**ASSIGNMENT :2**

**Problem statement-**

Write a program to implement II pass assembler. (For hypothetical instruction set from Dhamdhere).

Consider Literal processing, forward references not expected

a.Use of literals and not symbols

b. LTORG,END

c. Error handling

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

struct MOTtable

{

char Mnemonic[6];

int Class;

char Opcode[3];

};

struct symboltable

{

char Symbol[8];

int Address;

int Size;

}ST[20];

struct literaltable

{

int Literal;

int Address;

}LT[10];

struct intermediatecode

{

int LC;

int Code1,Type1;

int Code2,Type2;

int Code3,Type3;

}IC[30];

int LC=0; **//Location counter**

int iLT=0; **//Index of next entry in Literal Table**

int iST=0; **//Index of next entry in Symbol Table**

int iIC=0; **//Index of next entry in intemediate code table**

int main()

{

FILE \*ptr1;

char file1[40],nextline[80];

char code1[5],code2[5],code3[5],code4[5],code5[5],code6[5];

int i,count;

printf("\nEnter Symbol Table Entries : ");

printf("\nEnter No. entries : ");

scanf("%d",&count);

iST=count;

for(i=0;i<count;i++)

{

printf("\nEnter the address of the symbol : ");

scanf("%d",&(ST[i].Address));

}

printf("\nEnter Literal Table Entries : ");

printf("\nEnter No. entries : ");

scanf("%d",&count);

iLT=count;

for(i=0;i<count;i++)

{

printf("\nEnter the address of the literal : ");

scanf("%d",&(LT[i].Address));

}

printf("\nenter Source file name(containing Intermediate Code):");

scanf("%s",file1);

ptr1=fopen(file1,"r");

while(!feof(ptr1))

{

**//Read a line of intermediate code and remove spaecial characters**

i=0;

nextline[i]=fgetc(ptr1);

while(nextline[i]!='\n'&& nextline[i]!=EOF )

{

if(!isalnum(nextline[i]))

nextline[i]=' ';

else

nextline[i]=toupper(nextline[i]);

i++;

nextline[i]=fgetc(ptr1);

}

nextline[i]='\0';

count=sscanf(nextline,"%s%s%s%s%s%s",code1,code2,code3

,code4,code5,code6);

if(strcmp(code1,"AD")==0)

{

if(strcmp(code1,"AD")==0 && strcmp(code2,"01")==0)

{

LC=atoi(code4);

continue;

}

else

{

printf("\n%3d) ",LC);

LC=LC+1;

continue;

}

}

if(strcmp(code1,"IS")==0 )

{

printf("\n%3d) %s ",LC,code2);

if(count==2)

{

printf("00 000");

LC=LC+1;

continue;

}

if(count==4)

{

strcpy(code5,code3);

strcpy(code6,code4);

printf("00 ");

}

else

printf("%s ",code4);

if(strcmp(code5,"S")==0)

printf("%d",ST[atoi(code6)].Address);

else

printf("%d",LT[atoi(code6)].Address);

LC=LC+1;

continue;

}

if(strcmp(code1,"DL")==0 )

{

printf("\n%3d) ",LC);

if(strcmp(code2,"01")==0)

LC=LC+atoi(code4);

else

{

printf("00 00 %3s",code4);

LC=LC+1;

}

}

}

return 0;

}

/\*

**OUTPUT:-**

**Intermediate Code-**

(AD,01) (C,200)

(IS,04) (RG,01) (L, 0)

(IS,05) (RG,01) (S, 0)

(IS,04) (RG,01) (S, 0)

(IS,04) (RG,03) (S, 2)

(IS,01) (RG,03) (L, 1)

(IS,07) (CC,07) (S, 3)

(DL,02) (L, 5)

(DL,02) (L, 1)

(IS,02) (RG,01) (L, 2)

(IS,07) (CC,02) (S, 4)

(IS,00)

(AD,03)

(IS,03) (RG,03) (S, 2)

(AD,03)

(DL,01) (C, 1)

(AD,04) (S, 1)

(DL,01) (C, 1)

(DL,02) (L, 1)

lab7@ubuntu:~$ gcc ps2a.c

lab7@ubuntu:~$ ./a.out

Enter Symbol Table Entries :

Enter No. entries : 6

Enter the address of the symbol : 211

Enter the address of the symbol : 202

Enter the address of the symbol : 213

Enter the address of the symbol : 208

Enter the address of the symbol : 202

Enter the address of the symbol : 210

Enter Literal Table Entries :

Enter No. entries : 3

Enter the address of the literal : 206

Enter the address of the literal : 207

Enter the address of the literal : 214

enter Source file name(containing Intermediate Code):inter.txt

200) 04 01 206

201) 05 01 211

202) 04 01 211

203) 04 03 213

204) 01 03 207

205) 07 07 208

206) 00 00 5

207) 00 00 1

208) 02 01 214

209) 07 02 202

210) 00 00 000

211)

212) 03 03 213

213)

214)

215)

216)

217) 00 00 1lab7@ubuntu:~$

\*/