



Laboratório de Programação

Projecto II

Entrada de texto preditiva (T9)

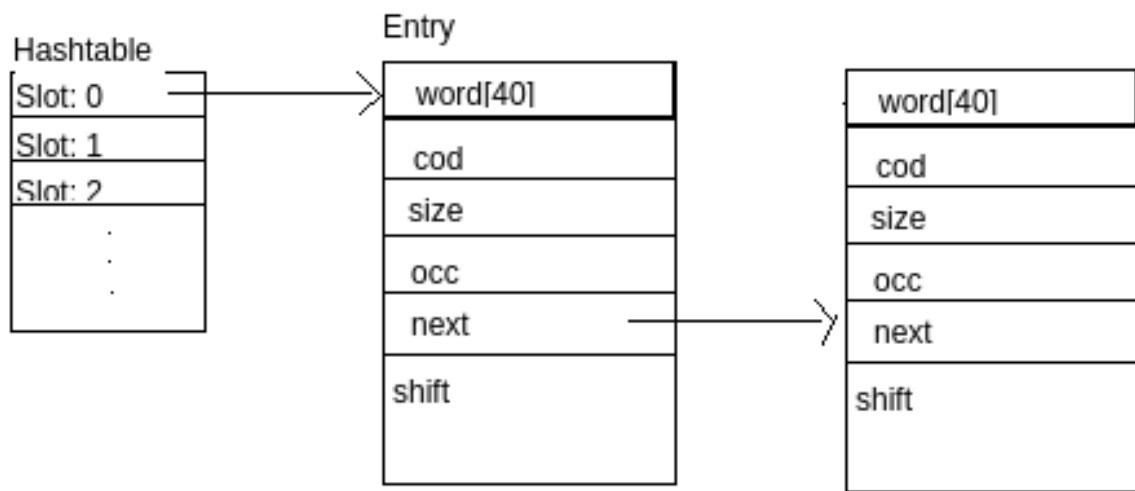
Resumo

Este trabalho tem como objetivo a implementação de um algoritmo de texto preditivo, usado pelos telemóveis antigos de 12 teclas onde cada tecla pode escrever três ou quatro letras, consoante o número de vezes que é pressionada.

A implementação deste algoritmo tira proveito das hashtables, que permitem um acesso aos dados mais eficientes, e listas ligadas que servem para armazenar objetos semelhantes sequencialmente nas entradas da hashtables.

O texto preditivo é gerado através do acesso a uma hashtable, onde foram guardados por codificação t9, e por ordem de ocorrência as palavras lidas a partir de um ficheiro de texto.

Esquema das estruturas de dados utilizadas



Como referido anteriormente, neste trabalho foram usadas hashtables e linked lists.

A hashtable é definida como um struct *ht_t* constituído por:

- Um array 2d de entradas, onde cada entrada tem um número associado gerado por uma hash function.

Cada entrada é definida por um struct *entry_t* constituído por:

- Um array de caracteres (*word*), para armazenar a palavra.
- Um array de inteiros para representar a codificação t9 da palavra (*cod*).
- O número de ocorrências de cada palavra (*occ*).
- Um inteiro *shift*, para resolver colisões.
- Um apontador (*next*), que aponta para a seguinte palavra, que na sua essência é uma lista ligada.

Estrutura geral do programa

O programa está dividido em 3 partes, contagem e leitura de palavras, processamento das palavras e inserção numa hashtable e por fim a parte de introdução de palavras pelo utilizador no dicionário, tal como a previsão de texto inserido pelo utilizador usando a codificação t9.

Contagem e leitura de palavras: O programa recebe como argumento um ficheiro de texto, conta as palavras, assumindo que cada palavra está separada por um caracter que não é uma letra pertencente ao alfabeto português, e transforma-as em caracteres minúsculos para serem adicionados à hashtable.

Inserção na hashtable: Para cada palavra a ser inserida na hashtable, é criada uma cópia onde os caracteres especiais (‘à’. ‘ê’, ‘ç’ ...) são transformados em caracteres normais, sem acentos ou cedilhas. Esta cópia é então usada para obter a codificação t9. A codificação t9 é usada na hash function que gera um número inteiro, que depois é usado como slot na hashtable.

Na inserção começamos por verificar se a slot gerada pela hash function está livre, se estiver, copiamos os dados da palavra (palavra não normalizada, codificação t9 e tamanho) a ocorrência é definida como 1, pois é a primeira ocorrência da palavra (a hash function gera um valor único para cada codificação t9). Caso a slot esteja ocupada, temos dois casos possíveis, estamos na slot certa, ou seja, a codificação é a mesma da palavra já existente na slot, e então é inserida como next da palavra já existente, ou no caso já estar inserida na hashtable, é incrementada a sua ocorrência, e corrigida a sua posição na lista se necessário, ou, estamos na slot errada, temos uma colisão, pelo que, incrementamos a slot, até encontrar uma slot vazia ou a slot correcta da palavra, que é verificado comparando os valores de shift de cada entrada, que são incrementados juntamente com a slot, e a codificação t9.

Interação com utilizador: Após inseridas as palavras na hashtable, o utilizador tem a sua primeira interação com o programa, onde tem duas opções:

Inserir uma palavra nova na hashtable, onde é levado para uma página onde tem que escrever manualmente a palavra, ou seja, carregar mais do que uma vez em cada tecla para chegar à letra desejada, e após inserida a palavra nova na tabela, regressa ao menu anterior

A segunda opção é escrever texto de forma inteligente, onde o utilizador escreve a palavra que utilizando a codificação t9, onde é sugerido as várias palavras presentes no dicionário, caso não exista, é exibida uma mensagem que diz que a codificação inserida não existe no dicionário. É ainda pedido ao utilizador após ter inserido a codificação se quer continuar a escrever, caso contrário é apresentada a mensagem escrita separada por espaços.

Nota: A interface gráfica não foi implementada.

Estrutura dos ficheiros de armazenamento dos dicionários

O dicionário de cada execução é armazenado num ficheiro de texto com o nome dict.txt, seguindo a seguinte codificação:

slot-cod.size-word.occ-word.occ ... -word.occ-word.occ \$

Onde:

- slot é o um número inteiro que representa a entrada da tabela onde estão guardadas as palavras.
- cod é um conjunto de inteiros que representa a codificação t9 das palavras.
- size é um número inteiro que representa o tamanho da codificação.
- word é um conjunto de caracteres que representam a palavra.
- occ é um inteiro que representa as ocorrências da palavra.

Execução do programa

Para compilar o programa: gcc -g -Wall -o hash hash.c

Para executar o programa: ./hash ficheiro_de_texto

Quando executado aparece um menu onde é possível escolher entre introduzir uma palavra no dicionário, ou escrever usando escrita preditiva.

```
Pretende escrever ou inserir uma nova palavra no dicionário ?
1. Inserir uma nova palavra
2. Escrever
3. Sair
```

Se for escolhido 1:

```
1
+-----+
| 1 | 2 | 3 |
|   | a b c | d e f |
+-----+
| 4 | 5 | 6 |
| g h i | j k l | m n o |
+-----+
| 7 | 8 | 9 |
| p q r s | t u v | w x y z |
+-----+
|   | 0 | ^ |
| * + | _ _ | # |
+-----+
6665552
Adicionado 'ola' ao dicionário, enter para continuar
█
```

Se for escolhido 2:

```
2
+-----+
| t9 predictive test |
| Introduza os numeros |
+-----+
| 's' para sair |
+-----+
| 'Enter' termina o |
| input da palavra |
+-----+
| '1' para ver as |
| diferentes sugestões |
+-----+
| 'e' para escolher |
| a palavra |
+-----+
767
1: por
2: pôs
3: pôr
4: sós
2
Se não quiser escrever mais escreva s, caso contrário carregue enter
s
pôs
```