

Escuela Superior de Cómputo

Reporte práctica 3

Compiladores, grupo 3CM6

Cruz Cruz Jorge
Feregrino Bolaños Antonio
Gonzales Rodríguez Florencio Javier

Expresiones regulares empleadas:

DIGITO

[0-9]

Expresión encargada de reconocer los números del 0 al 9.

KEYWORDS

"auto"|"break"|"case"|"const"|"continue"|"default"|"do"|"else"|"enum"|"extern"|"for"|"goto"
|"if"|"register"|"return"|"signed"|"sizeof"|"static"|"typedef"|"union"|"unsigned"|"volatile"|"w
hile"

Expresión de cadenas encargada de reconocer las palabras reservadas de C.

TIPOSDATO

"char"|"int"|"long"|"short"|"float"|"double"|"void"

Expresión encargada de reconocer los tipos de dato existentes en C.

DIRECTIVAS

"#[a-zA-Z]+

Esta expresión encuentra las directivas de preprocesador.

LIBS

"<[a-z]+(("[a-zA-Z]+|")>"

Expresión encargada de encontrar definiciones de librerías en C.

OPERADORES_ARITMETICOS

("+"|"-"|"*"|"/"|"%"| "=" | "("+"|"-"|"*"|"/"|"%"| "=" | "+" | "--"

Reconoce los operadores aritméticos

OPERADORES_COMPARACION

"<"| ">"| ">="| "<="| "!="| "=="| "!"| "&&"| "|" |"

Reconoce los operadores de comparación así como los operadores lógicos

OPERADORES_BIT

Reconoce los operadores a nivel de bit.

OPERADORES_OTROS

":"| "::" | "?" | "\\ "

Otros tipos de operadores de C.

OPERADORES

{OPERADORES_ARITMETICOS}|{OPERADORES_COMPARACION}|{OPERADORES_BIT}|{OPERADORE
S_OTROS}

Expresión que agrupa los todos los operadores.

SEPARADORES

```
"{"|"}"|"["|"]"|"("|")"|" ";"|","
```

Expresión para encontrar los separadores que se pueden utilizar en C.

ID

```
([a-zA-Z]|" _")( [a-zA-Z0-9]|" _")*
```

Expresión regular para ubicar los identificadores.

ID_MAL

```
{DIGITO}+([a-zA-Z]|" _")( [a-zA-Z0-9]|" _")*
```

Expresión regular para ubicar los identificadores malformados.

NUMERICOS

```
{DIGITO}+|"."{DIGITO}+|{DIGITO}+
```

Expresión regular que se encarga de reconocer todo tipo de dato numérico

SECUENCIAS_ESCAPE

```
"\\\\"|\"|\"\\n"
```

Expresión que reconoce algunas secuencias de escape.

CARACTERES_ESPECIALES

```
".|" |"|"|"#"|"?"|":|"\\|"@"
```

Nos ayuda a ubicar los caracteres especiales.

STRINGS

```
"\\\"{SECUENCIAS_ESCAPE}|[a-zA-Z0-9]|{CARACTERES_ESPECIALES}|{SEPARADORES}|{OPERADORES}|{ID}|{KEYWORDS}|{TIPOSDATO}|{LIBS}|{DIRECTIVAS}|{NUMERICOS})*\\\""
```

Agrupación de otras reglas para reconocer cadenas de caracteres.

CADENAS

```
{STRINGS}
```

Agrupación “envolvente”.

COMENTARIOS_UNALINEA

```
"/"/*"+([a-zA-Z0-9]|{CARACTERES_ESPECIALES}|{SEPARADORES}|{OPERADORES}|{ID}|{KEYWORDS}|{TIPOSDATO}|{CADENAS}|{LIBS}|{DIRECTIVAS}|{NUMERICOS}|{SECUENCIAS_ESCAPE})*/*+"/"
```

Encargada de ubicar los comentarios de una sola línea.

COMENTARIOS_CONTINUACION

```
("/" | "" ) "*" ([a-zA-Z0-9] | {CARACTERES_ESPECIALES} | {SEPARADORES} | {OPERADORES} | {ID} | {KEYWORDS} | {TIPOSDATO} | {CADENAS} | {LIBS} | {DIRECTIVAS} | {NUMERICOS} | {SECUENCIAS_ESCAPE})*
```

Reconoce la continuación de los comentarios multilínea.

COMENTARIOS

```
{COMENTARIOS_UNALINEA} | {COMENTARIOS_CONTINUACION}
```

Agrupar todos los tipos de comentarios.

```
[\t]
```

Reconoce las tabulaciones.

```
[\n]
```

Reconoce los saltos de línea.

```
.
```

Reconoce todo lo que no fue reconocido anteriormente.

Funciones adicionales

Además de usar la función main, añadimos la función:

```
imp_pal
```

Prototipo:

```
void imp_pal(FILE *,char *);
```

Explicación:

Nos auxilia en la generación de los archivos de salida, recibe el archivo sobre el que se escribirá y el tipo de elemento a escribir.

Dependiendo del tipo y del archivo escribirá en la tabla de símbolos o en la tabla de errores.