

Resumo

Este trabalho foca-se os conceitos básicos do funcionamento GAWK e das expressões regulares utilizadas para descrever padrões. Neste relatório descrevemos as decisões tomadas e as dificuldades encontradas bem como pequenos exemplos, para que qualquer um que o leia perceba facilmente como funciona o nosso projeto.

Conteúdo

1	Introdução	2
1.1	Um belo Projeto	2
2	Ficheiros de internacionalização	3
2.1	Descrição informal do problema	3
2.2	Especificação do Requisitos	3
3	Concepção/desenho da Resolução	4
3.1	Estruturas de Dados	4
3.2	Algoritmos	4
4	Codificação e Testes	5
4.1	Problemas de implementação, Decisões e Alternativas	5
4.2	Problemas de implementação	5
4.3	Decisões	5
4.4	Alternativas	5
4.5	Testes realizados e Resultados	5

Capítulo 1

Introdução

Uma parte importante do sistema linux são as expressões regulares, a capacidade de procurar num input um determinado padrão facilita muitas das operações.

1.1 Um belo Projeto

Enquadramento bla bla bla bla

Conteúdo do documento ble ble ble ble

Resultados – pontos a evidenciar bli bli bli bli

Estrutura do documento blo blo blo

letras gregas são estas $\alpha\beta\gamma\delta$ que aqui demonstro
exemplo simples de fração

$$\frac{\frac{a*b+c}{4-3}}{3*5}$$

simples

Mais exemplos de listas enumeradas mas agora com letras:

- a) Listar todas as Pessoas identificadas, sem repetições;
- b) Listar os Países e Cidades marcadas;
- c) Listar as Organizações.

A mesma enumeração mas no standard numérico

Etape 1: Listar todas as Pessoas identificadas, sem repetições;

Etape 2: Listar os Países e Cidades marcadas;

Etape 3: Listar as Organizações.

Capítulo 2

Ficheiros de internacionalização

2.1 Descrição informal do problema

É necessária um "script" que permita:

- a) Analisar vários ficheiros de formato PO, e devolver o número de traduções e os seus tradutores, bem como alguns metadados acharmos relevantes.
- b) Através de ficheiros PO's criar dicionários de triangulação de Português para Francês.
- c) Reformatar ficheiros de formato PO pra LaTeX.

2.2 Especificação do Requisitos

Os requisitos mínimos deste tarbalho são saber utilizar Linux, bash e GAWK.

Capítulo 3

Concepção/desenho da Resolução

3.1 Estruturas de Dados

3.2 Algoritmos

Utilizamos as seguintes expressões regulares:

- i) /msgid/
- ii) /msgid[]*/
- iii) /"Language: /
- iv) /i.*@.*i/ && !/"Report/ && !/FIRST/ && !/"Language/
- v) /"Language-Team: *
n"/
- vi) /"Language-Team: /
- vii) /portugu[êe]se?s?/
- viii) /msgid[]*/
- ix) /msgstr[]*/
- x) /msgid */
- xi) /msgstr */

Capítulo 4

Codificação e Testes

4.1 Problemas de implementação, Decisões e Alternativas

4.2 Problemas de implementação

Em geral, no trabalho encontramos problemas com a formatação de alguns ficheiros, pois não estavam na formatação prevista, e tivemos alguns problemas com o LaTeX.

4.3 Decisões

No problema 1 proposto no exercício, decidimos ignorar os "msgid" que não contivessem nenhuma mensagem ou que contivessem mais de uma tradução. Decidimos ignorar os "msgid" que não contivessem nenhuma mensagem ou que contivessem mais de uma tradução. Escolhemos como metadados mais importantes a linguagem, a percentagem de blocos, quantos blocos tem o ficheiro e quantos e quem são os tradutores. Decidimos também que a linguagem só está no "Language". No problema 2 decidimos utilizar a "Language-team" para definir a linguagem de tradução, pois nenhum dos ficheiros tinha linguagem no campo "Language". Usamos uma matriz descritiva sendo as linhas a frase em inglês e os descritivos das colunas a linguagem portuguesa ou francesa. Assumimos só a linguagem portuguesa e a francesa e imprimimos unicamente o que tem tradução em triangulação. No último problema ignoramos o "n" e o "#", devido ao LaTeX, substituindo por e por -, respetivamente.

4.4 Alternativas

Uma alternativa era modificar os ficheiros manualmente de modo a estar no formato certo, outra era fazer inúmeros casos especiais.

4.5 Testes realizados e Resultados

Mostram-se a seguir alguns testes feitos (valores introduzidos) e os respectivos resultados obtidos:

```
public class Aula()
{
    int n, m;
    int max(int a, int b)
    {
        .....
        return(max);
    }
}
```

}