

L2 Trabajo con flex

Se realizarán tres ejercicios donde se aplicará la herramienta flex

Se hará un reporte

Usando flex

Tres Ejemplos

- 1.- Simple flex
- 2.- wc
- 3.- tomando fichas (tokens)

Tomado de

<https://web.stanford.edu/class/archive/cs/cs143/cs143.1128/handouts/050%20Flex%20In%20A%20Nutshell.pdf>
y otra fuente

I.- Ejercicio Simple

Presentamos un ejercicio simple

```
/*
flex simple.flex
gcc -o simple lex.yy.c -ll
simple < ejemplo.txt

*/

%{
#include <stdio.h>
%}

%%
[0123456789]+      printf("NUMBER\n");
[a-zA-Z][a-zA-Z0-9]*  printf("WORD\n");
%%
```

Usted lo modificará para que se reconozcan los siguientes lexemas if, then else, for, begin, end
Usted propocricionará el archivo de prueba

II wc : emulando el mandato de linux correspondiente

```
/*
% flex ejewc.flex
% gcc -o count lex.yy.c -ll
% ./count < sample_4.flex
% 346 52 22
*/

%{
int numChars = 0, numWords = 0, numLines = 0;
%}

%%

\n {numLines++; numChars++;}
[^\t\n]+ {numWords++; numChars += yyleng;}
. {numChars++;}

%%

int main() {
yylex();
printf("%d\t%d\t%d\n", numChars, numWords, numLines);
}
```

Se modificará para por palabras usemos el mismo “criterio” del ejercicio 1

III Tercer caso lex de C.

```
/*
El tercer caso examina los tokens de un lenguaje como pascal
% flex sample_5.flex
% gcc -o pasc lex.yy.c -ll
% ./pasc < sample_5.txt
-----
*/

/* scanner for a toy Pascal-like language */

%{
/* need this for the call to atof() below */
#include <math.h>
%}

DIGIT [0-9]
ID [a-z][a-z0-9]*

%%
```

```

{DIGIT}+      { printf( "An integer: %s (%d)\n", yytext, atoi( yytext ) ); }
{DIGIT}+"."{DIGIT}*      { printf( "A float: %s (%g)\n", yytext, atof( yytext ) ); }
if|then|begin|end|procedure|function      { printf( "A keyword: %s\n", yytext ); }
{ID}          { printf( "An identifier: %s\n", yytext ); }

"+"|"-"|"*"|"/"      { printf( "An operator: %s\n", yytext ); }

"{"[^}\n]hola*"}"    /* eat up one-line comments */

[ \t\n]+          /* eat up whitespace */

.      { printf( "Unrecognized character: %s\n", yytext ); }

%%

int main(int argc, char ** argv )
{
    ++argv, --argc; /* skip over program name */
    if ( argc > 0 ) yyin = fopen( argv[0], "r" );
    else yyin = stdin;
    return yylex();
}

/*

```

Y lo que sigue es sample_5.txt

```

-----
hola := como ;
if a = 5 then hola := 157
procedure esta(c : int)
begin
    hola := "Como va";
end
-----

*/

```

Usted creará otro archivo de prueba