

Análisis de proceso de scanner para language C

Proyecto I: Compiladores e Intérpretes

Carlos Adrián Gómez Segura

Kevin Jafet Zamora

Gabriel Solórzano Chanto

Tecnológico de Costa Rica

Francisco Torres Rojas

Abril 10 , 2019

Semestre 1

Table of Contents

Proceso de scanning
Herramienta Flex
Proceso de parsing
Herramienta Bison
Análisis del programa
Histogramas del análisis
Referencias

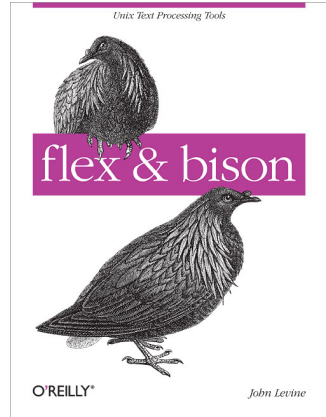


Table of Contents

Proceso de scanning

Herramienta Flex

Proceso de parsing

Herramienta Bison

Análisis del programa

Histogramas del análisis

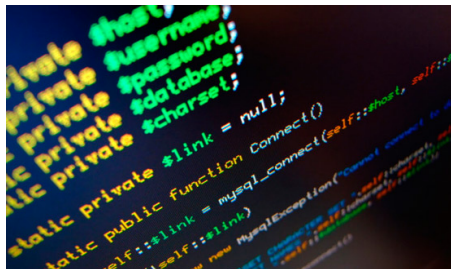
Referencias

Descripción del proceso de de scanning.

El preproceso es el primer proceso de ejecución que se realiza en el momento de compilación.

Se encarga principalmente de:

- #defines
- #includes



Cont. Descripción del proceso de de scanning.

La primera parte es el preproceso que se encarga de preparar el código para que luego el scanner lo utilice. Se encarga de hacer lo includes y hacer el define

Include, busca y carga código que se encuentra en la misma carpeta, en la que se está ejecutando el archivo, en el preproceso también se dirige al archivo, por ende, busca el código y lo agrega al archivo deseado.



Cont. Descripción del proceso de de scanning.

Define: busca y reemplaza en todo el código, lo establecido en la línea del Define, lo reemplaza por constantes.

Otra función:

- Eliminar los comentarios



Cont. Descripción del proceso de de scanning.

Sección de scanner

El scanner se va a encargar de generar unidades léxicas mínimas en este caso, tokens, a partir de Flex, recibe el programa C, y Flex reconocerá qué tipo de token es, el mismo va a llevar ciertos contadores sobre el programa para luego ser procesadas en una presentación Beamer.



Table of Contents

Proceso de scanning

Herramienta Flex

Proceso de parsing

Herramienta Bison

Análisis del programa

Histogramas del análisis

Referencias

Table of Contents

Proceso de scanning

Herramienta Flex

Proceso de parsing

Herramienta Bison

Análisis del programa

Histogramas del análisis

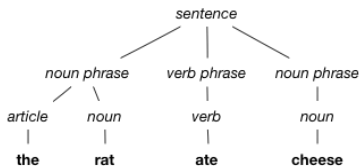
Referencias

Descripción del proceso del proceso de parsing.

El proceso de parsing consiste en que se analiza una cadena de símbolos de acuerdo a las reglas de una gramática formal que sido previamente establecida. En el proceso de parsing, se genera la estructura de un árbol sintáctico, con el objetivo de verificar que el contenido del programa cumpla con las reglas sintácticas.



Cont. Descripción del proceso del proceso de parsing.



El análisis sintáctico convierte el texto de entrada en árboles, que pueden ser estructuras de datos o estar implícitas en el proceso de parsing, que se vuelven de gran importancia y brindan una jerarquía implícita de la entrada.

Table of Contents

Proceso de scanning

Herramienta Flex

Proceso de parsing

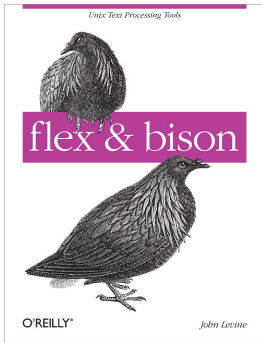
Herramienta Bison

Análisis del programa

Histogramas del análisis

Referencias

Descripción Herramienta Bison



GNU bison es un programa generador de analizadores sintácticos de propósito general. Bison convierte la descripción formal de un lenguaje, escrita como una gramática libre de contexto LALR, en un programa que realiza análisis sintáctico.

Table of Contents

Proceso de scanning

Herramienta Flex

Proceso de parsing

Herramienta Bison

Análisis del programa

Histogramas del análisis

Referencias

Notación.

Palabras Reservadas

Carácteres Especiales

Operadores

Strings

Identificadores

Constantes

Errores Léxicos

Análisis de código

```
int *a;  
void cleanCode(){  
    getcwd(direccion, sizeof(direccion)) ;  
    strcat(direccion, "/OUTPUT.txt");
```

Análisis de código

```
fileInput = fopen(direccion, "r");  
fileOutput = fopen("OUTPUT2.txt", "w");  
char in_char;  
in_char = fgetc(fileInput);  
int band = 0;
```

Análisis de código

```
while( !=  
Syntax Error, in line:3  
EOF && (in_char == '\n')){  
in_char = fgetc(fileInput);  
}  
while(in_char != EOF){  
if(in_char == '\n'){
```

Análisis de código

```
if(band == 0){  
fputc(in_char, fileOutput)  
band  
Syntax Error, in line:3  
= 1;  
in_char = fgetc(fileInput);  
}
```

Análisis de código

```
else{  
in_char = fgetc(fileInput);  
}  
}  
else{
```

Análisis de código

```
band = 0;  
fputc(in_char, fileOutput);  
in_char = fgetc(fileInput);  
}  
}
```

Análisis de código

```
return;
```

```
}
```

```
int *a;
```

```
void cleanCode(){
```


Análisis de código

```
getcwd(direccion, sizeof(direccion)) ;  
strcat( direccion, "/OUTPUT.txt" );  
fileInput = fopen(direccion, "r" );  
fileOutput = fopen( "OUTPUT2.txt", "w" );  
char in_char;
```

Análisis de código

```
in_char = fgetc(fileInput);
```

```
int band = 0;
```

```
while( !=
```

```
Syntax Error, in line:3
```

```
EOF && (in_char == '\n')){
```

```
in_char = fgetc(fileInput);
```

```
}
```

Análisis de código

```
while(in_char != EOF){  
if(in_char == '\n'){  
if(band == 0){  
fputc(in_char, fileOutput)  
band  
Syntax Error, in line:3  
= 1;  
}
```

Análisis de código

```
in_char = fgetc(fileInput);  
}  
else{  
in_char = fgetc(fileInput);  
}
```

Análisis de código

```
}  
else{  
band = 0;  
fputc(in_char, fileOutput);  
in_char = fgetc(fileInput);
```

Análisis de código

```
}  
}  
return;  
}
```

Análisis de código

Table of Contents

Proceso de scanning

Herramienta Flex

Proceso de parsing

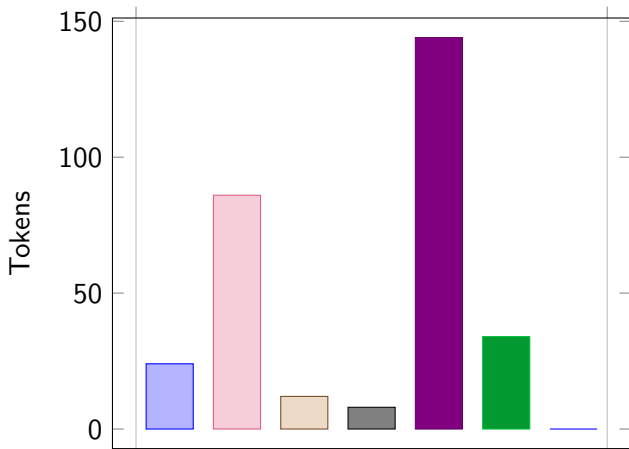
Herramienta Bison

Análisis del programa

Histogramas del análisis

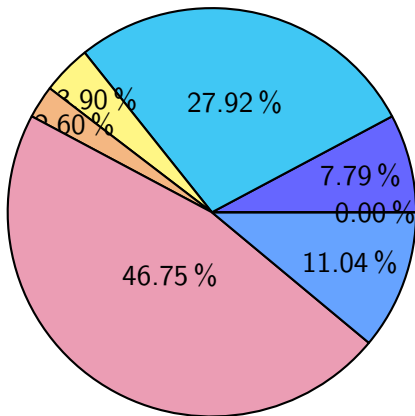
Referencias

Histograma de la cantidad de tokens



Palabras reservadas Identificadores Strings
Constantes Símbolos Especiales Operadores
Errores Léxicos

Gráfico circular del porcentaje de uso de tokens



- Palabras reservadas: 7.79 %
- Identificadores: 27.92 %
- Strings: 3.90 %
- Constantes: 2.60 %
- Símbolos especiales: 46.75 %
- Operadores: 11.04 %
- Errores Léxicos: 0.00 %

Table of Contents

Proceso de scanning

Herramienta Flex

Proceso de parsing

Herramienta Bison

Análisis del programa

Histogramas del análisis

Referencias

Referencias

Levine, J. (2009). *Flex & Bison: Text Processing Tools*. O'Reilly Media, Inc.