Projeto II - relatório

v1.0

Gerado por Doxygen 1.8.20

Capítulo 1

Trabalho de Implementação II - Identificação de prefixos e Indexação de dicionários

Objetivo

Este trabalho consiste na construção e utilização de estrutura hierárquica denominada **trie** (do inglês "re**tri**eval", sendo também conhecida como **árvore de prefixos** ou ainda **árvore digital**) para a indexação e recuperação eficiente de palavras em grandes arquivos de dicionários (mantidos em memória secundária). A implementação deverá resolver dois problemas (listados a seguir), e os resultados deverão ser formatados em saída padrão de tela de modo que possam ser automaticamente avaliados no VPL.

A figura a seguir exemplifica a organização de um arquivo de dicionário. Cada linha apresenta a definição de uma palavra, sendo composta, no início, pela própria palavra com todos os caracteres em minúsculo (somente entre 'a' (97) e 'z' (122) da tabela ASCII) e envolvida por colchetes, seguida pelo texto de seu significado. Não há símbolos especiais, acentuação, cedilha, etc, no arquivo.

Primeiro problema: identificação de prefixos

Construir a trie, em memória principal, a partir das palavras (definidas entre colchetes) de um arquivo de dicionário, conforme o exemplo acima. A partir deste ponto, a aplicação deverá receber uma série de palavras quaisquer (pertencentes ou não ao dicionário) e responder se trata de um prefixo (a mensagem '**is prefix**' deve ser produzida) ou não (a mensagem '**is not prefix**' deve ser produzida na saída padrão). Sugestão de nó da trie:

```
NoTrie {
    char           letra;        //opcional
    NoTrie        *filhos[26];   //pode ser uma 'LinkedList' de ponteiros
    unsigned long  posição;
    unsigned long  comprimento;  //se maior que zero, indica último caracter de uma palavra
}
```

Segundo problema: indexação de arquivo de dicionário

A contrução da trie deve considerar a localização da palavra no arquivo e o tamanho da linha que a define. Para isto, ao criar o nó correspondente ao último caracter da palavra, deve-se atribuir a **posição do caracter inicial** (incluindo o abre-colchetes '['), seguida pelo **comprimento da linha** (não inclui o caracter de mudança de linha) na qual esta palavra foi definida no arquivo de dicionário. Caso a palavra recebida pela aplicação exista no dicionário, estes dois inteiros devem ser produzidos. **Importante**: uma palavra existente no dicionário também pode ser prefixo de outra; neste caso, o caracter final da palavra será encontrado em um nó não-folha da trie e também deve-se produzir os dois inteiros (posição e comprimento) na saída padrão.

Capítulo 2

Índice dos namespaces

2.1 Lista de namespaces

Lista dos namespaces com uma breve descrição:

Capítulo 3

Índice dos componentes

3.1 Lista de componentes

Lista de classes, estruturas, uniões e interfaces com uma breve descrição:

Capítulo 4

Índice dos ficheiros

4.1 Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

Capítulo 5

Documentação dos namespaces

5.1 Referência ao namespace structures

Nó de uma Trie.

Componentes

- struct NoTrie

Funções

- struct NoTrie ∗ initi ()
- void insert (struct NoTrie ∗root, std::string key, unsigned long pos, unsigned long size)

5.1.1 Descrição detalhada

Nó de uma Trie.

5.1.2 Documentação das funções

5.1.2.1 initi()

```
struct NoTrie* structures::initi ( )
```

5.1.2.2 insert()

```
void structures::insert (
             struct NoTrie * root,
             std::string key,
             unsigned long pos,
             unsigned long size )
```

Capítulo 6

Documentação da classe

6.1 Referência à estrutura structures::NoTrie

`#include <trie.h>`

Atributos Públicos

- struct NoTrie * filhos [26]
- unsigned long size
- unsigned long pos

6.1.1 Descrição detalhada

Definido na linha 9 do ficheiro trie.h.

6.1.2 Documentação dos dados membro

6.1.2.1 filhos

`struct NoTrie* structures::NoTrie::filhos[26]`

Definido na linha 10 do ficheiro trie.h.

6.1.2.2 pos

`unsigned long structures::NoTrie::pos`

Definido na linha 11 do ficheiro trie.h.

Referenciado por outputResults().

6.1.2.3 size

```
unsigned long structures::NoTrie::size
```

Definido na linha 11 do ficheiro trie.h.

Referenciado por outputResults().

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- trie.h

Capítulo 7

Documentação do ficheiro

7.1 Referência ao ficheiro dic.h

```
#include "trie.h"
```

Funções

- NoTrie ∗ parseDict (std::string filename)

  Gera uma Trie a partir de um arquivo de dicionário.

Variáveis

- int nextPos = 0

7.1.1 Documentação das funções

7.1.1.1 parseDict()

```
NoTrie* parseDict (
            std::string filename )
```

Gera uma Trie a partir de um arquivo de dicionário.

Parâmetros

string	Caminho para o arquivoo

Retorna

Nó raiz da trie gerada pelo arquivo

Definido na linha 16 do ficheiro dic.h.

Referenciado por main().

7.1.2 Documentação das variáveis

7.1.2.1 nextPos

```
int nextPos = 0
```

Definido na linha 6 do ficheiro dic.h.

Referenciado por parseDict().

7.2 Referência ao ficheiro main.cpp

```
#include <fstream>
#include <iostream>
#include <string>
#include "trie.h"
#include "dic.h"
```

Funções

- void outputResults (structures::NoTrie *node, std::string word)

  Printa a resposta correspondente ao nó e à palavra buscada.
- int main ()

  Executa o programa, e lê as entradas.

7.2.1 Documentação das funções

7.2.1.1 main()

```
int main ( )
```

Executa o programa, e lê as entradas.

Retorna

0 se o programa funcionar;

1 se não foi possível abrir o arquivo;

Definido na linha 37 do ficheiro main.cpp.

7.2.1.2 outputResults()

```
void outputResults (
            structures::NoTrie * node,
            std::string word )
```

Printa a resposta correspondente ao nó e à palavra buscada.

Parâmetros

NoTrie	nó correspondente a palavra buscada
string	palavra buscada

Definido na linha 15 do ficheiro main.cpp.

Referenciado por main().

7.3 Referência ao ficheiro README.md

7.4 Referência ao ficheiro trie.h

```
#include "trie.inc"
```

Componentes

- struct structures::NoTrie

Namespaces

- structures

  Nó de uma Trie.

Funções

- struct NoTrie ∗ structures::initi ()
- void structures::insert (struct NoTrie ∗root, std::string key, unsigned long pos, unsigned long size)

7.5 Referência ao ficheiro trie.inc

```
#include <iostream>
```

Funções

- struct NoTrie ∗ init ()

  Cria um NoTrie.
- void insert (struct NoTrie ∗root, std::string key, unsigned long pos, unsigned long size)

  Insere valores em uma Trie.
- NoTrie ∗ find (struct NoTrie ∗root, std::string key)

  encontra o último nó de uma palavra de uma Trie.
- int countSufixes (struct NoTrie ∗root)

  Conta os sufixos de um nó.

7.5.1 Documentação das funções

7.5.1.1 countSufixes()

```
int countSufixes (
            struct NoTrie * root )
```

Conta os sufixos de um nó.

Parâmetros

NoTrie	raiz da trie.

Retorna

Quantidade de sufixos de uma palavra no dicionário.

Definido na linha 78 do ficheiro trie.inc.

Referenciado por outputResults().

7.5.1.2 find()

```
NoTrie* find (
             struct NoTrie * root,
             std::string key )
```

encontra o último nó de uma palavra de uma Trie.

Parâmetros

NoTrie	raiz da trie
string	palavra

Retorna

nó desejado

Definido na linha 55 do ficheiro trie.inc.

Referenciado por main().

7.5.1.3 init()

```
struct NoTrie* init ( )
```

Cria um NoTrie.

Retorna

Nó inicializado.

Definido na linha 8 do ficheiro trie.inc.

Referenciado por find(), insert() e parseDict().

7.5.1.4 insert()

```
void insert (
             struct NoTrie * root,
             std::string key,
             unsigned long pos,
             unsigned long size )
```

Insere valores em uma Trie.

Parâmetros

NoTrie	raiz da trie
string	palavra
long	posição no dicionário
long	tamanho da linha

Definido na linha 30 do ficheiro trie.inc.

Referenciado por parseDict().

Índice