Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

Кафедра интеллектуально-информационных технологий

Лабораторная работа №5

По дисциплине «Языковые процессы интеллектуальных систем»

Выполнил:

студент 4 курса

группы ИИ-22

Копанчук Е. Р.

Проверил:

Монтик Н. С.

Брест-2024

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

**Ход работы:**

**Задание:** входной язык содержит упрощенные операторы цикла типа  
while <логическое выражение> do <оператор присваивания>; Логическое выражение может содержать идентификаторы, знаки операций сравнения, целые десятичные числа без знака, скобки и логические операции and и or. Оператор присваивания должен состоять из идентификатора, знака присваивания и целой десятичной константы без знака.

**Код программы:**

**int yylex() {**

**int c;**

**while ((c = getchar()) != EOF) {**

**// Пропускаем пробельные символы**

**if (isspace(c)) continue;**

**// Проверка на ключевые слова и идентификаторы**

**if (isalpha(c)) {**

**char buffer[100];**

**int i = 0;**

**// Считываем символы, пока они алфавитные или цифровые (идентификатор)**

**do {**

**buffer[i++] = c;**

**c = getchar();**

**} while (isalnum(c) || c == '\_'); // Добавляем поддержку символа '\_'**

**buffer[i] = '\0'; // Завершаем строку**

**ungetc(c, stdin); // Возвращаем последний символ в поток**

**// Сравниваем со словами "while" и "do"**

**if (strcmp(buffer, "while") == 0) {**

**printf("Token: WHILE\n");**

**return WHILE;**

**} else if (strcmp(buffer, "do") == 0) {**

**printf("Token: DO\n");**

**return DO;**

**} else {**

**printf("Token: IDENTIFIER (%s)\n", buffer);**

**return IDENTIFIER;**

**}**

**}**

**// Числа**

**if (isdigit(c)) {**

**yylval = c - '0';**

**while (isdigit(c = getchar()))**

**yylval = yylval \* 10 + (c - '0');**

**ungetc(c, stdin);**

**printf("Token: NUMBER (%d)\n", yylval);**

**return NUMBER;**

**}**

**// Операторы и символы**

**switch (c) {**

**case '=':**

**if ((c = getchar()) == '=') {**

**printf("Token: EQ\n");**

**return EQ;**

**}**

**ungetc(c, stdin);**

**printf("Token: ASSIGN\n");**

**return ASSIGN; // Если это просто '=', это оператор присваивания**

**case '!':**

**if ((c = getchar()) == '=') {**

**printf("Token: NEQ\n");**

**return NEQ;**

**}**

**ungetc(c, stdin);**

**break;**

**case '<':**

**if ((c = getchar()) == '=') {**

**printf("Token: LE\n");**

**return LE;**

**}**

**ungetc(c, stdin);**

**printf("Token: LT\n");**

**return LT;**

**case '>':**

**if ((c = getchar()) == '=') {**

**printf("Token: GE\n");**

**return GE;**

**}**

**ungetc(c, stdin);**

**printf("Token: GT\n");**

**return GT;**

**case '&':**

**if ((c = getchar()) == '&') {**

**printf("Token: AND\n");**

**return AND;**

**}**

**ungetc(c, stdin);**

**break;**

**case '|':**

**if ((c = getchar()) == '|') {**

**printf("Token: OR\n");**

**return OR;**

**}**

**ungetc(c, stdin);**

**break;**

**case '+':**

**case '-':**

**case '\*':**

**case '/':**

**printf("Token: '%c'\n", c);**

**return c;**

**case ';':**

**printf("Token: ';'\n");**

**return ';';**

**default:**

**printf("Unknown character: '%c'\n", c);**

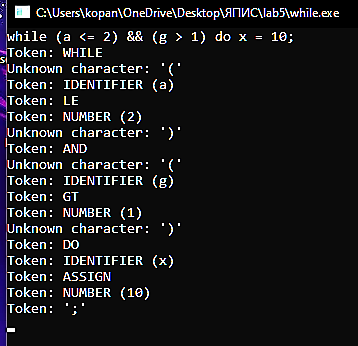
**break;**

**}**

**}**

**return 0; // Конец ввода**

**}**

****