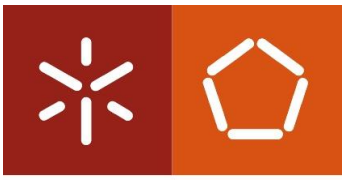


Universidade do Minho

Departamento de Informática

Processamento de Linguagens
Trabalho Prático nº 1 (FLex)

Gonçalo Almeida (A84610)

Emanuel Rodrigues (A84776)

Lázaro Pinheiro (A86788)

5 de Abril de 2020

Conteúdo

Capítulo 1

Introdução

Este projeto foi realizado no âmbito da unidade curricular Processamento de Linguagens e tem como objetivo o parse de um ficheiro HTML que contém comentários a uma notícia publicada no jornal Público para um ficheiro JSON com as chaves id, user, date, timestamp, commentText, likes, hasReplies e numberOfReplies e os seus valores associados. Para isto foi utilizada a ferramenta Flex de modo a gerar os filtros de texto necessários e a biblioteca Glib para o armazenamento dos conteúdos de cada comentário.

Capítulo 2

Objetivos do trabalho prático

- Aumentar a experiência de uso do ambiente Linux e de algumas ferramentas de apoio à programação;
- Aumentar a capacidade de escrever Expressões Regulares (ER) para descrição de padrões de frases;
- Desenvolver, a partir de ERs, sistemática e automaticamente Processadores de Linguagens Regulares, que filtrem ou transformem textos com base no conceito de regras de produção Condição-Ação;
- Utilizar o Flex para gerar filtros de texto em C.

Capítulo 3

Notas iniciais

Como o ficheiro inicial HTML fornecido encontrava-se com um encoding que usa caracteres de 8bits foi necessário convertê-lo para UTF8 antes de o processar de modo a que os caracteres acentuados sejam válidos. Para isto foi utilizado o seguinte comando:

```
$  iconv -f cp1252 -t utf8 Publico_extraction_portuguese_comments_4.html > publico-utf8.html
```

Como o ficheiro HTML não tem referência ao número de likes de um comentário, foi-nos indicado para colocar este valor a 0 em todas as ocorrências de comentários.

Capítulo 4

Implementação

4.1 Estrutura de dados

Com a ajuda da biblioteca Glib definimos as seguintes estruturas, a primeira para armazenar o conteúdo processado de um comentário e a segunda para armazenar as estruturas anteriores:

```
typedef struct comment
{
    GString* id;
    GString* user;
    GString* date;
    gint timestamp;
    GString* commentText;
    gboolean hasReplies;
    gint numberOfReplies;
    GArray* replies;
}* Comment;

GArray* commentThread;
```

A estrutura comment contém todos os campos pedidos para o formato JSON exceto o campo likes, que foi razões anteriormente referidas não precisa de ser guardado em memória.

A estrutura commentThread é um array de elementos arbitrários, neste caso de estruturas comment, que cresce automaticamente sendo os elementos acrescentados no fim desta.

4.2 Criação do ficheiro JSON

Após ter o conteúdo do ficheiro HTML formatado e em memória nas estruturas criadas, este é escrito num ficheiro TXT que serve como ficheiro intermédio. De seguida é feita uma system call no step1.fl para compilar e executar o step2.fl, dando como entrada o ficheiro gerado anteriormente e escrevendo no ficheiro final JSON.

4.3 Filtros de Texto Flex

De modo a processar o ficheiro de entrada desenvolvemos um ficheiro Flex step1.fl que faz o parse de todos os campos pedidos para um ficheiro TXT. Após esse processamento desenvolvemos um outro ficheiro Flex step2.fl que coloca o carácter '\' antes das aspas dos campos commentText do ficheiro resultante do passo anterior para formatar corretamente o ficheiro JSON final. Para os

campos hasReplies e numberOfReplies não utilizamos expressões regulares pois estes são obtidos através das estruturas criadas.

4.3.1 Id

Para filtrar o id do comentário decidimos capturar os 6 caracteres (letras minúsculas ou dígitos) após a expressão -t"id=". Vimo-nos obrigados a incluir o -t"pois existem duas ocorrências da expressão id= mas apenas uma se refere ao pretendido.

4.3.2 User

Para filtrar o user do comentário decidimos capturar todo o texto entre as expressões rel="nofollow"> e , excluindo a última parte. De modo a capturar também quando uma conta é desativada capturamos, em alternativa, todo o texto entre as expressões Conta e \r, excluindo a última parte.

4.3.3 Date

Para filtrar a data do comentário decidimos capturar a sequência de texto que contém dígitos e os caracteres espaço, ponto e dois pontos após a expressão permalink».

4.3.4 Timestamp

Para filtrar o timestamp do comentário decidimos capturar a sequência de texto que contém dígitos e os caracteres backslash, traço, 'T', dois pontos e ponto entre as expressões datetime="e ">, excluindo a última parte. De seguida é chamada a função dateToTimestamp que processa o texto capturado para o formato do timestamp do Linux.

4.3.5 CommentText

Para filtrar o texto do comentário tivemos em conta que o miolo se encontra entre as expressões \r\n e . Para isto definimos uma condição de contexto TEXT em que entramos nela quando encontramos a primeira expressão e saímos quando encontramos a segunda. Deste modo capturamos todo esse texto de forma a não ser processado indevidamente por outra expressão regular e também ignoramos os \n presentes.

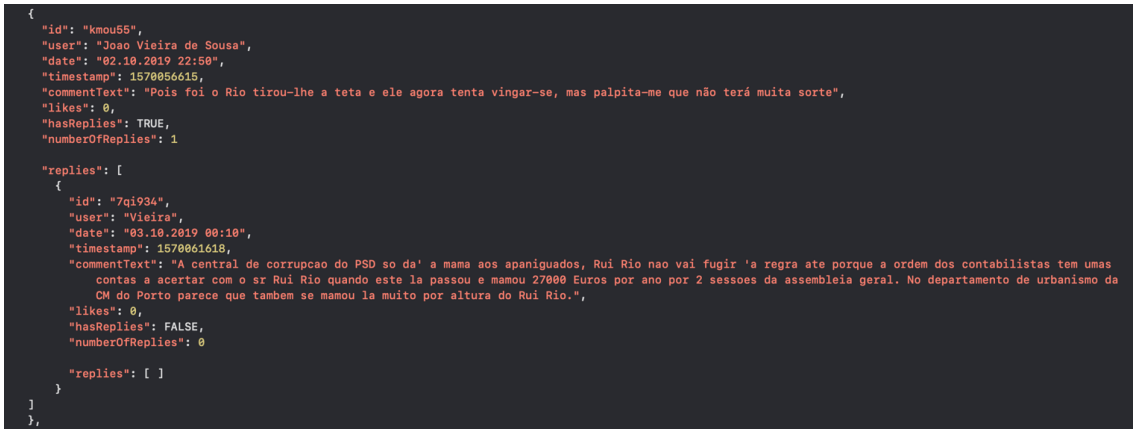
4.3.6 Ficheiro step2.fl

Para que o ficheiro JSON esteja corretamente formatado foi necessário colocar um backslash antes de cada aspa do commentText. Para isto criamos uma condição de contexto TEXT de modo a não processar as aspas que não fazem parte do campo. Ao entrar na condição quando apanhamos a expressão "commentText": "e saindo quando apanhamos a expressão ",\n\t garantimos que apenas apanhamos as aspas pretendidas. Dentro da condição, sempre que uma aspa é apanhada, é escrito o backslash antes desta.

Capítulo 5

Resultado Final

Após processar todo o ficheiro HTML obtemos um ficheiro JSON com estruturas como a representada na figura abaixo.



Capítulo 6

Conclusão

A realização deste trabalho prático permitiu-nos compreender a grande utilidade da ferramenta Flex no tratamento de informação. Encontramos dificuldades com a formatação do ficheiro inicial, sendo que não reconhecia os caracteres acentuados, mas com a ajuda dos docentes ultrapassamos esse obstáculo. Em suma, obtivemos uma melhor compreensão e prática relativamente às expressões regulares e consideramos que cumprimos todos os objetivos estipulados.