

Construa os programas e as bibliotecas indicados na Parte II, usando a linguagem C, tendo o cuidado de eliminar repetições no código fonte e de isolar funcionalidades distintas em diferentes ficheiros fonte. Entregue o código desenvolvido, devidamente indentado e comentado, bem como os *makefile* para gerar os executáveis e as bibliotecas a partir do código fonte. Assegure-se de que o compilador não emite qualquer aviso sobre o seu código com a opção `-Wall` activa, e de que no final da execução do programa não existem recursos por libertar (memória alocada dinamicamente e ficheiros abertos). Deve verificar essa situação com o utilitário Valgrind. Em caso de erro irreversível, o programa não deve terminar descontroladamente, deve, no mínimo, emitir uma mensagem informativa e em seguida terminar. O código desenvolvido será valorizado segundo os seguintes critérios, em importância decrescente: correção, eficiência, clareza.

Na realização dos programas propostos deve utilizar a biblioteca de uso genérico [Glib-2.0](#), disponibilizada pela fundação [Gnome](#) em [código aberto](#) de [utilização livre](#).

---

## Parte I - Preparação do ambiente de desenvolvimento

### Valgrind

Para verificar se um programa liberta toda a memória alocada dinamicamente poderá utilizar a ferramenta **valgrind**.

Instalação: `$ sudo apt install valgrind`

Documentação: `$ man valgrind`

Utilização: `$ valgrind <nome do executável> <lista de argumentos do executável>`

### Glib

Instalação: `$ sudo apt install libglib2.0-dev`

Documentação: [Glib-2.0](#)

Utilização: [Compiling Glib Applications](#)

### Dicionário

O [Projeto Natura](#), da Universidade do Minho, coloca disponíveis recursos para o processamento da linguagem natural.

Na página [Dicionários](#), na secção Outros Recursos, dá acesso a [ficheiros de texto](#) contendo uma extensa lista de palavras que pode ser utilizada como dicionário.

---

## Parte II - Realização

Pretende-se realizar um programa para verificação ortográfica de textos. O programa recebe em parâmetros os dicionários e os textos a verificar. O resultado da verificação é uma lista das palavras não encontradas.

1. Realização da representação de um dicionário e das respectivas operações. A representação do dicionário deve basear-se num contentor que privilegie a operação de procura de palavras. Considere a utilização de um dos seguintes tipos de dados, pertencentes à biblioteca Glib: HashTable, Tree ou PtrArray.

**typedef struct Dictionary { ... } Dictionary;** - tipo opaco para representação de um dicionário;

**Dictionary \*dictionary\_create()** - Criar e inicializar uma variável struct Dictionary;

**void dictionary\_add(Dictionary \*dictionary, const char \*filename)** - Adicionar ao dicionário o conjunto de palavras presentes no ficheiro indicado;

**int dictionary\_lookup(Dictionary \*dictionary, const char \*word)** - Verifica se o dicionário contém a palavra indicada;

**void dictionary\_destroy(Dictionary \*dictionary)** - Liberta os recursos associados ao dicionário;

- a. Defina o tipo Dictionary e programe as funções indicadas acima, de acordo com as definições dadas.
  - b. Realize um programa de teste das funções anteriores que receba como argumentos um dicionário e uma palavra e verifique se existe no dicionário. No âmbito deste exercício, crie um *makefile* para geração do executável.
  - c. Crie um *makefile* para gerar a biblioteca na versão de ligação estática e na versão de ligação dinâmica.
  - d. Teste o funcionamento de ambas as versões da biblioteca modificando o programa realizado em b.
2. Construa um programa que verifique a correção ortográfica de um texto e apresente a lista das palavras que não constam dos dicionários. A cada palavra não encontrada deve ser associada uma lista de coordenadas do local do texto onde ocorrem. As coordenadas são formadas por pares linha - coluna. As palavras devem ser apresentadas pela ordem da primeira ocorrência no texto.

Sintaxe de invocação do programa:

**\$ spell\_check <option>**

Opções:

**-t <nome de ficheiro>** - indicação do ficheiro com o texto para verificar;

**-w <palavra>** - verificar palavra isolada;

**-d <nome de ficheiro>** - indicação do ficheiro com a lista de palavras que contribuem para o dicionário.

O programa deve aceitar várias ocorrências da opção **-d**, mas não deve aceitar simultaneamente as opções **-t** e **-w**.

Exemplo de invocação:

**\$ spell\_check -d português.txt -d estrangeirismos.txt -t relatório.md**

Data limite de entrega: 2 de janeiro de 2024

ISEL, 28 de novembro de 2023