|  |  |
| --- | --- |
| **Изображение выглядит как текст, эмблема, герб, нашивка  Автоматически созданное описание** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехника и комплексная автоматизация (РК)

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**по курсу «Программное обеспечение систем автоматизированного проектирования»**

Студент Долженко Анастасия Тимофеевна

Группа РК6-32Б

Тип задания Лабораторная работа №4

Вариант Y29

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженко А.Т.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бочаров В.А.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2023 г.*

# Задание

Разработать синтаксическиӗ анализатор для грамматического разбора входного потока цифровых данных с целью распознавания записей двоичных наборов, которые соответствуют формальному языку L = {1n0m1m0n | n, m > 0}. Все записи двоичных наборов должны передаваться синтаксическому анализатору строками потока стандартного ввода. Результаты грамматического разбора должны отображать соответствующие диагностические сообщения потока стандартного вывода.

# Выполнение задания

**Требования к функциональным характеристикам**

Программа lab4 должна обеспечивать синтаксический анализ записей двоичных наборов, которые соответствуют формальному языку L = {1n0m1m0n | n, m > 0}, передаваемые символьной строкой из потока стандартного ввода.

Любые нарушения допустимого двоичного числа во входной строке из потока стандартного ввода должны сопровождаться информационным сообщением ("Incorrect") в потоке стандартного вывода.

Результат распознавания программой корректного двоичного числа должен сопровождаться отображением в потоке стандартного вывода информационным сообщением ("Correct").

**Требования к составу и параметрам технических средств**

Программа la4 должна быть разработана исходя из возможности реализации на стандартном составе технических средств компьютеров любой архитектуры, после соответствующей трансляции исходного кода.

**Условия эксплуатации**

Программа lab4 должна:

* быть ориентирована на эксплуатацию в операционной среде OS UNIX любой версии
* быть реализована в виде выполняемого файла с именем lab4, по которому она должна вызываться средствами любого командного процессора OS UNIX
* эксплуатироваться в интерактивном режиме, читая строку входного выражения из потока стандартного ввода и отображая результат ее обработки в потоке стандартного вывода.

**Требования к информационной и программной совместимости**

Чтобы обеспечить выполнение требуемых технических характеристик, программа lab4 должна реализовывать синтаксический анализ любой входной строки двоичного числа из потока стандартного ввода.

Синтаксический анализатор программы lab4 должен обеспечивать грамматический разбор двоичного числа с целью установить соответствие или несоответствие содержащей его строки потока стандартного ввода требуемому формату двоичного числа.

Грамматику двоичного числа синтаксического анализатора программы lab4 должны составлять следующие элементы:

* терминальные символы (терминалы), соответствующие структурным единицам (лексемам) входного двоичного числа;
* начальный нетерминальный символ (начальный нетерминал), к которому приводится входное двоичное число;
* нетерминальные символы (нетерминалы), обозначающие допустимые варианты комбинации лексем во входном двоичном числе;
* система правил вывода, обеспечивающая грамматический разбор входного двоичного числа.

Терминальные символы грамматики двоичного числа синтаксического анализатора программы lab4 представляются лексемами, специфицированными литералами ‘0’ и ‘1’, которые обозначают коды ASCII соответствующих цифр двоичных чисел.

Нетерминальные символы грамматики двоичного числа обозначаются именами: zeros, ones.

Они должны определяться продукциями грамматики скобок синтаксического анализатора программы lab4 в форме, близкой нотации Бэкуса-Наура (БНФ), следующим образом:

* zero: идущие подряд 0 в записи двоичного числа
* ones: идущие подряд 1 в записи двоичного числа

Начальный нетерминал грамматики двоичного числа синтаксического анализатора программы lab4 должен обозначаться именем correct. Он должен выводиться из любой корректной записи двоичного числа, соответствующего формальному языку L = {1n0m1m0n | n, m > 0}. Грамматический вывод начального нетерминала из любой входной строки, содержащей корректную запись двоичного числа, должна обеспечивать система продукций грамматики двоичного числа.

Система продукций (правил вывода) грамматики двоичного числа синтаксического анализатора программы lab4 должна обеспечивать грамматический разбор произвольной входной строки потока стандартного ввода путем приведения терминалов и нетерминалов к начальному нетерминалу методом LR(1) анализа снизу-вверх по дереву вывода. Приведение входной строки к начальному нетерминалу по правилам вывода грамматики двоичного числа должно означать успех грамматического разбора двоичного числа, т. е. распознавание корректного двоичного числа. Альтернативный результат должен рассматриваться как синтаксическая ошибка двоичного числа во введенной строке.

Для разработки синтаксического анализатора программы lab4, необходимо использовать генератор синтаксических анализаторов (далее по тексту - YACC) из состава OS UNIX, инструментальные средства которого ориентированы на обработку файла спецификаций (далее по тексту, Yacc-файл) проектируемого синтаксического анализатора.

При разработке синтаксического анализатора программы lab4 необходимо составить Yacc-файл, отражающий специфику грамматического разбора двоичного числа, и сохранить его под именем lab4.y в выбранном доступном рабочем каталоге файловой системы OS UNIX.

Проектируемый Yacc-файл lab4.y должен состоять из 3-х секций: деклараций, правил и функций. Разделителем секций должны быть символические пары %%, расположенные в начальных позициях содержащих их строк Yacc-файла. Каждая секция Yacc-файла lab4.y должна содержать соответствующее число комментариев в формате, принятом для системы программирования C.

Секция деклараций Yacc-файла lab4.y должна включать:

* объявление начального нетерминала грамматики двоичного числа с помощью директивы %Start
* cпецификацию блока внешних описаний, ограниченную директивами %{ и %}, в котором необходимо подключить библиотеки stdio и stdlib

В секции правил Yacc-файла lab4.y должны быть приведены описания продукций приведения нетерминалов грамматики двоичного числа в соответсвии с требованиями предыдущих пунктов.  
Каждая продукция секции правил Yacc-файла lab4.y должна быть задана в нотации, близкой к форме Бэкуса-Наура, где в левой части указывается приводимый нетерминал, а в правой - последовательность терминалов и/или нетерминалов грамматики двоичного числа, которые перечисляются через пробел. Для разделения частей продукции должен использоваться символ двоеточия (:). Каждую продукцию нужно начинать с новой строки и завершать либо символом точки с запятой (;), либо блоком действий в фигурных скобках.

Секция функций Yacc-файла lab4.y должна содержать исходный код, оформленный по правилам системы программирования C, для 3-х функций с предопределенными именами: yylex(), yyerror() и main(), которые должны иметь целочисленный (типа int) код возврата. Исходный код перечисленных функций используется непосредственно для формирования кода программы lab4.

Функция yylex() секции функций Yacc-файла lab4.y должна выполнять лексический анализ входной строки из потока стандартного ввода со следующими целями: обнаружить единицы. Вызов функции yylex() будет автоматически производиться при грамматическом разборе входной строки для получения двоичного числа, пока не достигнут символ перевода строки ('\n').

Для достижения целей лексического анализа исходный код функции yylex() должен предусматривать:

* побайтное чтение любой заданной строки входного двоичного числа из потока стандартного ввода, например, с помощью библиотечной функции getchar() системы программирования C;
* распознавание единиц и нулей во входной строке
* возврат кодов ASCII обнаруженных других символов с помощью оператора return
* возврат нулевого кода при достижении конца входной строки с помощью оператора return(0)

Функция yyerror() секции функций Yacc-файла lab4.y должна обеспечивать обработку ошибок записи двоичного числа, обнаруженных при грамматическом разборе входной строки двоичного числа. Ее вызов будет происходить автоматически, когда двоичное число не может быть приведено к начальному нетерминальному символу грамматики двоичного числа, обеспечивая аварийное прерывание грамматического разбора текущей входной строки потока стандартного ввода.

Спецификация функции yyerror() в Yacc-файле lab4.y не должна иметь аргументов. Для отображения диагностического сообщения в потоке стандартного вывода, рекомендуется применить библиотечную функцию printf() системы программирования С.

Спецификация функции main() в секции функций Yacc-файла lab4.y должна содержать исходный код основной функции программы lab4, который обеспечивает обращение к функции синтаксического анализа yyparse() и передачу ее кода возврата во внешнюю операционную среду. Исходный код функции yyparse() формируется генератором синтаксических анализаторов YACC по Yacc- файлу и предусматривает вызов функции yylex() для выполнения лексического анализа, также как передачу управления функции yyerror() для обработки ошибок грамматического разбора. Код возврата функции yyparse() должен однозначно отражать результат грамматического разбора, принимая значение 0 при его успехе, или 1 - при обнаружении синтаксических ошибок.

Исходный код секции функций Yacc-файла lab4.y и функции yyparse() образует исходный код программы lab4, который должен формироваться генератором синтаксических анализаторов YACC в файле с предопределенным именем y.tab.c в текущем рабочем каталоге файловой системы OS UNIX. Выполняемый модуль программы lab4 должен быть создан по файлу исходного кода y.tab.c в выполняемом файле lab4 средствами компилирующей системы программирования C.

# Стадии и этапы разработки

1. Поставить Yacc-файл lab4.y
2. Получить исходный код синтаксического анализатора в файле y.tab.c, обработав Yacc-файл lab4.y командой yacc, следующим образом:

*$ yacc lab4.y*

1. Сформировать выполняемый модуль в файле lab4, компилируя исходный код синтаксического анализатора в файле y.tab.c следующей командой:

*$ cc -o lab4 y.tab.c*

# Код

# %{

# #include <stdio.h>

# int yylex();

# void yyerror();

# %}

# %token ZERO ONE

# %start input

# %%

# input : { printf("Start input:\n"); } | input line ;

# line : ones zeros ones zeros '\n' { if ($1 == $4 && $2 == $3) { printf("Correct\n");} else { yyerror(); }} | '\n' { printf("Empty line\n"); } | error'\n' { yyerrok; } ;

# zeros:ZERO{++$$;} | zeros ZERO {++$$;};

# ones:ONE{++$$;} | ones ONE {++$$;};

# %%

# int yylex() {

# char c;

# c = getchar();

# switch(c) {

# case '0': return(ZERO);

# case '1': return(ONE);

# case EOF: return(0);

# default: return(c);

# }

# }

# void yyerror() { printf("Incorrect\n"); }

# int main() { return(yyparse()); }

# Результаты тестирования

# Изображение выглядит как текст, Шрифт, снимок экрана, типография Автоматически созданное описание Изображение выглядит как текст, Шрифт, снимок экрана, типография Автоматически созданное описание

# Список литературы

1. Кристиан К. Введение в операционную систему UNIX, M.: Финансы и статистика, 1985.
2. Тихомиров В.П., Давидов М.И. Операционная система ДЕМОС: инструментальные средства программирования, М.: Финансы и статистика, 1988.
3. SCO XENIX, Development System, LEX Programmer Guide, SCO Inc., 1986. (Имеется русский перевод: Генератор лексических анализаторов LEX. Руководство Программиста)