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 №7**

З дисципліни «СП»

Виконав: Перевірив:

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

Коноз А.О.

Дата здачі\_\_\_\_\_\_\_\_\_\_\_\_\_

Захищено з балом\_\_\_\_\_

Київ 2014

**Завдання**

double b, a[4]={2,3,4,5};

unsigned n=1,d=5;

b=d!=0;

b=2\*a[n];

**Лістинг коду**

// spLb8.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "..\spLb3\token.h"

#include "..\spLb3\visgrp.h"

#include "..\spLb4\tables.h"

#include "..\spLb4\lexan.h"

#include "..\spLb5\syntaxP.h"

#include "..\spLb4\langio.h"

#include "..\spLb7\seman.h"

#include "interp.h"

#include <stdio.h>

#include <string.h>

extern struct recrdKWD tablKWD[];

extern struct lxNode nodes[MAX\_NODES]; // масив приймач вузлів дерева

extern enum ltrType ltClsC[256];

extern enum ltrType ltClsP[256];

extern enum tokType dlCdsC[256];

extern enum tokType dlCdsP[256];

extern enum ltrType ltClsC[256];

extern enum ltrType ltClsP[256];

enum ltrType \*ltCls=ltClsC;

enum tokType \*dlCds=dlCdsC;

char file\_name[20];

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

{int nn=-1, nr=0, nc=1; //np,

if (argc>1)

{strcpy(file\_name,argv[1]);

printf("Processing file -- %s\n",file\_name);}

else

{printf("Please enter file Name: ");

scanf("%s",file\_name);

strcat(file\_name,".h");

}

opFls(file\_name);

LxAnInit('C');

// srtBin(tablKWD, 67);

do{//np=nn;

nn=LxAnlzr();

}while(nodes[nn].ndOp!=\_EOF);

prLaTxt(nodes,nn);

printf("\n");

SxAnInit('C');

nr=0; nc=1; nodes[0].prnNd=-1;

do nr=nxtProd(nodes,nr,nc);

while(++nc<nn);

// конверсія до семантичної обробки

prLxTxt(nodes+nr);

printf("\n");

// nr=prCmpr(nodes,nn,nr);// компресія для скорочення графа

SmAnlzr(nodes+nr,nr);

prLxTxt(nodes+nr);

printf("\n");

SmIntrp(nodes+nr,1);

printf("\n");

return 0;

}

#include "stdafx.h"

#include "syntaxP.h"

enum tokPrec opPrFC[290]=

{nil,ptrm,//\_nil, \_nam, //0 зовнішнє подання

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

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

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

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

pbkw,prep,nil,nil,//\_continue, \_repeat, \_untilN, \_endloop, //16 continue repeat until

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

nil,nil,pbkw,nil,//\_untilP, \_loop, \_with, \_endif,

pbkw,pbkw,nil,pbkw, pekw,pekw,pekw,pbkw,

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

pbkw,pbkw,pbkw,pbkw, pbkw,pbkw,pbkw,pbkw,

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

pbkw,pbkw,nil,pbkw, pbkw,nil,pbkw,nil,

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

nil,nil,nil,nil, nil,nil,nil,

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

nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil, nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil, nil,

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

nil,nil,nil,nil, nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil, nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

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

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

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

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

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

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

peos,peos,//\_eosP, eosS, // паралельні та послідовні

peos,pdol,psmc,pcnd,//\_EOS=begOprtr, \_comma, \_cln, \_qmrk,// ; , : ?

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

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

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

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

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

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

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

puno,puno,puno,puno,//\_addU,\_subU,\_mulU, \_andU, // + - \* & унарні

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

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

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

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

peos,peos,//\_rem,\_remL, //

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

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

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

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

popf,//\_fork, \_join, // замкнені дужки паралельних операторів 2

popf,//\_opbr, \_ocbr, // замкнені дужки операторів 2

nil,//\_ctbr, \_fcbr, // замкнені дужки конкатенацій 3

pops,//\_ixbr, \_scbr, // замкнені дужки індексу 4

popb,//\_brkt, \_bckt, // замкнені дужки порядку і функцій 5

popd//\_tdbr, \_tcbr, // замкнені дужки даних 6

// \_pnil

};

enum tokPrec opPrGC[290]=

{nil,ptrm,//\_nil, \_nam, //0 зовнішнє подання

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

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

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

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

pbkw,prep,nil,nil,//\_continue,peos \_repeat, \_untilN, \_endloop, //16 continue repeat until

pskw,peos,nil,nil,//pclf\_for, \_to, \_downto, \_step,// for to downto step

nil,nil,nil,nil,//\_untilP, \_loop, \_with, \_endif,

nil,pbkw,nil,pbkw, pekw,pekw,pekw,nil,

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

pbkw,pbkw,pbkw,pbkw, pbkw,pbkw,pbkw,pbkw,

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

pbkw,pbkw,pbkw,pbkw, pbkw,nil,nil,nil,

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

nil,nil,nil,nil, nil,nil,nil,

//\_real,\_array,\_set,\_file,\_object, \_string, \_label,

nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil, nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil, nil,

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

nil,nil,nil,nil, nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil, nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil,nil,nil,

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

nil,nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

nil,nil,nil,nil, nil,nil,

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

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

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

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

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

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

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

peos,peos,//\_eosP, eosS, // паралельні та послідовні

peos,pdol,psmc,pcnd,//\_EOS=begOprtr, \_comma, \_cln, \_qmrk,// ; , : ?

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

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

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

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

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

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

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

puno,puno,puno,puno,//\_addU,\_subU,\_mulU, \_andU, // + - \* & унарні

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

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

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

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

peos,peos,//\_rem,\_remL, //

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

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

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

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

popf,//\_fork, \_join, // замкнені дужки паралельних операторів 2

popf,//\_opbr, \_ocbr, // замкнені дужки операторів 2

nil,//\_ctbr, \_fcbr, // замкнені дужки конкатенацій 3

pops,//\_ixbr, \_scbr, // замкнені дужки індексу 4

popb,//\_brkt, \_bckt, // замкнені дужки порядку і функцій 5

popd//\_tdbr, \_tcbr, // замкнені дужки даних 6

/\* pclf,//\_fork, \_join, // замкнені дужки паралельних операторів 2

pclf,//\_opbr, \_ocbr, // замкнені дужки операторів 2

nil,//\_ctbr, \_fcbr, // замкнені дужки конкатенацій 3

pcls,//\_ixbr, \_scbr, // замкнені дужки індексу 4

pclb,//\_brkt, \_bckt, // замкнені дужки порядку і функцій 5

pcld//\_tdbr, \_tcbr, // замкнені дужки даних 6\*/

// \_pnil

};

#include "stdafx.h"

#include "..\spLb3\token.h"

#include "automat.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_SCNST 1024 // граничний обсяг символьних образів

struct fileInp

{char \* src\_name;

FILE \* stream;

} include\_files[20];

int x=0, y=0, f=0, cl;

//extern

unsigned nImBg=0, nImCr=0; // Початковий та поточний номери образів в буфері

//extern

char imgBuf[MAX\_SCNST]; // буфер вхідних образів

extern struct lxNode nodes[MAX\_SCNST];

FILE \*fi\_progr;

void opFls(char\* fn)

{ //search in startup dir

fi\_progr=fopen(fn,"r+b");

if (!fi\_progr)

{printf("\nImpossible to open file: %s\n",fn);exit(2);}

include\_files[f].stream=fi\_progr;

include\_files[f].src\_name=fn;

f++;

}

FILE \*fi\_progrC;

void opFlsC(char\* fn)

{/\* //search in startup dir

char \*ps, s[30]="";

ps=strcat(s,"..\\spLb9c\\");//"");

ps=strcat(ps,fn);\*/

fi\_progrC=fopen(fn,"wb");

if (!fi\_progrC)

{printf("\nImpossible to open file: %s\n",fn);exit(2);}

}

void WriteCod(char\*cd)

{fprintf(fi\_progrC,cd);

}

char ReadLtr(void)

{static char c=0;

cl=fgetc(fi\_progr);

if (c==13||c==10)

{x=0; y++;

if(cl==10)cl=fgetc(fi\_progr);

}

x++;

c=char(cl);

if (cl==EOF)c=0; //c=13;

imgBuf[nImCr++]=c;//cl;

return c;

}

void lxInit(int lxNmb,enum ltrType cl)

{nodes[lxNmb].x=x;

nodes[lxNmb].y=y;

nodes[lxNmb].f=f;

nodes[lxNmb].ndOp=\_nil;

nodes[lxNmb].prnNd=NULL;

nodes[lxNmb].prvNd=(struct lxNode \*)(imgBuf+nImCr);

if(cl<nc/\*&&nImCr!=0\*/)nodes[lxNmb].prvNd=(struct lxNode \*)(imgBuf+nImCr-1);

nodes[lxNmb].pstNd=NULL;

nodes[lxNmb].dataType=0;

nodes[lxNmb].resLength=0;

nodes[lxNmb].stkLength=0;

}

void frmCns(enum autStat s, int lxNmb)

{;

}

void eNeut(int lxNmb)

{;

}

void dGroup(int lxNmb)

{;

}

#include "stdafx.h"

#include "automat.h"

#include "langio.h"

#include "..\spLb3\token.h"

#include "tables.h"

#include "index.h"

#include <stdlib.h>

extern enum ltrType \*ltCls; // уточнюються для версій та режимів

extern enum autStat nxtStsC[Eo+1][ltrcode+1];

extern enum autStat nxtStsP[Eo+1][ltrcode+1];

extern char \*oprtrC[], \*oprtrP[], \*oprtrV[],

\*cprC[], \*cprP[], \*cprV[];

char \*\*oprtr, \*\*cpr;

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

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

modeL;

extern int x, y, f;

extern unsigned nImBg, nImCr; // Початковий та поточний номери образів в буфері

extern char imgBuf[]; // буфер вхідних образів

extern enum tokType \*dlCds;

extern enum ltrType \*ltCls;

unsigned nNode=0;

extern struct lxNode nodes[100];

extern unsigned nNdxNds;

extern struct indStrUS \*pRtNdx, ndxNds[50];

struct recrdKWD \*tablKWD;

extern struct recrdKWD tablKWDC[67];

extern struct recrdKWD tablKWDP[67];

enum autStat //\*nxtStsR=&nxtStsC[0][0],

nxtSts[Eo+1][ltrcode+1];//=&nxtStsR;//(enum autStat\*\*)nxtStsC;

void LxAnInit(char nl)

{char i=0;

switch (nl)

{case 'P':modeL=modeP;

oprtr=oprtrP, cpr=cprP;

tablKWD = tablKWDP;

for(i=0;i<=Eo;i++)for(char j=0;j<=ltrcode;j++)

nxtSts[i][j]=nxtStsP[i][j];

break;

case 'V':

default:

case 'C':modeL=modeC;

oprtr=oprtrC, cpr=cprC;

tablKWD = tablKWDC;

for(i=0;i<=Eo;i++)for(char j=0;j<=ltrcode;j++)

nxtSts[i][j]=nxtStsC[i][j];

}

}

// функція лексичного аналізу чергової лексеми

int cntMdB=0;

int LxAnlzr(void)

{//static int lxNmb=0;

static enum autStat s=S0, sP; // поточний та попередній стан лексеми

enum autStat SP;

static enum ltrType c; // клас чергової літери

static char l=1; // чергова літера (початок фыктивний)

struct recrdKWD\* pRt;

int s1, c1;

char l1, l0; // чергова літера

lxInit(nNode,c);

do {SP=

sP=s; l1=l; // запам'ятовування стану

l0=l=ReadLtr(); // читання літери

c1=c=ltCls[l]; // визначення класу літери

if(s==Scl&&c!=dlmeorml)continue;

s=nxtSts[s][c];//[c<dlmaux?c:dlmaux];// стан лексеми

s1=s;

if(s==Scr)continue;

if((sP==S2||sP==S3)&&(c>nc&&c<dlmeormr))

{// пошук в таблиці групових роздільників

imgBuf[nImCr]=0;

pRt=selBin(imgBuf+nImBg, tablKWD, 67);

if(pRt!=NULL)

{nodes[nNode].ndOp=pRt->func;

if(pRt->func==\_remL)

{s=Scr; nodes[nNode].pstNd=(struct lxNode \*)(imgBuf+nImCr);

continue;}

if(pRt->func==\_rem)

{s=Scl;} continue;

break;}else

{if(sP!=S3)

{nodes[nNode].ndOp=/\*(enum tokType)\*/dlCds[l1];

if(nodes[nNode-1].ndOp>\_cnst&&nodes[nNode-1].ndOp!=\_bckt

&&nodes[nNode-1].ndOp!=\_scbr)// перевірка унарності

{if(nodes[nNode].ndOp==\_add)nodes[nNode].ndOp=\_addU;

if(nodes[nNode].ndOp==\_sub)nodes[nNode].ndOp=\_subU;

if(nodes[nNode].ndOp==\_mul)nodes[nNode].ndOp=\_refU;

if(nodes[nNode].ndOp==\_andB)nodes[nNode].ndOp=\_ptrU;

}

imgBuf[nImBg]=imgBuf[nImBg+1];

nodes[nNode].prvNd=NULL;

}else{imgBuf[nImBg]=imgBuf[nImBg+1]; nImCr--;}

nImCr--;

sP=S0;

s=nxtSts[sP][c];//[c<dlmaux?c:dlmaux];// стан лексеми

return nNode++;

}

}

s1=s;}while(s!=S0&&s!=S2&&!((sP==S0||sP==S2||sP==S3)&&s<S2)); // перевірка кінця лексеми

s1=sP;

switch(sP)

{case Scr: case Scl:

imgBuf[nImCr++]=0;

// ((char\*)(nodes[nNode].prvNd))--;

nImBg=nImCr;

break;

case S2:

case S0:

/\* if(s==S0)

// dGroup(nNode);// аналіз групових роздільників

{//imgBuf[nImBg]=l1;

imgBuf[++nImCr]=0;//nImCr++;

}\*/

if(nodes[nNode].ndOp!=\_nil)

{nImCr=nImBg; l=' '; return nNode++;}

// if(sP!=S0)

nodes[nNode].ndOp=/\*(enum tokType)\*/dlCds[l1];//dlCds[l1];

if(nodes[nNode-1].ndOp>\_cnst&&nodes[nNode-1].ndOp!=\_bckt

&&nodes[nNode-1].ndOp!=\_scbr)// перевірка унарності

{if(nodes[nNode].ndOp==\_add)nodes[nNode].ndOp=\_addU;

if(nodes[nNode].ndOp==\_sub)nodes[nNode].ndOp=\_subU;

if(nodes[nNode].ndOp==\_mul)nodes[nNode].ndOp=\_refU;

if(nodes[nNode].ndOp==\_andB)nodes[nNode].ndOp=\_ptrU;

}

if(nodes[nNode].ndOp==\_opbr&&(nodes[nNode-1].ndOp==\_ass||cntMdB))

{cntMdB++;nodes[nNode].ndOp=\_tdbr;}

if(nodes[nNode].ndOp==\_ocbr&&cntMdB)

{nodes[nNode].ndOp=\_tcbr;cntMdB--;}

if(nodes[nNode].ndOp!=\_nil)

// &&imgBuf[nImBg]==)

{nodes[nNode].prvNd=NULL;

if(nImBg+1!=nImCr)

// if(dlCdsC[l0]!=\_nil||ltClsC[l0]==dlmaux||ltClsC[l0]==dlmeormr)

{imgBuf[nImBg]=imgBuf[nImBg+1];

if(s!=S0){nImCr--;

imgBuf[nImBg]=imgBuf[nImCr];nImCr=nImBg+1;}// 04.07.07

else nImCr=nImBg;}

return nNode++;}

else if(ltCls[imgBuf[nImBg]]==dlmaux/\*&&ltClsC[imgBuf[nImBg]]>nc\*/)

{imgBuf[nImBg]=imgBuf[nImBg+1];

nImCr--;}

return nNode;

case S1n:// пошук ключових слів та імен

imgBuf[nImCr-1]=0;

// пошук у таблиці ключів;

if(\*(imgBuf+nImBg)==13)nImBg++;

pRt=selBin(imgBuf+nImBg, tablKWD, 67);

if(pRt){nodes[nNode].ndOp=pRt->func;

// якщо знайдено

nodes[nNode].prvNd=NULL;

nImCr=nImBg;

if(c!=dlmeormr&&c!=dlmaux)imgBuf[nImCr++]=l;

return nNode++;}

// якщо не знайдено

nodes[nNode].ndOp=\_nam;

insBTr(nodes+nNode,pRtNdx);

nImBg=nImCr;

if(c!=dlmeormr&&c!=dlmaux)imgBuf[nImCr++]=l;

break;

default: // не дійшли до класифікованих помилок

case Eu: case Ec: case Ep: case Eq: case En: case Eo:// обробка помилок

eNeut(nNode); // фіксація помилки

case S1c: case S2c: case S1p: case S2s: // формування констант

imgBuf[nImCr-1]=0;

nodes[nNode].resLength=sP;// frmCns(sP, nNode); break;

nodes[nNode].ndOp=\_srcn; nodes[nNode].dataType=sP;

insBTr(nodes+nNode,pRtNdx);

nImBg=nImCr;

if(c!=dlmeormr&&c!=dlmaux)imgBuf[nImCr++]=l;

break;

case S3: nImCr=nImBg;

nodes[nNode].prvNd=NULL;

imgBuf[nImBg]=imgBuf[nImBg+2];

s1=s;

if(s!=S0)nImCr=nImBg+1;//else nImCr--;

}

return nNode++;

}

**Результат виконання**

Microsoft Windows [Version 6.1.7601]

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

C:\Users\User>"E:\Education\5\SP\spLb1 - Copy\spLb1\spLb8\Debug\spLb8.exe"

Please enter file Name: E:\v11

double b,a[ 4]={2,3,4,5};

unsigned n=1,d=5;

b=d!=0;

b=2\*a[ n];

double b,a[4]={2,3,4,5};

unsigned n=1,d=5;

b=d!=0;

b=2\*(a[n]);

double b,a[4]={2,3,4,5};

unsigned n=1,d=5;

b=d!=0;

b=2\*(a[n]);

a[4]={2,3,4,5} => 0 ->a[4]

n=1 => 1 -> n

d=5 => 5 -> d

b=d!=0 => 0 -> b

b=2\*(a[n]) => -1.08e-019 ->b

C:\Users\User>

**Висновок.**

В ході виконання лабораторної роботи були одержані навички використання вставок на мові Асемблера для побудови та оптимізації абстрактної машини інтерпретації комп’ютерної мови. Вивчені угоди про зв’язки для створення процедур і функцій інтерпретації операцій і операторів комп’ютерних мов і звертання до них за допомогою операторів мови С з використанням функціонального типа даних.