Санкт-Петербургский политехнический университет

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**К У Р С О В А Я Р А Б О Т А**

**Разработка учебной системы программирования**

Дисциплина: «Системы программирования»

Вариант № 10

Исполнители:

Студенты группы 3540904/10101 Томилин И. С.

Руководитель Расторгуев В. Я.

Санкт-Петербург

2022

# Задание на курсовую работу

Задание на КР дано в следующей общей формулировке:

«Необходимо выполнить доработку элементов макета учебной системы программирования до уровня, позволяющего обрабатывать «новые» для макета конструкции языка высокого уровня, примененные в соответствующем варианте».

Общая формулировка персонализирована в варианте № 10, приведенном на рис. 1.

**Условный персональный вариант № 10**

EX10: PROC OPTIONS (MAIN);

DCL A DEC FIXED INIT ( 5 );

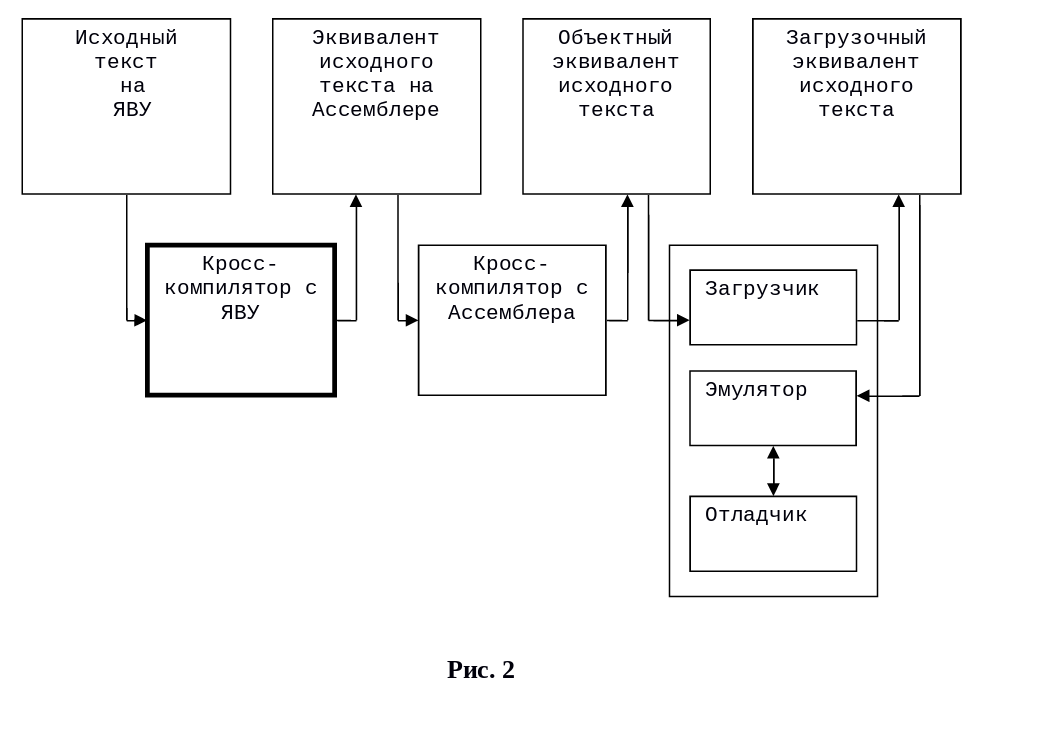
DCL B BIT ( 16 );

B = A;

END EX10;

**Рис. 1**

Схема функционирования макета учебной системы программирования, являющегося, согласно заданию на КР, одним из элементов исходных данных, приведена на рис. 2.



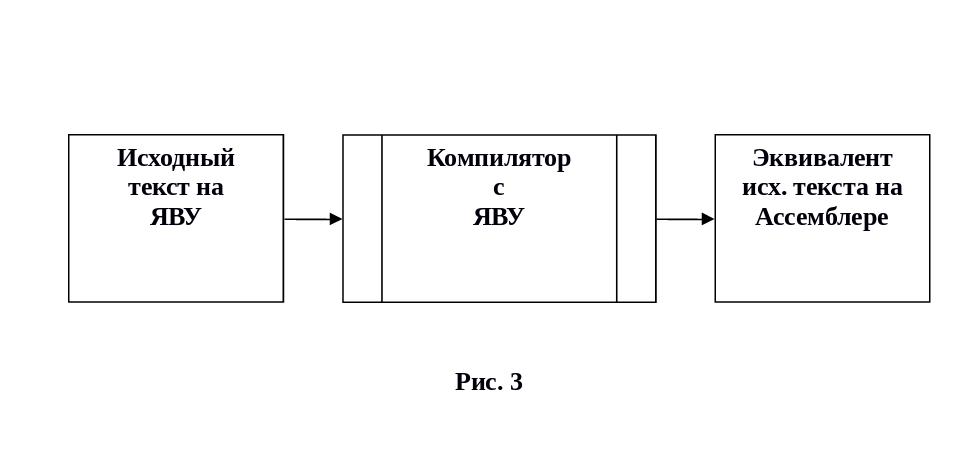
Дополнительные требования к КР, вытекающие из рис.2, предполагают то, что данная система программирования работает на технологической ЭВМ (IBM PC) и является по существу кросс-системой для объектной ЭВМ (IBM 370). В этой системе:

* в качестве языка высокого уровня (ЯВУ) выбран язык, образованный из подмножества языковых конструкций ПЛ1, а исходная программа готовится в виде текстового файла технологической ЭВМ с расширением \*.pli,
* язык АССЕМБЛЕРА сформирован из языковых конструкций АССЕМБЛЕРА IBM 370, а ассемблеровский эквивалент исходной программы формируется в виде текстового файла технологической ЭВМ с расширением \*.ass,
* объектный эквивалент исходной программы готовится в формате объектных файлов операционной системы ОС IBM 370 и хранится в виде двоичного файла технологической ЭВМ с расширением \*.tex;
* загрузочный эквивалент исходной программы представляет собой машинный код IBM 370, запоминаемый в области ОЗУ технологической ЭВМ, являющейся зоной загрузки для эмулятора объектной ЭВМ.

На рис. 2 жирным контуром выделен компилятор с ЯВУ, являющийся тем элементом макета, результаты модификации которого под персональный вариант № 10 (см. рис 1) приведены в данном отчете.

# 2. Постановка задачи модификации макета в части компилятора с ЯВУ

Схема функционирования модифицированного макетного компилятора с ЯВУ должна соответствовать рис. 3. Функционал модифицированного макетного компилятора должен обеспечить компиляцию исходного текста на ЯВУ (см. рис. 1) в эквивалент исходного текста на Ассемблере (см. рис. 4).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Метка | Коп | Операнды | Пояснения |
| 1 | EX10 | START | 0 | Счетчик относительно адреса в 0 |
| 2 |  | BALR | @RBASE, 0 | Загрузка регистра @RBASE |
| 3 |  | USING | \*, @RBASE | Назначить регистр базой |
| 4 |  | MVC | @BUF+2(6), A | Перемещаем из области памяти A в @BUF |
| 5 |  | CVB | @RRAB, @BUF | Преобразование |
| 6 | @LBL | LR | @RRAB1, @RRAB | Метка для цикличного выполнения |
| 7 |  | N | @RRAB1, @PTR | AND один операнд в регистре второй в памяти |
| 8 |  | C | @RRAB1, @NULL | Сравнение рабочего регистра с нулевой const |
| 9 |  | BC | 6, @CONT | Проверка маски для меньше или больше |
| 10 |  | SLL | @RRAB, 1 | Смещаем влево на 1 бит |
| 11 |  | BC | 15, @LBL | Замыкание метки цикла |
| 12 | @CONT | SRL | @RRAB, 16 | Метка в которую выпрыгиваем, когда в BC единица и смещаем на 16 вправо |
| 13 |  | STH | @RRAB, B | Сброс полуслова |
| 14 |  | BCR | 15, @RVIX | Выход из программы |
| 15 | A | DC | PL6’5’ | Инициализация переменной |
| 16 | B | DS | BL16 | Выделение памяти |
| 17 |  | DS | 0F | Выравнивание памяти для CVB |
| 18 | @BUF | DC | PL8’0’ | Выделение памяти для размещения 15 разрядов (Превращаем 11 разрядов в 15) |
| 19 | @PTR | DC | BL32’1’ | Указатель в памяти |
| 20 | @NULL | DC | F’0’ | Заводим const 0 для сравнения |
| 21 | @RRAB | EQU | 3 | RRAB назначим 3 |
| 22 | @RRAB1 | EQU | 4 | RRAB1 назначим 4 |
| 23 | @RBASE | EQU | 15 | @RBASE назначим 15 |
| 24 | @RVIX | EQU | 14 | @RVIX назначим 14 |
| 25 |  | END |  | Конец текста блока |

**Рис. 4**

***Согласованные функциональные ограничения***

По согласованию с преподавателем в модифицированном макетном компиляторе введены следующие функциональные ограничения:

1) Следует отметить, что код будет работать корректно только если значения A > 0, если A=0 то цикл будет бесконечно выполняться. Поэтому ограничиваемся значениями A > 0, в нашем случае A = 5

**3. Модификация состава и структуры БД макета**

**компилятора с ЯВУ**

***3.1 Модификация БД лексического анализатора***

БД лексического анализатора определяется путем выделения подмножества

терминальных символов, в которое входят все терминальные символы, являющиеся

разделителями термов языка.

Для персонального варианта (см. рис. 1) разделителями термов являются терминальные

символы из множества:

{ ":" , "(" , ")" , ";" , "=", “\_”},

где модификация лексических настроек демо-примера не требуется.

***3.2 Модификация БД синтаксического анализатора***

Высокоуровневое определение модифицированного синтаксиса для персонального варианта с выделением модификаций **жирным** **шрифтом**.

Были внесены следующие изменения:

1. Нетерминал <ODC> расширен терминалом для объявления типа DEC FIXED
2. Нетерминал <ODC> расширен терминалом для объявления типа BIT
3. Добавлен нетерминал **<DCF>** для определения десятичного числа, требуется для объявления DEC FIXED

Измененный синтаксис согласно формы БНФ представлен ниже.

1. <PRO> ::= <OPR><TEL><OEN>

2. <OPR> ::= <IPR>:PROC\_OPTIONS(MAIN);

3. <IPR> ::= <IDE>

4. <IDE> ::= <BUK> | <IDE><BUK> | <IDE><CIF>

5. <BUK> ::= A | B | C | D | E | M | P | X

6. <CIF> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

7. <TEL> ::= <ODC> | <TEL><ODC> | <TEL><OPA>

8. <ODC> ::= DCL\_<IDE>\_BIN\_FIXED(<RZR>); |

DCL\_<IDE>\_BIN\_FIXED(<RZR>)INIT(<LIT>); |

**DCL\_<IDE>\_CHAR(<RZR>); |**

**DCL\_<IDE>\_DEC\_FIXED\_INIT(<DCF>); |**

**DCL\_<IDE>\_BIT\_(<RZR>);**

9. <IPE> ::= <IDE>

10. <RZR> ::= <CIF> | <RZR><CIF>

11. <LIT> ::= <MAN>B

12. <MAN> ::= 1 | <MAN>0 | <MAN>1

13. <OPA> ::= <IPE>=<AVI>;

14. <AVI> ::= <LIT> | <IPE> | <AVI><ZNK><LIT> | <AVI><ZNK><IPE>

15. <ZNK> ::= + | -

16. <OEN> ::= END\_<IPR>;

**17. <DCF> ::= <CIF> | <DCF><CIF>**

Здесь использованы следующие метасимволы и символы:

"<" и ">" - левый и правый ограничители нетерминального символа,

"::=" - метасимвол со смыслом "равно по определению",

"|" - метасимвол альтернативного определения "или",

"\_" - терминальный символ со смыслом "пробел",

"<PRO>" - нетерминал "программа",

"<OPR>" - нетерминал "оператор пролога программы",

"<IPR>" - нетерминал "имя программы",

"<IDE>" - нетерминал "идентификатор",

"<BUK>" - нетерминал "буква",

"<CIF>" - нетерминал "цифра",

"<TEL>" - нетерминал "тело программы",

"<ODC>" - нетерминал "оператор declare",

"<IPE>" - нетерминал "имя переменной",

"<RZR>" - нетерминал "разрядность",

"<LIT>" - нетерминал "литерал",

"<MAN>" - нетерминал "мантисса",

"<OPA>" - нетерминал "оператор присваивания арифметический",

"<AVI>" - нетерминал "арифметическое выражение",

"<ZNK>" - нетерминал "знак",

"<OEN>" - нетерминал "оператор эпилога программы"**,**

**"<DCF>" - нетерминал "некое десятичное значение".**

Сначала перепишем каждое правило грамматики, заменяя формат метаязыка БНФ на формат продукций (или на форму распознавания). Для этого в каждой альтернативе каждого правила в формате метаязыка БНФ поменяем местами правую и левую части, изменив при этом символ "::=" на "****". Получим следующий набор продукций:

1. <OPR><TEL><OEN> **** <PRO>
2. <IPR>:PROC\_OPTIONS(MAIN); **** <ORP>
3. <IDE> **** <IPR>
4. <BUK> **** <IDE>
5. <IDE><BUK> **** <IDE>
6. <IDE><CIF> **** <IDE>
7. A **** <BUK>
8. B **** <BUK>
9. C **** <BUK>
10. D **** <BUK>
11. E **** <BUK>
12. M **** <BUK>
13. P **** <BUK>
14. X **** <BUK>
15. 0 **** <CIF>
16. 1 **** <CIF>
17. 2 **** <CIF>
18. 3 **** <CIF>
19. 4 **** <CIF>
20. 5 **** <CIF>
21. 6 **** <CIF>
22. 7 **** <CIF>
23. 8 **** <CIF>
24. 9 **** <CIF>
25. <ODC> **** <TEL>
26. <TEL><ODC> **** <TEL>
27. <TEL><OPA> **** <TEL>
28. DCL\_<IDE>\_BIN\_FIXED(<RZR>); **** <ODC>
29. DCL\_<IDE>\_BIN\_FIXED(<RZR>)INIT(<LIT>); **** <ODC>
30. **DCL\_<IDE>\_CHAR(<RZR>);  <ODC>**
31. **DCL\_<IDE>\_DEC\_FIXED\_INIT(<DCF>);  <ODC>**
32. **DCL\_<IDE>\_BIT\_(<RZR>);  <ODC>**
33. <IDE> **** <IPE>
34. <CIF> **** <RZR>
35. <RZR><CIF> **** <RZR>
36. <MAN>B **** <LIT>
37. 1 **** <MAN>
38. <MAN>0 **** <MAN>
39. <MAN>1 **** <MAN>
40. <IPE>=<AVI>; **** <OPA>
41. <LIT> **** <AVI>
42. <IPE> **** <AVI>
43. <AVI><ZNK><LIT> **** <AVI>
44. <AVI><ZNK><IPE> **** <AVI>
45. + **** <ZNK>
46. - **** <ZNK>
47. END\_<IPR>; **** <OEN>
48. **<CIF>  <DCF>**
49. **<DCF><CIF>  <DCF>**

**Рис. 5**

Теперь, просматривая каждую из продукций слева-направо, сгруппируем продукции, имеющие общие части в "кусты", в которых роль "ствола" играют общие части продукций, а роль "ветвей" – различающиеся части продукций. Получим следующий результат:

1. <OPR><TEL><OEN> **** <PRO>
2. <IPR>:PROC\_OPTIONS(MAIN); **** <OPR>
3. <IDE> **** <IPR>

└**** <BUK> **** <IDE>

└**** <CIF> **** <IDE>

└**** <IPE>

1. <BUK> **** <IDE>
2. A **** <BUK>
3. B **** <BUK>
4. C **** <BUK>
5. D **** <BUK>

❘

└──****CL\_<IDE>\_BIN\_FIXED(<RZR>); **** <ODC>

| └⟶ INIT(<LIT>); **** <ODC>

**|**

**└ CHAR(RZR);  <ODC>**

**|**

**└ DEC\_FIXED\_INIT(<DCF>);  <ODC>**

**|**

**└ BIT(<RZR>);  <ODC>**

1. E **** <BUK>

└**** ND\_<IPR>; **** <OEN>

1. M **** <BUK>
2. P **** <BUK>
3. X **** <BUK>
4. 0 **** <CIF>
5. 1 **** <CIF>

└**** <MAN>

1. 2 **** <CIF>
2. 3 **** <CIF>
3. 4 **** <CIF>
4. 5 **** <CIF>
5. 6 **** <CIF>
6. 7 **** <CIF>
7. 8 **** <CIF>
8. 9 **** <CIF>
9. <ODC> **** <TEL>
10. <TEL> **** <ODC> **** <TEL>

└**** <OPA> **** <TEL>

1. <CIF> **** <RZR>

└** <DCF>**

1. <RZR><CIF> **** <RZR>
2. <MAN>B **** <LIT>

└**** 0 **** <MAN>

└**** 1 **** <MAN>

1. <IPE>=<AVI>; **** <OPA>

└**** <AVI>

1. <LIT> **** <AVI>
2. <AVI><ZNK> **** <LIT> **** <AVI>

└**** <IPE> **** <AVI>

1. + **** <ZNK>
2. - **** <ZNK>
3. **<DCF><CIF>  <DCF>**

**Рис. 6**

# Изменения в таблице продукций.

**/\* \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_**

**| NN : посл : пред : дер : альт |**

**|\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_:\_\_\_\_\_\_\_:\_\_\_\_\_\_| \*/**

**{**

**{/\*. 0 .\*/ -1 , -1 , "\*\*\*" , -1 },**

**/\* вход с символа - 0 \*/**

**{/\*. 1 .\*/ 2 , 0 , "0 " , 0 },**

**{/\*. 2 .\*/ 3 , 1 , "CIF" , 0 },**

**{/\*. 3 .\*/ 0 , 2 , "\* " , 0 },**

**/\* вход с символа - 1 \*/**

**{/\*. 4 .\*/ 5 , 0 , "1 " , 0 },**

**{/\*. 5 .\*/ 6 , 4 , "CIF" , 7 },**

**{/\*. 6 .\*/ 0 , 5 , "\* " , 0 },**

**{/\*. 7 .\*/ 8 , 4 , "MAN" , 0 },**

**{/\*. 8 .\*/ 0 , 7 , "\* " , 0 },**

**/\* вход с символа - 2 \*/**

**{/\*. 9 .\*/ 10 , 0 , "2 " , 0 },**

**{/\*. 10 .\*/ 11 , 9 , "CIF" , 0 },**

**{/\*. 11 .\*/ 0 , 10 , "\* " , 0 },**

**/\* вход с символа - 3 \*/**

**{/\*. 12 .\*/ 13 , 0 , "3 " , 0 },**

**{/\*. 13 .\*/ 14 , 12 , "CIF" , 0 },**

**{/\*. 14 .\*/ 0 , 13 , "\* " , 0 },**

**/\* вход с символа - 4 \*/**

**{/\*. 15 .\*/ 16 , 0 , "4 " , 0 },**

**{/\*. 16 .\*/ 17 , 15 , "CIF" , 0 },**

**{/\*. 17 .\*/ 0 , 16 , "\* " , 0 },**

**/\* вход с символа - 5 \*/**

**{/\*. 18 .\*/ 19 , 0 , "5 " , 0 },**

**{/\*. 19 .\*/ 20 , 18 , "CIF" , 0 },**

**{/\*. 20 .\*/ 0 , 19 , "\* " , 0 },**

**/\* вход с символа - 6 \*/**

**{/\*. 21 .\*/ 22 , 0 , "6 " , 0 },**

**{/\*. 22 .\*/ 23 , 21 , "CIF" , 0 },**

**{/\*. 23 .\*/ 0 , 22 , "\* " , 0 },**

**/\* вход с символа - 7 \*/**

**{/\*. 24 .\*/ 25 , 0 , "7 " , 0 },**

**{/\*. 25 .\*/ 26 , 24 , "CIF" , 0 },**

**{/\*. 26 .\*/ 0 , 25 , "\* " , 0 },**

**/\* вход с символа - 8 \*/**

**{/\*. 27 .\*/ 28 , 0 , "8 " , 0 },**

**{/\*. 28 .\*/ 29 , 27 , "CIF" , 0 },**

**{/\*. 29 .\*/ 0 , 28 , "\* " , 0 },**

**/\* вход с символа - 9 \*/**

**{/\*. 30 .\*/ 31 , 0 , "9 " , 0 },**

**{/\*. 31 .\*/ 32 , 30 , "CIF" , 0 },**

**{/\*. 32 .\*/ 0 , 31 , "\* " , 0 },**

**/\* вход с символа - A \*/**

**{/\*. 33 .\*/ 34 , 0 , "A " , 0 },**

**{/\*. 34 .\*/ 35 , 33 , "BUK" , 0 },**

**{/\*. 35 .\*/ 0 , 34 , "\* " , 0 },**

**/\* вход с символа - B \*/**

**{/\*. 36 .\*/ 37 , 0 , "B " , 0 },**

**{/\*. 37 .\*/ 38 , 36 , "BUK" , 0 },**

**{/\*. 38 .\*/ 0 , 37 , "\* " , 0 },**

**/\* вход с символа - C \*/**

**{/\*. 39 .\*/ 40 , 0 , "C " , 0 },**

**{/\*. 40 .\*/ 41 , 39 , "BUK" , 0 },**

**{/\*. 41 .\*/ 0 , 40 , "\* " , 0 },**

**/\* вход с символа - D \*/**

**{/\*. 42 .\*/ 43 , 0 , "D " , 0 },**

**{/\*. 43 .\*/ 44 , 42 , "BUK" , 45 },**

**{/\*. 44 .\*/ 0 , 43 , "\* " , 0 },**

**{/\*. 45 .\*/ 46 , 42 , "C " , 0 },**

**{/\*. 46 .\*/ 47 , 45 , "L " , 0 },**

**{/\*. 47 .\*/ 48 , 46 , " " , 0 },**

**{/\*. 48 .\*/ 49 , 47 , "IDE" , 0 },**

**{/\*. 49 .\*/ 50 , 48 , " " , 0 }, /\* tomilin: ancestor DEC\*/**

**{/\*. 50 .\*/ 51 , 49 , "B " , 187 }, /\* tomilin: entry point \*/**

**{/\*. 51 .\*/ 52 , 50 , "I " , 0 },**

**{/\*. 52 .\*/ 53 , 51 , "N " , 0 },**

**{/\*. 53 .\*/ 54 , 52 , " " , 0 },**

**{/\*. 54 .\*/ 55 , 53 , "F " , 0 },**

**{/\*. 55 .\*/ 56 , 54 , "I " , 0 },**

**{/\*. 56 .\*/ 57 , 55 , "X " , 0 },**

**{/\*. 57 .\*/ 58 , 56 , "E " , 0 },**

**{/\*. 58 .\*/ 59 , 57 , "D " , 0 },**

**{/\*. 59 .\*/ 60 , 58 , "( " , 0 },**

**{/\*. 60 .\*/ 61 , 59 , "RZR" , 0 },**

**{/\*. 61 .\*/ 62 , 60 , ") " , 0 },**

**{/\*. 62 .\*/ 63 , 61 , "; " , 65 },**

**{/\*. 63 .\*/ 64 , 62 , "ODC" , 0 },**

**{/\*. 64 .\*/ 0 , 63 , "\* " , 0 }, /\* ?? successor 65?? was65 \*/**

**{/\*. 65 .\*/ 66 , 61 , "I " , 0 }, /\* ?? ancestor 61?? ) \*/**

**{/\*. 66 .\*/ 67 , 65 , "N " , 0 },**

**{/\*. 67 .\*/ 68 , 66 , "I " , 0 },**

**{/\*. 68 .\*/ 69 , 67 , "T " , 0 },**

**{/\*. 69 .\*/ 70 , 68 , "( " , 0 },**

**{/\*. 70 .\*/ 71 , 69 , "LIT" , 0 },**

**{/\*. 71 .\*/ 72 , 70 , ") " , 0 },**

**{/\*. 72 .\*/ 73 , 71 , "; " , 0 },**

**{/\*. 73 .\*/ 186 , 72 , "ODC" , 0 },**

**/\* вход с символа - E \*/**

**{/\*. 74 .\*/ 75 , 0 , "E " , 0 },**

**{/\*. 75 .\*/ 76 , 74 , "N " , 82 },**

**{/\*. 76 .\*/ 77 , 75 , "D " , 0 },**

**{/\*. 77 .\*/ 78 , 76 , " " , 0 },**

**{/\*. 78 .\*/ 79 , 77 , "IPR" , 0 },**

**{/\*. 79 .\*/ 80 , 78 , "; " , 0 },**

**{/\*. 80 .\*/ 81 , 79 , "OEN" , 0 },**

**{/\*. 81 .\*/ 0 , 80 , "\* " , 0 },**

**{/\*. 82 .\*/ 83 , 74 , "BUK" , 0 },**

**{/\*. 83 .\*/ 0 , 82 , "\* " , 0 },**

**/\* вход с символа - M \*/**

**{/\*. 84 .\*/ 85 , 0 , "M " , 0 },**

**{/\*. 85 .\*/ 86 , 84 , "BUK" , 0 },**

**{/\*. 86 .\*/ 0 , 85 , "\* " , 0 },**

**/\* вход с символа - P \*/**

**{/\*. 87 .\*/ 88 , 0 , "P " , 0 },**

**{/\*. 88 .\*/ 89 , 87 , "BUK" , 0 },**

**{/\*. 89 .\*/ 0 , 88 , "\* " , 0 },**

**/\* вход с символа - X \*/**

**{/\*. 90 .\*/ 91 , 0 , "X " , 0 },**

**{/\*. 91 .\*/ 92 , 90 , "BUK" , 0 },**

**{/\*. 92 .\*/ 0 , 91 , "\* " , 0 },**

**/\* вход с символа - BUK \*/**

**{/\*. 93 .\*/ 94 , 0 , "BUK" , 0 },**

**{/\*. 94 .\*/ 95 , 93 , "IDE" , 0 },**

**{/\* 95 .\*/ 0 , 94 , "\* " , 0 },**

**/\* вход с символа - IDE \*/**

**{/\*. 96 .\*/ 97 , 0 , "IDE" , 0 },**

**{/\*. 97 .\*/ 98 , 96 , "BUK" , 100 },**

**{/\*. 98 .\*/ 99 , 97 , "IDE" , 0 },**

**{/\*. 99 .\*/ 0 , 98 , "\* " , 0 },**

**{/\*. 100 .\*/ 101 , 96 , "CIF" , 103 },**

**{/\*. 101 .\*/ 102 , 100 , "IDE" , 0 },**

**{/\*. 102 .\*/ 0 , 101 , "\* " , 0 },**

**{/\*. 103 .\*/ 104 , 96 , "IPE" , 105 },**

**{/\*. 104 .\*/ 0 , 103 , "\* " , 0 },**

**{/\*. 105 .\*/ 106 , 96 , "IPR" , 0 },**

**{/\*. 106 .\*/ 0 , 105 , "\* " , 0 },**

**/\* вход с символа - + \*/**

**{/\*. 107 .\*/ 108 , 0 , "+ " , 0 },**

**{/\*. 108 .\*/ 109 , 107 , "ZNK" , 0 },**

**{/\*. 109 .\*/ 0 , 108 , "\* " , 0 },**

**/\* вход с символа - - \*/**

**{/\*. 110 .\*/ 111 , 0 , "- " , 0 },**

**{/\*. 111 .\*/ 112 , 110 , "ZNK" , 0 },**

**{/\*. 112 .\*/ 0 , 111 , "\* " , 0 },**

**/\* вход с символа - IPR \*/**

**{/\*. 113 .\*/ 114 , 0 , "IPR" , 0 },**

**{/\*. 114 .\*/ 115 , 113 , ": " , 0 },**

**{/\*. 115 .\*/ 116 , 114 , "P " , 0 },**

**{/\*. 116 .\*/ 117 , 115 , "R " , 0 },**

**{/\*. 117 .\*/ 118 , 116 , "O " , 0 },**

**{/\*. 118 .\*/ 119 , 117 , "C " , 0 },**

**{/\*. 119 .\*/ 120 , 118 , " " , 0 },**

**{/\*. 120 .\*/ 121 , 119 , "O " , 0 },**

**{/\*. 121 .\*/ 122 , 120 , "P " , 0 },**

**{/\*. 122 .\*/ 123 , 121 , "T " , 0 },**

**{/\*. 123 .\*/ 124 , 122 , "I " , 0 },**

**{/\*. 124 .\*/ 125 , 123 , "O " , 0 },**

**{/\*. 125 .\*/ 126 , 124 , "N " , 0 },**

**{/\*. 126 .\*/ 127 , 125 , "S " , 0 },**

**{/\*. 127 .\*/ 128 , 126 , "( " , 0 },**

**{/\*. 128 .\*/ 129 , 127 , "M " , 0 },**

**{/\*. 129 .\*/ 130 , 128 , "A " , 0 },**

**{/\*. 130 .\*/ 131 , 129 , "I " , 0 },**

**{/\*. 131 .\*/ 132 , 130 , "N " , 0 },**

**{/\*. 132 .\*/ 133 , 131 , ") " , 0 },**

**{/\*. 133 .\*/ 134 , 132 , "; " , 0 },**

**{/\*. 134 .\*/ 135 , 133 , "OPR" , 0 },**

**{/\*. 135 .\*/ 0 , 134 , "\* " , 0 },**

**/\* вход с символа - CIF \*/**

**{/\*. 136 .\*/ 137 , 0 , "CIF" , 0 },**

**{/\*. 137 .\*/ 138 , 136 , "RZR" , 234 }, /\* tomilin: entry point \*/**

**{/\*. 138 .\*/ 0 , 137 , "\* " , 0 }, /\* ancestor was 0 \*/**

**/\* вход с символа - RZR \*/**

**{/\*. 139 .\*/ 140 , 0 , "RZR" , 0 },**

**{/\*. 140 .\*/ 141 , 139 , "CIF" , 0 },**

**{/\*. 141 .\*/ 142 , 140 , "RZR" , 0 },**

**{/\*. 142 .\*/ 0 , 141 , "\* " , 0 },**

**/\* вход с символа - MAN \*/**

**{/\*. 143 .\*/ 144 , 0 , "MAN" , 0 },**

**{/\*. 144 .\*/ 145 , 143 , "B " , 147 },**

**{/\*. 145 .\*/ 146 , 144 , "LIT" , 0 },**

**{/\*. 146 .\*/ 0 , 145 , "\* " , 0 },**

**{/\*. 147 .\*/ 148 , 143 , "0 " , 150 },**

**{/\*. 148 .\*/ 149 , 147 , "MAN" , 0 },**

**{/\*. 149 .\*/ 0 , 148 , "\* " , 0 },**

**{/\*. 150 .\*/ 151 , 143 , "1 " , 0 },**

**{/\*. 151 .\*/ 152 , 150 , "MAN" , 0 },**

**{/\*. 152 .\*/ 0 , 151 , "\* " , 0 },**

**/\* вход с символа - IPE \*/**

**{/\*. 153 .\*/ 154 , 0 , "IPE" , 0 },**

**{/\*. 154 .\*/ 155 , 153 , "= " , 159 },**

**{/\*. 155 .\*/ 156 , 154 , "AVI" , 0 },**

**{/\*. 156 .\*/ 157 , 155 , "; " , 0 },**

**{/\*. 157 .\*/ 158 , 156 , "OPA" , 0 },**

**{/\*. 158 .\*/ 0 , 157 , "\* " , 0 },**

**{/\*. 159 .\*/ 160 , 153 , "AVI" , 0 },**

**{/\*. 160 .\*/ 0 , 159 , "\* " , 0 },**

**/\* вход с символа - LIT \*/**

**{/\*. 161 .\*/ 162 , 0 , "LIT" , 0 },**

**{/\*. 162 .\*/ 163 , 161 , "AVI" , 0 },**

**{/\*. 163 .\*/ 0 , 162 , "\* " , 0 },**

**/\*. вход с символа - AVI \*/**

**{/\*. 164 .\*/ 165 , 0 , "AVI" , 0 },**

**{/\*. 165 .\*/ 166 , 164 , "ZNK" , 0 },**

**{/\*. 166 .\*/ 167 , 165 , "LIT" , 168 },**

**{/\*. 167 .\*/ 197 , 166 , "AVI" , 0 },**

**{/\*. 168 .\*/ 169 , 165 , "IPE" , 0 },**

**{/\*. 169 .\*/ 170 , 168 , "AVI" , 0 },**

**{/\*. 170 .\*/ 0 , 169 , "\* " , 0 },**

**/\* вход с символа - OPR \*/**

**{/\*. 171 .\*/ 172 , 0 , "OPR" , 0 },**

**{/\*. 172 .\*/ 173 , 171 , "TEL" , 0 },**

**{/\*. 173 .\*/ 174 , 172 , "OEN" , 0 },**

**{/\*. 174 .\*/ 175 , 173 , "PRO" , 0 },**

**{/\*. 175 .\*/ 0 , 174 , "\* " , 0 },**

**/\*. вход с символа - ODC \*/**

**{/\*. 176 .\*/ 177 , 0 , "ODC" , 0 },**

**{/\*. 177 .\*/ 178 , 176 , "TEL" , 0 },**

**{/\*. 178 .\*/ 0 , 177 , "\* " , 0 },**

**/\*. вход с символа - TEL \*/**

**{/\*. 179 .\*/ 180 , 0 , "TEL" , 0 },**

**{/\*. 180 .\*/ 181 , 179 , "ODC" , 183 },**

**{/\*. 181 .\*/ 182 , 180 , "TEL" , 0 },**

**{/\*. 182 .\*/ 0 , 181 , "\* " , 0 },**

**{/\*. 183 .\*/ 184 , 179 , "OPA" , 0 },**

**{/\*. 184 .\*/ 185 , 183 , "TEL" , 0 },**

**{/\*. 185 .\*/ 0 , 184 , "\* " , 0 },**

**{/\*. 186 .\*/ 0 , 73 , "\* " , 0 },**

**{/\*. 187 .\*/ 188 , 49 , "C " , 201 }, /\* tomilin: entry point \*/**

**{/\*. 188 .\*/ 189 , 187 , "H " , 0 },**

**{/\*. 189 .\*/ 190 , 188 , "A " , 0 },**

**{/\*. 190 .\*/ 191 , 189 , "R " , 0 },**

**{/\*. 191 .\*/ 192 , 190 , "( " , 0 },**

**{/\*. 192 .\*/ 193 , 191 , "RZR" , 0 },**

**{/\*. 193 .\*/ 194 , 192 , ") " , 0 },**

**{/\*. 194 .\*/ 195 , 193 , "; " , 0 },**

**{/\*. 195 .\*/ 196 , 194 , "ODC" , 0 },**

**{/\*. 196 .\*/ 0 , 195 , "\* " , 0 },**

**{/\*. 197 .\*/ 0 , 167 , "\* " , 0 },**

**{/\*. 198 .\*/ 199 , 0 , "\* " , 0 },**

**{/\*. 199 .\*/ 200 , 198 , "ZNK" , 0 },**

**{/\*. 200 .\*/ 0 , 199 , "\* " , 0 },**

**{/\*. 201 .\*/ 202 , 49 , "D " , 221 }, /\* tomilin: DEC\_FIXED... \*/**

**{/\*. 202 .\*/ 203 , 201 , "E " , 0 },**

**{/\*. 203 .\*/ 204 , 202 , "C " , 0 },**

**{/\*. 204 .\*/ 205 , 203 , " " , 0 },**

**{/\*. 205 .\*/ 206 , 204 , "F " , 0 },**

**{/\*. 206 .\*/ 207 , 205 , "I " , 0 },**

**{/\*. 207 .\*/ 208 , 206 , "X " , 0 },**

**{/\*. 208 .\*/ 209 , 207 , "E " , 0 },**

**{/\*. 209 .\*/ 210 , 208 , "D " , 0 },**

**{/\*. 210 .\*/ 211 , 209 , " " , 0 },**

**{/\*. 211 .\*/ 212 , 210 , "I " , 0 },**

**{/\*. 212 .\*/ 213 , 211 , "N " , 0 },**

**{/\*. 213 .\*/ 214 , 212 , "I " , 0 },**

**{/\*. 214 .\*/ 215 , 213 , "T " , 0 },**

**{/\*. 215 .\*/ 216 , 214 , "( " , 0 },**

**{/\*. 216 .\*/ 217 , 215 , "DCF" , 0 },**

**{/\*. 217 .\*/ 218 , 216 , ") " , 0 },**

**{/\*. 218 .\*/ 219 , 217 , "; " , 0 },**

**{/\*. 219 .\*/ 220 , 218 , "ODC" , 0 },**

**{/\*. 220 .\*/ 0 , 219 , "\* " , 0 },**

**{/\*. 221 .\*/ 222 , 49 , "B " , 0 }, /\* tomilin: BIT(INIT... \*/**

**{/\*. 222 .\*/ 223 , 221 , "I " , 0 },**

**{/\*. 223 .\*/ 224 , 222 , "T " , 0 },**

**{/\*. 224 .\*/ 225 , 223 , "( " , 0 },**

**{/\*. 225 .\*/ 226 , 224 , "RZR" , 0 },**

**{/\*. 226 .\*/ 227 , 225 , ") " , 0 },**

**{/\*. 227 .\*/ 228 , 226 , "; " , 0 },**

**{/\*. 228 .\*/ 229 , 227 , "ODC" , 0 },**

**{/\*. 229 .\*/ 0 , 228 , "\* " , 0 },**

**{/\*. 230 .\*/ 231 , 0 , "DCF" , 0 }, /\* tomilin: DEC\_FIXED... \*/**

**{/\*. 231 .\*/ 232 , 230 , "CIF" , 0 },**

**{/\*. 232 .\*/ 233 , 231 , "DCF" , 0 },**

**{/\*. 233 .\*/ 0 , 232 , "\* " , 0 },**

**{/\*. 234 .\*/ 235 , 136 , "DCF" , 0 }, /\* tomilin: DEC\_FIXED... \*/**

**{/\*. 235 .\*/ 0 , 234 , "\* " , 0 }**

**};**

Изменения в таблице входов.

**struct**

**{**

**char SYM [4];**

**int VX;**

**char TYP;**

**} VXOD [ NVXOD ] =**

**/\* \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_**

**| NN | символ | вход| тип |**

**|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\_| \*/**

**{**

**{/\*. 1 .\*/ "AVI" , 164 , 'N' },**

**{/\*. 2 .\*/ "BUK" , 93 , 'N' },**

**{/\*. 3 .\*/ "CIF" , 136 , 'N' },**

**{/\*. 4 .\*/ "IDE" , 96 , 'N' },**

**{/\*. 5 .\*/ "IPE" , 153 , 'N' },**

**{/\*. 6 .\*/ "IPR" , 113 , 'N' },**

**{/\*. 7 .\*/ "LIT" , 161 , 'N' },**

**{/\*. 8 .\*/ "MAN" , 143 , 'N' },**

**{/\*. 9 .\*/ "ODC" , 176 , 'N' },**

**{/\*. 10 .\*/ "OEN" , 0 , 'N' },**

**{/\*. 11 .\*/ "OPA" , 0 , 'N' },**

**{/\*. 12 .\*/ "OPR" , 171 , 'N' },**

**{/\*. 13 .\*/ "PRO" , 0 , 'N' },**

**{/\*. 14 .\*/ "RZR" , 139 , 'N' },**

**{/\*. 15 .\*/ "TEL" , 179 , 'N' },**

**{/\*. 16 .\*/ "ZNK" , 0 , 'N' },**

**{/\*. 17 .\*/ "DCF" , 234 , 'N' },**

**{/\*. 18 .\*/ "A " , 33 , 'T' },**

**{/\*. 19 .\*/ "B " , 36 , 'T' },**

**{/\*. 20 .\*/ "C " , 39 , 'T' },**

**{/\*. 21 .\*/ "D " , 42 , 'T' },**

**{/\*. 22 .\*/ "E " , 74 , 'T' },**

**{/\*. 23 .\*/ "M " , 84 , 'T' },**

**{/\*. 24 .\*/ "P " , 87 , 'T' },**

**{/\*. 25 .\*/ "X " , 90 , 'T' },**

**{/\*. 26 .\*/ "0 " , 1 , 'T' },**

**{/\*. 27 .\*/ "1 " , 4 , 'T' },**

**{/\*. 28 .\*/ "2 " , 9 , 'T' },**

**{/\*. 29 .\*/ "3 " , 12 , 'T' },**

**{/\*. 30 .\*/ "4 " , 15 , 'T' },**

**{/\*. 31 .\*/ "5 " , 18 , 'T' },**

**{/\*. 32 .\*/ "6 " , 21 , 'T' },**

**{/\*. 33 .\*/ "7 " , 24 , 'T' },**

**{/\*. 34 .\*/ "8 " , 27 , 'T' },**

**{/\*. 35 .\*/ "9 " , 30 , 'T' },**

**{/\*. 36 .\*/ "+ " , 107 , 'T' },**

**{/\*. 37 .\*/ "- " , 110 , 'T' },**

**{/\*. 38 .\*/ ": " , 0 , 'T' },**

**{/\*. 39 .\*/ "I " , 0 , 'T' },**

**{/\*. 40 .\*/ "R " , 0 , 'T' },**

**{/\*. 41 .\*/ "N " , 0 , 'T' },**

**{/\*. 42 .\*/ "O " , 0 , 'T' },**

**{/\*. 43 .\*/ "T " , 0 , 'T' },**

**{/\*. 44 .\*/ "S " , 0 , 'T' },**

**{/\*. 45 .\*/ "( " , 0 , 'T' },**

**{/\*. 46 .\*/ ") " , 0 , 'T' },**

**{/\*. 47 .\*/ " " , 0 , 'T' },**

**{/\*. 48 .\*/ "; " , 0 , 'T' },**

**{/\*. 49 .\*/ "L " , 0 , 'T' },**

**{/\*. 50 .\*/ "F " , 0 , 'T' },**

**{/\*. 51 .\*/ "= " , 0 , 'T' },**

**{/\*. 52 .\*/ "H " , 0 , 'T' },**

**{/\*. 53 .\*/ "\* " , 198 , 'T' }**

**};**

Изменения в таблице матрицы смежности.

**char TPR [ NVXOD ] [ NNETRM ] =**

**{**

**/\***

**\_\_\_\_\_\_\_\_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_**

**| AVI:BUK:CIF:IDE:IPE:IPR:LIT:MAN:ODC:OEN:OPA:OPR:PRO:RZR:TEL:ZNK:DCF|**

**|\_\_\_\_\_\_\_\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_| \*/**

**{/\*AVI\*/ 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*BUK\*/ 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*CIF\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 },**

**{/\*IDE\*/ 0 , 0 , 0 , 1 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*IPE\*/ 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*IPR\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 },**

**{/\*LIT\*/ 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*MAN\*/ 0 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*ODC\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 },**

**{/\*OEN\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*OPA\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*OPR\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 },**

**{/\*PRO\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*RZR\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 },**

**{/\*TEL\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 },**

**{/\*ZNK\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\*DCF\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**/\***

**\_\_\_\_\_\_\_\_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_**

**| AVI:BUK:CIF:IDE:IPE:IPR:LIT:MAN:ODC:OEN:OPA:OPR:PRO:RZR:TEL:ZNK:DCF|**

**|\_\_\_\_\_\_\_\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_| \*/**

**{/\* A\*/ 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* B\*/ 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* C\*/ 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* D\*/ 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* E\*/ 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* M\*/ 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* P\*/ 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* X\*/ 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* 0\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* 1\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* 2\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* 3\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* 4\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* 5\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 },**

**{/\* 6\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 },**

**{/\* 7\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**/\***

**\_\_\_\_\_\_\_\_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_**

**| AVI:BUK:CIF:IDE;IPE:IPR:LIT:MAN:ODC:OEN:OPA:OPR:PRO:RZR:TEL:ZNK:DCF|**

**|\_\_\_\_\_\_\_\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_| \*/**

**{/\* 8\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* 9\*/ 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* +\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 },**

**{/\* -\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 },**

**{/\* :\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* I\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* R\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* N\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* O\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* T\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* S\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* (\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* )\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* ;\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* \*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**{/\* L\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 },**

**/\***

**\_\_\_\_\_\_\_\_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_**

**| AVI:BUK:CIF:IDE:IPE:IPR:LIT:MAN:ODC:OEN;OPA:OPR:PR0:RZR:TEL:ZNK:DCF|**

**|\_\_\_\_\_\_\_\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_:\_\_\_| \*/**

**{/\* F\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0, 0 },**

**{/\* =\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0, 0 },**

**{/\* H\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0, 0 },**

**{/\* \*\*/ 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1, 0 }**

**/\*|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| \*/**

**};**

# 4. Модификация алгоритма макета компилятора с ЯВУ

Модификация алгоритма макета компилятора с ЯВУ была произведена параметрически путем модификации той части параметров настройки компилятора ЯВУ, которая относится к семантике ЯВУ. Этой частью семантических параметров является библиотека семантических подпрограмм. Объем модификаций этой библиотеки в общем случае определяется объемом модификации синтаксиса ЯВУ.

Модификации синтаксиса затронули или породили следующие семантические подпрограммы: DCF1, DCF2, ODC1, AVI2, OEN2, OPA2, OPR2, RZR2

В подпрограмму DCF1 изменения внесены не были**.**

Изменения в подпрограмме **ODC1** для введения новых типов по заданию:

**int ODC1() {**

**…..**

**else if ( !strcmp ( FORMT [2], "DEC" ) && !strcmp ( FORMT [3], "FIXED" ) )**

**{**

**SYM [ISYM].TYPE = 'D';**

**}**

**else if ( !strcmp ( FORMT [2], "BIT" ) )**

**{**

**SYM [ISYM].TYPE = 'b';**

**}**

**…..**

**}**

Изменения в подпрограмме **AVI2** для внесения изменений связанных с командой MVC:

**int AVI2() {**

**….**

**char op[] = "@BUF+2(6),";**

**strcat(op, SYM[i].NAME);**

**ZKARD2("", "MVC", op, "Перемещаем из области памяти A в @BUF" );**

**…**

**ZKARD2("", "RRAB,", FORMT [IFORMT-1], "Формирование промежуточного значения" );**

**….**

**}**

Изменения в подпрограмме **OEN2** для описания выхода из программы, объявленных переменных и меток:

**int OEN2 () *//* *эпилог***

**{**

**char RAB [20];**

**char i = 0;**

**FORM ();**

**ZKARD2( "", "BCR", "15,@RVIX", "Выход из программы" );**

***….***

**memcpy ( *ASSTXT* *[IASSTXT++]*, DCF\_CARD.BUFCARD, *sizeof*(DCF\_CARD.BUFCARD) );**

**memcpy ( *ASSTXT* *[IASSTXT++]*, RZR\_CARD.BUFCARD, *sizeof*(RZR\_CARD.BUFCARD) );**

**ZKARD2("", "DS", "0F", "Выравнивание памяти для CVB");**

**ZKARD2("@BUF", "DC", "PL8\'0\'", "Превращаем 11разрядов в 15");**

**ZKARD2("@PTR", "DC", "BL32\'1\'", "Указатель");**

**ZKARD2("@NULL", "DC", "F\'0\'", "Заводим const 0 для сравнения");**

**ZKARD2("@RRAB", "EQU", "3", " @RRAB1 назначим 3");**

**ZKARD2("@RRAB1", "EQU", "4", " @RRAB1 назначим 4");**

**ZKARD2("@RBASE", "EQU", "15", " @RBASE назначим 15");**

**ZKARD2("@RVIX", "EQU", "14", " @RVIX назначим 14");**

**ZKARD2( "", "END", "", "Конец текста блока" );**

***….***

**}**

Изменения в подпрограмме **OPA2** для работы с преобразованием типа через присваивание:

**int OPA2() {**

**….**

**else if ( SYM[i].TYPE == 'b' ) // В случае типа DEC FIXED**

**{**

**ZKARD2("", "CVB", "@RRAB,@BUF", "Преобразование");**

**ZKARD2("@LBL", "LR", "@RRAB1,@RRAB", "Копирование содержимого в @RRAB1из @RRAB");**

**ZKARD2(" ", "N", "@RRAB1,@PTR", "Поразрядная операция И");**

**ZKARD2(" ", "C", "@RRAB1,@NULL", "Сравнение рабочего регистра с нулевой const");**

**ZKARD2("", "BC", "6,@CONT", "Проверка маски для меньше или больше");**

**ZKARD2("", "SLL", "@RRAB,1", "Смещаем влево на 1 бит");**

**ZKARD2("", "BC", "15,@LBL", "Замыкание метки цикла");**

**ZKARD2("@CONT", "SRL", "@RRAB,16", "Метка в которую выпрыгиваем,когда в BCединица затем смещаем на 16вправо");**

**char op[] = "@RRAB,";**

**strcat( op, SYM[i].NAME );**

**ZKARD2("", "STH", op, "Сброс полуслова");**

**return 0;**

**}**

**…..**

**}**

Изменения в подпрограмме **OPR2** добавлены символы @:

**int OPR2() {**

**….**

**ZKARD2( ASS\_CARD.\_BUFCARD.METKA, "START", "0", "Начало программы" );**

**ZKARD2( "", "BALR", "@RBASE,0", "Загрузка регистра @RBASE" );**

**ZKARD2( "", "USING", "\*,@RBASE", "Назначить регистр базой" );**

**….**

**}**

Изменения в подпрограмме **RZR2** для обработки разрядов типа BIT:

**int RZR2 ()**

**{**

**char RAB [20];**

**char i = 0;**

**FORM ();**

***if* ( is\_fv == 't' )**

***return* 0;**

***for* ( i = 0; i < ISYM; i++ )**

**{**

***if* ( isalpha ( SYM [i].NAME [0] ) )**

**{**

***if* ( SYM[i].TYPE == 'b' )**

**{**

**char op[] = "BL";**

**strcat( *op*, SYM[i].RAZR );**

**ZKARD\_RZR(SYM[i].NAME, "DS", op, "Выделение памяти");**

**is\_fv = 't';**

**}**

**}**

**}**

***return* 0;**

**}**

Изменения в подпрограмме **DCF2** для обработки десятичных чисел типа DEC FIXED:

**int DCF2 ()**

**{**

**char RAB [20];**

**char i = 0;**

**FORM ();**

**printf( "DCF\_2\_%i\n", COUNTER++ );**

***for* ( i = 0; i < ISYM; i++ )**

**{**

***if* ( isalpha ( SYM [i].NAME [0] ) )**

**{**

***if* ( SYM[i].TYPE == 'D' ) *//* *Если* *тип* *DEC* *FIXED,* *тогда***

**{**

**char op[] = "PL6\'";**

**strcat( *op*, SYM[i].INIT );**

**strcat( *op*, "\'" );**

**ZKARD\_DCF(SYM[i].NAME, "DC", op, "Инициализация переменной А");**

**}**

**}**

**}**

***return* 0;**

**}**

# 5. Результаты экспериментальной проверки модифицированного компилятора с ЯВУ

Работоспособность модифицированного под персональный вариант макета компилятора с ЯВУ была подтверждена испытательным прогоном его в среде ОС Linux (ubuntu 14.04). В результате на выходе был получен текст на Ассемблере, рис. 7, который совпал с текстом, приведенным на рис. 4

**Рис. 7**

# 6. Выводы по результатам работы

Порученное задание выполнено успешно. Успешность выполнения порученного задания подтверждена полным совпадением выходного текста, полученного при экспериментальной проверке компилятора с ЯВУ, с эталоном на рис. 4.