

Pré-Processador de Texto para HTML e/ou L^AT_EX

Diogo Alves - A61030
Ricardo Branco - A61075
Hélder Gonçalves - A61084

15 de Abril de 2013

Conteúdo

1	Introdução	2
2	Funcionamento do PPLH	3
2.1	Syntax	3
2.2	Inicialização do programa	3
3	Estrutura do Trabalho	4
3.1	<i>MAN</i>	5
3.2	Modulo do Parser	6
3.2.1	Inicializações	6
3.2.2	Parser	6
3.2.3	Função <i>main</i>	6
3.3	Modulo do Processador	6
3.4	Modulo PPLH	7
3.5	Modulo de Lista Ligada	8
4	Conclusão	9

1 Introdução

Para o trabalho da Unidade Curricular de Processamento de Linguagens existe uma grande variedade de escolhas, da qual, decidiu-se fazer um processador de texto para HTML e/ou \LaTeX . No enunciado, é pedido para escolher apenas uma destas linguagens, mas com o objectivo final de melhorar o conhecimento sobre as duas linguagens em questão optou-se por criar um *processador* que fizesse a conversão para as duas linguagens.

2 Funcionamento do PPLH

O *PPLH* consiste num pré-processador de texto que gera ficheiros em *HTML* e/ou \LaTeX .

Este pré-processador funciona com ficheiros com extensão *.pplh*.

2.1 Syntax

Cada ficheiro *.pplh* contém um conjunto de comandos pré-definidos e que podem facilmente ser convertidos em comandos *HTML* e \LaTeX .

Comando	Descrição
$\backslash\text{t:}TEXTO$	Título do Documento
$\backslash\text{a:}TEXTO$	Nome do autor
$\backslash\text{s}[1-6]$	Secção com nível de 1 a 6
$\backslash\text{lo}$	Lista Ordenada
$\backslash\text{li}$	Lista com itens
$\text{->}TEXTO$	Item
$\backslash\text{indice}$	Índice
$\backslash\text{href}(URL,TEXTO)$	Hiperligação
$\backslash\text{fig}$	Modo de Imagem
$\backslash\text{img}(IMAGEM)$	Inserir Imagem
$\backslash\text{leg:}TEXTO$	Legenda da Imagem
$\backslash\text{s:Número}$	Escala da Imagem
$\backslash\text{tabela}$ -[lcr]>	Modo tabela Insere uma linha na tabela (l -> left, c -> center, r -> right)
$\backslash\text{l}$	Quebra de linha
$\backslash\text{latex}$	Texto que só aparece no documento \LaTeX
$\backslash\text{html}$	Texto que só aparece no documento <i>HTML</i>
$\backslash\text{bs}$	Inserir um Backslash
$\backslash\text{! }TEXTO$	Inserir um comentário
$\backslash\text{bie}$	Texto Formatado (b -> Bold, i -> italic, e -> emphase)
$\backslash\text{tnf}$	Modo Texto não formatado

Além dos comandos referidos na tabela existe um outro comando - $\backslash\text{end}$. Quando é detetado este comando, o *PPLH* termina o modo de texto atual e retorna para o modo de texto anterior.

2.2 Inicialização do programa

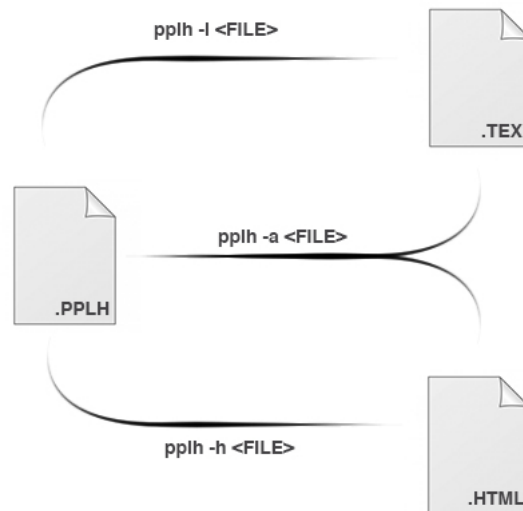


Figura 1: Inicialização do programa.

3 Estrutura do Trabalho

Para este trabalho, foram criados 4 módulos

1. Módulo do Parser

Módulo criado em *Flex* responsável por conter:

- A inicialização das estruturas de dados.
- As diferentes expressões regulares existentes.
- A função *main*.

2. Módulo do Processador de Texto

Este modulo é responsavel por converter a linguagem de programação que foi tratada pelo *Módulo de Parser* e convertê-la em *HTML* e em *L^AT_EX*

3. Modulo do PPLH

Este modulo é o responsável por escrever para ficheiros *.tex* e/ou *.html* a informação presente na estrutura de dados.

4. Modulo de Lista Ligada

Este modulo é utilizado para guardar na estrutura de dados o texto já processado, na versão final (HTML e/ou *L^AT_EX*).

3.1 *MAN*

Além disto também fez-se um *MAN* do comando *pplh*, a explicar como o usar.

3.2 Modulo do Parser

Como referido anteriormente, este modulo pode ser dividido em três partes:

3.2.1 Inicializações

Nesta parte, inicia-se a estrutura do pré-processador onde se guarda toda a informação relacionada com a análise do *FILE *.pplh*, como também faz-se os *includes* necessários ao programa.

3.2.2 Parser

Aqui, começa-se por fazer algumas inicializações locais, como por exemplo, de algumas listas ligadas, que nos são muito uteis aquando a utilização das *tags* de fecho, pois, estão a funcionar como *stacks*, o que permite saber sempre qual é a próxima *tag* de fecho a utilizar.

Além disto, cria-se variáveis de contexto que servem de filtros para as expressões regulares, o que permite desta forma tornar o processador mais eficiente.

3.2.3 Função *main*

Aqui encontra-se a função *main* que tem como o objectivo analisar os dados passados na linha de comandos. Este testa se as *flags* passadas estão corretas (que podem ser *'-a'*, *'-l'*, *'-h'*, *'-o'*, das quais é obrigatório escolher uma das 3 primeiras). A *'-a'* serve se quisermos imprimir em HTML e \LaTeX , a *'-l'* é só para o caso de imprimir em \LaTeX , a *'-h'* é para o caso de imprimir apenas em HTML, e por ultimo temo a *'-o'* que serve para escolhermos o nome do *output*. Em caso dos parâmetros não serem respeitados, o programa dá um *printf* com o respectivo erro.

3.3 Modulo do Processador

Neste módulo estão as funções responsáveis por fazer a análise (se necessário) dos argumentos passados pelo módulo do Parser e em seguida insere na estrutura de dados as respectivas *emphstrings* já com a formatação final. Existem várias funções com este objectivo, que são:

- void addTitulo(PPLH*,char*);
- void addAutor(PPLH*,char*);
- void addSeccao(PPLH*,char*,int,int);
- void addHRef(PPLH*,char*);
- void addItem(PPLH*,char*);

- void addOrdList(PPLH*);
- void addItemList(PPLH*);
- void addEndTAG(PPLH*,char*,char*);
- void addTextoNF(PPLH*);
- void addComentario(PPLH*);
- void addQuebra(PPLH*);
- void addFormat(PPLH*,char*);
- void addHTML(PPLH*,char*);
- void addLATEX(PPLH*,char*);
- void addImagem(PPLH*,Image*);
- void addModImg(PPLH*);
- void addTexto(PPLH*,char*);
- void addTabela(PPLH*,Table,int);
- void addBackSlash(PPLH*);
- void addAnd(PPLH*);
- void addLinha(Table,Row);
- void addCelula(Row,char*,char);
- void addIndice(PPLH*,int,char*);
- void addIndList(PPLH*);
- void addIndFecho(PPLH*);

3.4 Modulo PPLH

Este módulo é bastante simples, é apenas constituído pelas funções *'geraHTML'* e *'geraLATEX'*. Estas tem por base fazer a inicialização do *FILE* de *output* seguida das inicializações base de cada linguagem. Carregamento das definições básicas do ficheiro (como tamanho de folha, letra, tamanho de letra...). Após as inicializações básicas de cada formato de texto, a função começa a aceder à estrutura de dados para inserir o título e respectivo autor de cada formato. Feito isto, entramos num ciclo que percorre a respectiva *linkedlist* (a de HTML ou de L^AT_EX), escrevendo no ficheiro criado anteriormente. Quando o ciclo termina, a função começa a fechar a edição dos respectivos documentos e em seguida o *FILE* criado.

3.5 Modulo de Lista Ligada

Aqui temos um módulo de lista ligada genérico, que visa um '*olhar verde*'. Este módulo é utilizado na estrutura do processador, de modo a guardar o *texto* já processado.

4 Conclusão

Com a execução deste trabalho foi-se capaz de exercitar as capacidades no uso do *flex*, pois para a criação do modulo do Parser utilizou-se grande parte dos conhecimentos adquiridos nas aulas, Neste caso, o uso de expressões regulares teve uma grande importância. Mas não foram só vantagens a este nível que obtivemos, pois agora, com este pré-processador podemos utiliza-lo na execução de trabalhos futuros.

Espera-se continuar a desenvolver este pré-processador de forma a abranger o maior numero de funcionalidades comuns entre o *HTML* e o *L^AT_EX*.