Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа № 5

По дисциплине ЯПИС

За шестой семестр

Тема: «Компилятор компиляторов»

Вариант 7

Выполнил:

Студент 3 курса

Группы ИИ-16 (2)

Пешко А.С.

Проверил:

Слинко Е.В.

Брест, 2021

Цель: изучить и опробовать на практике генератор синтаксических анализаторов YACC.

**Ход работы**

**Задание**

**7.** Входной язык содержит последовательность вызовов процедур, разделенных символом ;(точка с запятой). Вызов процедуры должен состоять из имени процедуры и списка параметров. В качестве параметров могут выступать идентификаторы, строковые константы, заключенные в двойные кавычки и одиночные символы, заключенные в одинарные кавычки.

**Задание**. Выполнить задание по варианту, реализовав синтаксический анализатор, который для каждой обнаруженной конструкции выдает сообщение на экран.

Код программы:

**lab5l.l**

%{

#include "lab5b.tab.h"

#include <stdio.h>

void err();

%}

%%

"(" {return(LEFT);}

")" {return(RIGHT);}

"," {return(COMMA);}

";" {return(SEMICOLON);}

[\'].[\'] {

yylval.sval = malloc(strlen(yytext));

strncpy(yylval.sval, yytext, strlen(yytext));

yylval.sval[strlen(yytext)] = '\0';

return(SYMBOL);

}

[\_a-zA-Z][a-zA-Z0-9\_]\* {

yylval.sval = malloc(strlen(yytext));

strncpy(yylval.sval, yytext, strlen(yytext));

yylval.sval[strlen(yytext)] = '\0';

return(FUNC);

}

[\"][a-zA-Z0-9\_]\*[\"] {

yylval.sval = malloc(strlen(yytext));

strncpy(yylval.sval, yytext, strlen(yytext));

yylval.sval[strlen(yytext)] = '\0';

return(STR);

}

[ \n\t]+ {}

. {err(); exit(0);}

%%

void err(){

printf("Wrong input");

}

yywrap () {

return (0);

}

**lab5b.y**

%{

#include <stdio.h>

void yyerror (char const \*);

%}

%union

{

char \*sval;

}

%token LEFT

%token RIGHT

%token <sval> FUNC

%token <sval> SYMBOL

%token <sval> STR

%token SEMICOLON

%token COMMA

%start functions

%%

functions: | functions function

;

function: FUNC LEFT {printf("\nProcedure name: %s\n", $1);}

variables

RIGHT

SEMICOLON

;

variables: /\* empty \*/

| vars identifier

;

vars: /\* empty \*/

| vars variable

;

variable: identifier COMMA

;

identifier: SYMBOL {printf("Single character : %s\n", $1);}

| STR {printf("Constant string : %s\n", $1);}

;

%%

void yyerror (char const \*s)

{

printf("Error: %s\n", s);

}

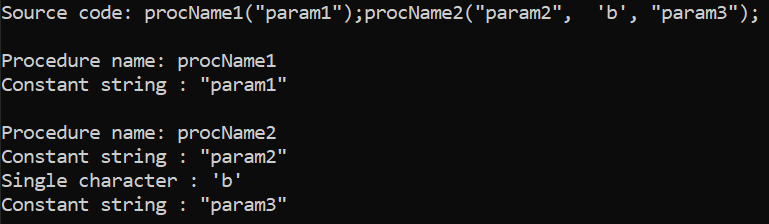
main (void) {

printf("Source code: ");

yyparse();

return 0;

}



**Вывод:**изучили и опробовали на практике генератор синтаксических анализаторов YACC.