Processamento de Linguagens e Compiladores
(3º ano de LCC - Pedro Rangel Henriques)

Trabalho Prático Nº1 (FLex)

2.3 Pré-Processador para LaTeX
Relatório de Desenvolvimento

Francisca Fernandes (A72450)
Francisco Oliveira (A82066)
Maria Luísa Silva (A82124)

25 de Julho de 2021

Resumo

Isto é um resumo do relatório de *Processamento de Linguagens e Compiladores* focado no contexto do trabalho pré-processador para LaTeX, os objetivos concretos e os resultados, tanto como as dificuldades atingidos por nós.

Conteúdo

Capítulo 1

Introdução

O trabalho prático 2.3 Pré-Processador para LaTeX, surge inserido no âmbito da disciplina de *Processamento de Linguagens e Compiladores*. É o primeiro trabalho prático desta unidade curricular, abordando o uso da ferramenta **FLex**.

Ao longo deste documento iremos abordar o tema, assim como as escolhas feitas para o resolver, problemas/dificuldades que tivemos na sua realização.

Capítulo 2

Análise e Especificação

2.1 Descrição informal do problema

Este trabalho consiste em criar um processador capaz de expandir abreviaturas da linguagem LaTeX.

Estas abreviaturas são decididas pelo grupo de forma a que a edição do documento LaTeXseja mais eficiente.

É depois necessário criar um filtro de texto que faça o reconhecimento das abreviaturas criadas e as converta em linguagem usual LaTeX, recorrendo ao Gerador **FLex**.

2.2 Especificação dos Requesitos

2.2.1 Linguagem Escolhida

Negrito
\n{texto}

Itálico
\it{texto}

Sublinhado
\s{texto}

Capítulo
\chap {texto}

Secção
\sec {texto}

Subsecção
\ssec{texto}

Nova Página
\newp

Lista de Tópicos Não Numerados

```
\ul
    * Item1
    * Item2
    ..
/ul
```

Lista de Tópicos Numerados

```
\ol
    * Item1
    * Item2
/ol
```

Texto centrado

```
\c
    Texto Centrado
/c
```

2.2.2 Pedidos

Neste trabalho, é pedida a criação de um tipo de anotação mais simples,que posteriormente, vai ser transformada em LaTeXatravés de um pré-processador criado com a ferramenta **FLex**.

Capítulo 3

Descrição Da Solução

Neste trabalho, usamos Expressões Regulares de forma a filtrar anotações num ficheiro de texto.

Submetemos o ficheiro de texto a um pre-processador desenvolvido por nós, de maneira a que as nossas anotações ficassem associadas a anotações específicas do LaTeX.

3.1 Uso de Stacks

Ao longo deste trabalho foi necessário utilizar stacks, devido a ser pedido no enunciado a utilização de listas de tópicos (items).

Desde cedo que nos deparamos com o problema de ter que expandir abreviaturas dentro de outras abreviaturas.

Para resolver esta situação, recorremos ao uso de uma stack utilizando as funções *yy_pop_state*() e *yy_push_state*().

Capítulo 4

Codificação e Testes

4.1 Alternativas, Decisões e Problemas de Implementação

A nossa principal dificuldade foi limpar o ficheiro dado, mais especificamente encontrar um record separator.

Ao fim de muita pesquisa e de várias tentativas erros acabamos por conseguir encontrar um que nos deu o resultado pretendido .

4.2 Testes realizados e Resultados

Para a implementação do Pré-Processador LaTeXutilizamos o seguinte texto anotado como exemplo.

```
A \n{UC} de \it{Processamento de Linguagens e Compiladores} tem inicialmente o seguinte \s{programa} :

\chap{I}


\sec{Introdução ao Processamento de Linguagens}


\ssec{Tópicos}
    \ul {
      * Conceitos básicos: Linguagem.
      * Gramáticas
        \ol{
            * Gramáticas Independentes
            * Gramáticas Regulares
            * Contexto Gramáticas Tradutoras.
       }/ol
    }/ul
```

4.2.1 Execução do programa, criação do LaTeX

Para a execução do programa, criação do LaTeXe geração do PDF criamos uma makefile com os comandos necessários.

4.2.2 Resultado do pré-processamento do texto de entrada, com as nossas anotações, dando origem a este código LateX:

```
A \textbf{UC} de \textit{Processamento de Linguagens e Compiladores} tem inicialmente o seguinte \underline{p

\chapter{I}


\section{Introdução ao Processamento de Linguagens}


\subsection{Tópicos}
    \begin{itemize}
        \item  Conceitos básicos: Linguagem
        \item  Gramáticas
        \begin{enumerate}
              \item  Gramáticas Independentes
              \item  Gramáticas Regulares
              \item  Contexto Gramáticas Tradutoras.
       \end{enumerate}
    \end{itemize}
```

Capítulo 5

Conclusão

Após a realização deste trabalho concluímos que o gerador **FLex** é uma ferramenta muito útil e bastante versátil para transformações de texto estruturado.

Além disso, percebemos melhor o seu funcionamento e o seu modo de aplicação.

Podemos afirmar que fazemos um balanço positivo deste trabalho e que conseguimos concretizar os objetivos pretendidos.

Apêndice A

Código do Programa

Encontra-se a seguir o código C/Flex que foi desenvolvido por nós para fazer o pré-processador para LaTeX.

```
1
2  %{
3  #include <string.h>
4  #include <stdlib.h>
5  %}
6
7
8  %s UNOREDERED_LIST ORDERED_LIST CENTER
9  %option stack
10
11
12 %%
13 \\n {printf("\\textbf");}
14
15 \\s {printf("\\underline"); }
16
17 \\it {printf("\\textit"); }
18
19 ^\\sec {printf("\\section"); }
20
21 ^\\ssec {printf("\\subsection"); }
22
23 ^\\chap {printf("\\chapter"); }
24
25 \\newp {printf("\\newpage");}
26
27 \* {printf("  \\item %s", yytext + 1);}
28
29 \\ol[  | \n]*\{ {printf("\\begin{enumerate}");yy_push_state(ORDERED_LIST);}
30 \\ul[  | \n]*\{ {printf("\\begin{itemize}");yy_push_state(UNOREDERED_LIST);}
31
32 <UNOREDERED_LIST>\}"/ul" {printf("\\end{itemize}");yy_pop_state();}
33 <ORDERED_LIST>\}"/ol" {printf("\\end{enumerate}");yy_pop_state();}
34
35 \\c[  | \n]*\{ {printf("\\begin{center}");yy_push_state(CENTER);}
36 <CENTER>\}"/c" {printf("\\end{center}");yy_pop_state();}
37
38
39 (.|\n) { ECHO; }
40
```

```
41 %%
42
43 int yywrap(){
44         return 1;
45 }
46
47 int main(){
48         yylex();
49         return 0;
50 }
```