Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №2**

З дисципліни «СП»

Виконав: Перевірив:

Студент групи ІО-21 доц. Пустоваров В.І.

Коноз А.О.

Дата здачі\_\_\_\_\_\_\_\_\_\_\_\_\_

Захищено з балом\_\_\_\_\_

Київ 2014

**Завдання**

|  |  |
| --- | --- |
| switch(c){case 0: b=2\*a[n]; break; default: b=d;} | С |

**Лістинг коду**

#include "stdafx.h"

#include <stdlib.h>

#include "visgrp.h"

#include "token.h"

#include "automat.h"

char \*imgs[100] = {"b", "a[n]", "1", "f", "c", "g", "k", "ky", "n",

"nD", "nU", "el", "clr", "red", "green", "blue", "0", "2", "d"};

struct lxNode pic4[] = // switch(c) {case 0: b = 2\*a[n]; break; default: b = d;}

{

{ \_nam, (struct lxNode\*)imgs[4], NULL, 0, 0, 0, 0, 0, NULL, 0}, // 0 + c

{ \_case, &pic4[0], &pic4[2], 0, 0, 0, 0, 0, NULL, 0}, // 1

{ \_nam, (struct lxNode\*)imgs[16], NULL, 0, 0, 0, 0, 0, NULL, 0}, // 2 + 0

{ \_cln, &pic4[1], &pic4[9], 0, 0, 0, 0, 0, NULL, 0}, // 3

{ \_nam, (struct lxNode\*)imgs[0], NULL, 0, 0, 0, 0, 0, NULL, 0}, // 4 + b

{ \_ass, &pic4[4], &pic4[7], 0, 0, 0, 0, 0, NULL, 0}, // 5 + b = 2\*a[n]

{ \_nam, (struct lxNode\*)imgs[17], NULL, 0, 0, 0, 0, 0, NULL, 0}, // 6 + 2

{ \_mul, &pic4[6], &pic4[8], 0, 0, 0, 0, 0, NULL, 0}, // 7 + 2\*a[n]

{ \_nam, (struct lxNode\*)imgs[1], NULL, 0, 0, 0, 0, 0, NULL, 0}, // 8 + a[n]

};

extern char \*oprtrC[], \*oprtrP[], \*oprtrV[], \*cprC[], \*cprP[], \*cprV[];

char \*\*oprtr = oprtrC, \*\*cpr = cprC,

modeP = 0, // тип роздільника операторних дужок для Паскаля

modeC = 1, // тип роздільника операторних дужок для С

modeL = modeP;

int n = 3, b, a[3] = {0, 1, 2};

int main(int argc, char\* argv[])

{

do

{

b += a[--n];

}

while(n);

prLxTxt(pic4 + 16);

printf("\n");

enum autSgn ASgn[10] = {sg3, sg5, sg4};

for (n = 0; n < 3; n++)

printf("%5d->%2d ", ASgn[n], nxtStat(ASgn[n]));

return 0;

}

#include "stdafx.h"

#include "automat.h"

enum autStat nxtSts[Se+1][sgE+1] =

{

{S0, S0, S0, S7, S0, S0, S0, S0, S0, S0}, // для S0

{S0, S0, S2, S3, S4, S5, S6, Se, S3, S3}, // S1

{Se, Se, Se, Se, Se, Se, Se, Se, Se, Se}, // S2

{Se, Se, Se, Se, Se, Se, Se, Se, Se, Se}, // S3

{Se, Se, Se, Se, Se, Se, Se, Se, Se, Se},

{Se, Se, Se, Se, Se, Se, Se, Se, Se, Se},

{S1, S1, S2, S3, S4, S5, S6, Se, S3, S3},

{S5, S5, S5, S5, S5, S5, S5, S5, S5, S5},

{S5, S5, S5, S5, S5, S5, S5, S5, S5, S5}, // Se

};

enum autStat nxtStat(enum autSgn sgn)

{

static enum autStat s = S0; // поточний стан лексеми

return s = nxtSts[s][sgn]; // новий стан лексеми

}

#define begOprtr 0x50+46+78 // зміщення початку виконавчих операторів

#define \_lVlu 0x1000 // lValue

enum tokType

{\_nil, \_nam, //0 зовнішнє подання

\_srcn, \_cnst, //2 вхідне і внутрішнє кодування константи

\_if,\_then,\_else,\_elseif, //4 if then else elseif

\_case, \_switch, \_default, \_endcase,//8 case switch defualt endcase

\_break, \_return, \_whileP, \_whileN, //12 break return while do

\_continue, \_repeat, \_untilN, \_endloop, //16 continue repeat until

\_for, \_to, \_downto, \_step,// for to downto step

\_untilP, \_loop, \_with, \_endif,

\_void,\_extern,\_var,\_const,\_enum,\_struct/\*\_record\*/,\_union,\_register,//

\_unsigned,\_signed,\_char,\_short,\_int,\_long,\_sint64,\_uint64,//

\_float,\_double, \_label,\_auto,\_static,\_volatile,\_typedef,\_sizeof,//

\_real,\_array,\_set,\_file,\_object, \_string,\_goto,

\_program,\_function,\_procedure /\*task V\*/,

\_macromodule,\_primitive,\_specify,\_table, //Verilog

\_generate,\_config,\_liblist,\_library, //Verilog

\_incdir,\_include,\_design,\_defaultS,\_instance,\_cell,\_use, //Verilog

\_automatic,\_endmodule,\_endfunction,\_endtask, //Verilog

\_endprimitive,\_endspecify,\_endtable,\_endgenerate,\_endconfig, //Verilog

\_endcaseV,\_casex,\_casez,\_wait,\_forever,\_disable,\_ifnone, //Verilog

\_pulsestyle\_onevent,\_pulsestyle\_ondetect,\_showcanceled,\_noshowcanceled, //Verilog

\_vectored,\_scalared,\_small,\_medium,\_large, //Verilog

\_genvar,\_parameter,\_localparam,\_defparam,\_specparam,\_PATHPULSE$, //Verilog

\_inlineF,\_forward,\_interrupt,\_exportF,\_extrn,\_asmb,

\_input,\_output,\_inout, //Verilog|SQL+3

\_objectP,\_constructor,\_desctructor,\_property,\_resP,\_abstract, //P++9

\_class,\_public,\_private,\_protected,\_virtual,\_friend, //C++16

\_new,\_delete,\_tryC,\_catch,\_throw/\*raise\*/, //C++20

\_initial,\_always,\_assign,\_deassign,\_force,\_release, //Verilog+26

\_reg,\_time,\_realtime,\_event,\_buf,\_not, //Verilog+32

\_andG,\_orG,\_xorG,\_nandG,\_norG,\_xnorG, //Verilog+38

\_tran,\_tranif0,\_tranif1,\_rtran,\_rtranif0,\_rtranif1, //Verilog+44

\_tri,\_trior,\_triand,\_trireg,\_tri0,\_tri1,//Verilog+50

\_wire,\_wand,\_wor,\_wres, //Verilog+54

\_supply0,\_supply1,\_highz0,\_highz1, //Verilog+58

\_strong0,\_strong1,\_pull0,\_pull1,\_weak0,\_weak1, //Verilog+64

\_pulldown,\_pullup,\_bufif0,\_bufif1,\_notif0,\_notif1, //Verilog+70

\_cmos,\_rcmos,\_nmos,\_pmos,\_rnmos,\_rpmos, //Verilog+76

\_fork, \_join, // відкриті і закриті дужки паралельних операторів 2

\_opbr, \_ocbr, // відкриті і закриті дужки операторів 2

\_ctbr, \_fcbr, // відкриті і закриті дужки конкатенацій 3

\_ixbr, \_scbr, // відкриті і закриті дужки індексу 4

\_brkt, \_bckt, // відкриті і закриті дужки порядку і функцій 5

\_tdbr, \_tcbr, // відкриті і закриті дужки даних 6

\_eosP, eosS, // паралельні та послідовні

\_EOS=begOprtr, \_comma, \_cln, \_qmrk,// ; , : ?

\_asOr, \_asAnd, \_asXor, \_asAdd, //|= =& =^ =+

\_asSub, \_asMul, \_asDiv, \_asMod, // -= \*= /= %=

\_asShr,\_asShl, \_ass, \_dcr, \_inr, // <<= >>= = -- ++

\_dcrN,\_inrN,\_mcrs,\_dbcln,\_eoCm,\_EOF, //-- ++ // # :: \*/

\_lt,\_le, \_eq, \_ne, \_ge,\_gt, // < <= == != >= >

\_add, \_sub, \_mul, \_div, \_fldDt, \_fldPt,// + - \* / . ->

\_pwr, \_shLfa, \_shRga, \_eqB, \_neB, // \*\* <<< >>> === !==

\_addU,\_subU,\_refU, \_ptrU, // + - \* & унарні

\_lmts,\_eqar,\_astar,\_trasand, // PV+4 .. => \*> &&&

\_orR,\_andR,\_xorR,\_norR,\_nandR,\_nxorR,\_xornR, //V+11 & | ^ ~| ~& ~^

\_delay,\_eventV,\_events, //V+14 # @ @\*

\_norB,\_nandB,\_nxorB,\_xornB,\_addr, //~| ~& ~^ ^~ \_ptr,

\_rem,\_remL, //

\_mod, \_orB, \_andB, \_xorB, // %(mod) |(or) &(and) ^(xor)

\_shLft,\_shRgt, \_or, \_and, //<<(shl) >>(shr) ||(or) &&(and)

\_xmrk,\_invB,\_divI,\_in, //\_not, \_notB, /(div)

\_posedge,\_negedge,\_orE, //Verilog+3

\_frkz, // відкриті і закриті дужки паралельних операторів 2

\_opbz, // відкриті і закриті дужки операторів 2

\_ctbz, // відкриті і закриті дужки конкатенацій 3

\_ixbz, // відкриті і закриті дужки індексу 4

\_brkz, // відкриті і закриті дужки порядку і функцій 5

\_tdbz // відкриті і закриті дужки даних 6

// \_pnil

};

struct lxNode//вузол дерева, САГ або УСГ

{enum tokType ndOp; //код типу лексеми

struct lxNode\* prvNd;// зв'язок з попередником

struct lxNode\* pstNd;// зв'язок з наступником

int dataType; // код типу даних, які повертаються

unsigned resLength; //довжина результату

int x, y, f;//координати розміщення у вхідному файлі

int prnNd;//struct lxNode\* prnNd;//зв'язок з батьківським вузлом

unsigned stkLength;//довжина стека обробки семантики або номер модуля

};

// порівняння за відношенням порядку

int cmpKys(struct lxNode\*, struct lxNode\*);

**Результат виконання програми:**

Microsoft Windows [Version 6.1.7601]

Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>E:\Lab2\Debug\Lab2.exe

switch(c)

{switch 0:

{ b=2\*a[n];

break;

};

defualt:b=d;

}

3-> 7 5-> 5 4-> 8

C:\Users\User>

**Висновок:**

Під час виконання лабораторної роботи були вивчені методи створення вузлів дерево­по­діб­них та ієрархічних графів, організації доступу до інформації, ре­кон­струкції вхідного тексту та скорочення графів через вилу­чення повторних вузлів.