aparecer a palavra pretendida pelo usuário, é preciso carregar na tecla '1' para enviar para o texto completo. Se o usuário pressionar '1' antes de pressionar '0', a palavra mais utilizada será enviada ou, se não houver tradução no dicionário, o campo de palavra a ser escrita será limpo. Além disso, palavras enviadas são escritas num ficheiro 'fullText.txt' para persistência após o fechamento do programa e a sua frequência de utilização é incrementada no dicionário, escrevendo ao final do 'dictionary.txt'.