

EA872 Laboratório de Programação de Software Básico

Atividade 2

8. Atividades Práticas

Para entender melhor as ferramentas lex (ou flex) e yacc (ou bison), estude os programas abaixo, fazendo referência sempre que necessário às páginas de manual online do Unix.

Programando com LEX

Os códigos a seguir estão disponíveis no arquivo zip fornecido junto com o roteiro e devem ser compilados com o comando `flex p_*.l`, seguido do comando `gcc lex.yy.c -lfl -o p_*` (* deve ser a, b, c, ...).

Atividades para o relatório

As atividades (c) a (g) deverão fazer parte do relatório a ser entregue até a próxima aula. Procure entender os analisadores léxicos abaixo, execute-os e **explique seu funcionamento no relatório, exemplificando com as várias saídas reais obtidas do computador. Observe que não é para dizer apenas o que os programas fazem, mas como eles fazem.**

(c) (1 pt.) p_c.l (exemplo de execução: `./p_c < input_p_c`)

Explique em detalhes o funcionamento deste analisador léxico e descreva seu propósito.

```
%%
[.]          ;
var          ;
[{}]         printf("/");
[]           printf("*");
mod          printf("%");
or           printf("|");
and          printf("&");
begin        ;
end          printf("}");
if           printf("if (");
then         printf(")");
program.*[ ( ) . * $ printf("main() \n{");
[^:;><] [=]   printf("==");
[:] [=]      printf("=");
[<] [>]      printf("!=");
^.*integer;   ShuffleInt();
.             ECHO;
%%
ShuffleInt()
{ int i;
  printf("int "); for(i=0; yytext[i]!=':'; i++) printf("%c", yytext[i]);
  printf(";");
}
```

(d) (2 pts.) p_d.l (execute o analisador com `./p_d` e digite diversas palavras e quebras de linha, terminando com `ctrl-d`.)

Documente seus testes e explique em detalhes o funcionamento deste analisador léxico, mostrando que compreendeu como ele consegue atingir seus objetivos.

```
%{
int lines=0, characs=0, words=0, pages=1;
%}

%START          Palavra
NovaLinha      [\n]
```

```

Espaco          [\t ]
NovaPagina      [\f]

%%
{Espaco}        { BEGIN 0; characs++; }
{NovaLinha}     { BEGIN 0; characs++; lines++; }
{NovaPagina}    { BEGIN 0; characs++; pages++; }
<Palavra>.      { characs++; }
.               { BEGIN Palavra; characs++; words++; }
%%

main()
{
    while (yylex());
    printf("\nResultados: \n%d pagina(s)\n%d linha(s)\n%d palavra(s)\n%d
        caracter(es)\n", pages,lines,words,characs);
}

```

(e) (1 pt.) p_e.l (exemplo de execução: `./p_e < input_p_e`)

Documente seus testes e explique em detalhes o funcionamento deste analisador léxico, mostrando que compreendeu como ele consegue atingir seus objetivos.

```

%{
    int valor;
}%

OCTAL          [0-7]+0

%%

{OCTAL}        { sscanf(yytext,"%o",&valor); printf("%d",valor); }
.              ECHO;

%%

```

(f) (2 pts.) p_f.l

Lembrando que comentários em shell script são caracterizados por '#', escreva um programa `p_f` usando flex para ler um shell script e retirar seus comentários. Explique detalhadamente como seu programa funciona e teste-o utilizando o arquivo `input_p_f`. Sua saída deve ser idêntica ao arquivo `output_p_f` (você pode comparar sua saída com a saída `output_p_f` com o comando `diff`).

```

./p_f < input_p_f > result_p_f
diff result_p_f output_p_f

```

(g) (2 pts.) p_g.l

Escreva um programa em flex para ler uma data no formato `dd/mm/aa`, usando as declarações já feitas no arquivo `p_g.l`, e escrever a mesma por extenso. Por exemplo, a data `19/08/15` deve ser escrita por extenso como: *dezesete de agosto de dois mil e dezesseis*. Sugere-se criar rotinas em C para as conversões de inteiro para string. Documente tudo como indicado no início.

Exemplo de execução: `./p_g < input_p_g`

```

%{
char *unidades[]={ "primeiro", "um", "dois", "tres", "quatro", "cinco",
                   "seis", "sete", "oito", "nove" };
char *interm[]={ "dez", "onze", "doze", "treze",
                 "quatorze", "quinze", "dezesseis",
                 "dezesete", "dezoito", "dezenove" };
char *dezenas[]={ "vinte", "trinta", "quarenta", "cinquenta",

```

```
        "sessenta", "setenta", "oitenta", "noventa");  
char *meses[]={ "janeiro", "fevereiro", "marco", "abril", "maio",  
                "junho", "julho", "agosto", "setembro",  
                "outubro", "novembro", "dezembro"};  
  
%}  
  
/* Dica de uma definição que pode ser usada na seção de regras.*/  
  
PrimeiroCampo  [ ]..[/]  
  
%%  
  
%%
```