Name: Aishwarya Shukla

Roll no.: 324002

Gr no. 22010492

Batch: D1

**LPCC Assignment no. 01**

Problem Statement: Generate Symbol table, Literal table, Pool table &amp;

Intermediate code of a two-pass Assembler for the given Assembly language source code.

Objective:

1. Student will be able to generate Symbol Table, Literal Table, Pool table, IC of a two-pass assembler

Theory:

1. Symbol Table is an important data structure created and maintained by the compiler in order to keep track of semantics of variables i.e. it stores information about the scope and binding information about names, information about instances of various entities such as variable and function names, classes, objects, etc.
2. Literal table is used for keeping track of literals that are encountered in the programs. We directly specify the value, literal is used to give a location for the value.
3. In computer science, and specifically in compiler and assembler design, a literal pool is a lookup table used to hold literals during assembly and execution.
4. Intermediate code lies between the high-level language and the machine language. If the compiler directly translates source code into the machine code without generating intermediate code then a full native compiler is required for each new machine.

Week – 1

Program:

file=open('w1\_p5.txt','r')

lines=file.readlines()

count1=1

# lc=100

lcList=[]

allSubstrings=[]

emot=[['STOP','ADD','SUB','MULT','MOVER','MOVEM','COMP','BC','DIV','READ','PRINT','START','END','ORIGIN','EQU','LTORG','DS','DC','AREG','BREG','CREG','EQ','LT','GT','NE','LE','GT','ANY'],[1,1,1,1,1,1,1,1,1,1,1,3,3,3,3,3,2,2,4,4,4,5,5,5,5,5,5,5],['00','01','02','03','04','05','06','07','08','09','10','01','02','03','04','05','01','02','01',]]

symbolTable=[]

sym\_lc=[]

symbls=[]

literal\_table=[]

lc=0

for line in lines:

    if 'START' in line:

        line=line.replace(',',' ')

        line=line.replace('\t',' ')

        line=line.strip()

        lc=int(line.split(' ')[1])-1

for line in lines:

    line=line.replace(',',' ')

    line=line.replace('\t',' ')

    line=line.strip()

    if '=' in line:

        literal=line.split('=')

        literal\_table.append(literal[1])

    substrings=line.split(' ')

    allSubstrings.append(substrings)

    lcList.append(lc)

    lc+=1

    if substrings[0] not in emot[0]:

        sym\_lc.append(lc)

        symbls.append(substrings[0])

symbolTable.append(symbls)

symbolTable.append(sym\_lc)

print("symbol table=",symbolTable)

print("Literal table=",literal\_table)

Output:



1. Week – 2

Program:

file=open('w1\_p5.txt','r')

lines=file.readlines()

count1=1

# lc=100

lcList=[]

allSubstrings=[]

emot=[['STOP','ADD','SUB','MULT','MOVER','MOVEM','COMP','BC','DIV','READ','PRINT','START','END','ORIGIN','EQU','LTORG','DS','DC','AREG','BREG','CREG','EQ','LT','GT','NE','LE','GT','ANY'],[1,1,1,1,1,1,1,1,1,1,1,3,3,3,3,3,2,2,4,4,4,5,5,5,5,5,5,5],['00','01','02','03','04','05','06','07','08','09','10','01','02','03','04','05','01','02','01',]]

symbolTable=[]

sym\_lc=[]

symbls=[]

literal\_table=[]

literal\_add=[]

literals=[]

lc=0

for line in lines:

    if 'START' in line:

        line=line.replace(',',' ')

        line=line.replace('\t',' ')

        line=line.strip()

        lc=int(line.split(' ')[1])-1

print(lc)

for line in lines:

    line=line.replace(',',' ')

    line=line.replace('\t',' ')

    line=line.strip()

    if '=' in line:

        literal=line.split('=')

        literals.append(literal[1][1])

    substrings=line.split(' ')

    allSubstrings.append(substrings)

    lcList.append(lc)

    if substrings[0] not in emot[0]:

        sym\_lc.append(lc)

        symbls.append(substrings[0])

    lc+=1

symbolTable.append(symbls)

symbolTable.append(sym\_lc)

length=[]

for i in range(len(symbolTable[0])):

            length.append(1)

symbolTable.append(length)

literal\_table.append(literals)

lc-=1

for i in range(len(literal\_table[0])):

        literal\_add.append(lc)

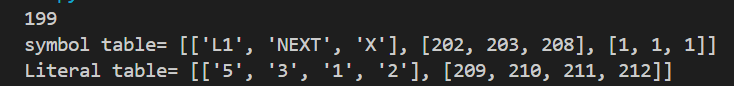
        lc+=1

literal\_table.append(literal\_add)

print("symbol table=",symbolTable)

print("Literal table=",literal\_table)

Output:



1. Week – 2 Part – 3

Program:

file=open('w2\_p3.txt','r')

lines=file.readlines()

count1=1

# lc=100

lcList=[]

allSubstrings=[]

emot=[['STOP','ADD','SUB','MULT','MOVER','MOVEM','COMP','BC','DIV','READ','PRINT','START','END','ORIGIN','EQU','LTORG','DS','DC','AREG','BREG','CREG','EQ','LT','GT','NE','LE','GT','ANY'],[1,1,1,1,1,1,1,1,1,1,1,3,3,3,3,3,2,2,4,4,4,5,5,5,5,5,5,5],['00','01','02','03','04','05','06','07','08','09','10','01','02','03','04','05','01','02','01',]]

symbolTable=[]

sym\_lc=[]

symbls=[]

literal\_table=[]

literal\_add=[]

literals=[]

pool\_table=[]

lc=0

pool\_no=0

for line in lines:

    if 'START' in line:

        line=line.replace(',',' ')

        line=line.replace('\t',' ')

        line=line.strip()

        lc=int(line.split(' ')[1])-1

print(lc)

lit\_count=0

for line in lines:

    line=line.replace(',',' ')

    line=line.replace('\t',' ')

    line=line.strip()

    if '=' in line:

        literal=line.split('=')

        literals.append(literal[1][1])

        lit\_count+=1

    substrings=line.split(' ')

    allSubstrings.append(substrings)

    if 'LTORG' in line:

        for i in range(lit\_count):

                literal\_add.append(lc)

                lc+=1

        pool\_table.append(pool\_no)

        pool\_no+=lit\_count

        lc-=1

        lit\_count=0

    lcList.append(lc)

    if substrings[0] not in emot[0]:

        sym\_lc.append(lc)

        symbls.append(substrings[0])

    lc+=1

symbolTable.append(symbls)

symbolTable.append(sym\_lc)

length=[]

for i in range(len(symbolTable[0])):

            length.append(1)

symbolTable.append(length)

literal\_table.append(literals)

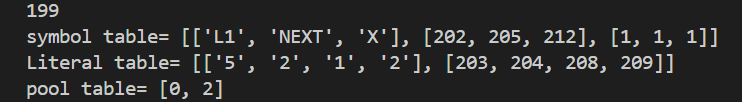
literal\_table.append(literal\_add)

print("symbol table=",symbolTable)

print("Literal table=",literal\_table)

print("pool table=",pool\_table)

Output:



1. Week – 2 Part – 4

Program:

file = open('w2\_p4.txt', 'r')

lines = file.readlines()

count1 = 1

# lc=100

lcList = []

allSubstrings = []

emot = [['STOP', 'ADD', 'SUB', 'MULT', 'MOVER', 'MOVEM', 'COMP', 'BC', 'DIV', 'READ', 'PRINT', 'START', 'END', 'ORIGIN', 'EQU', 'LTORG', 'DS', 'DC', 'AREG', 'BREG', 'CREG', 'EQ', 'LT', 'GT', 'NE', 'LE', 'GT', 'ANY'], [

    1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 2, 2, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5], ['00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '01', '02', '03', '04', '05', '01', '02', '01', ]]

symbolTable = []

sym\_lc = []

symbls = []

literal\_table = []

literal\_add = []

literals = []

pool\_table = []

lc = 0

pool\_no = 0

for line in lines:

    if 'START' in line:

        line = line.replace(',', ' ')

        line = line.replace('\t', ' ')

        line = line.strip()

        lc = int(line.split(' ')[1])-1

lit\_count = 0

for line in lines:

    line = line.replace(',', ' ')

    line = line.replace('\t', ' ')

    line = line.strip()

    if '=' in line:

        literal = line.split('=')

        literals.append(literal[1][1])

        lit\_count += 1

    substrings = line.split(' ')

    allSubstrings.append(substrings)

    if 'ORIGIN' in line:

        line = line.replace('+', ' ')

        ori = line.split(' ')

        idx = symbls.index(ori[1])

        lc = sym\_lc[idx]+int(ori[2])-1

        print("origin wala lc=", lc)

    if 'LTORG' in line:

        for i in range(lit\_count):

            literal\_add.append(lc)

            lc += 1

        pool\_table.append(pool\_no)

        pool\_no += lit\_count

        lc -= 1

        lit\_count = 0

    lcList.append(lc)

    if substrings[0] not in emot[0]:

        sym\_lc.append(lc)

        symbls.append(substrings[0])

    lc += 1

symbolTable.append(symbls)

symbolTable.append(sym\_lc)

length = []

for i in range(len(symbolTable[0])):

    length.append(1)

symbolTable.append(length)

literal\_table.append(literals)

literal\_table.append(literal\_add)

for line in lines:

    if 'EQU' in line:

        line = line.strip()

        find\_lit = line.split(' ')

        idx = symbolTable[0].index(find\_lit[2])

        lc = symbolTable[1][idx]

        idx1 = symbolTable[0].index(find\_lit[0])

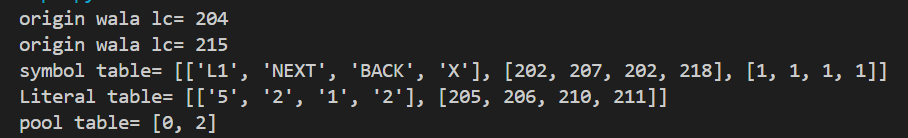
        symbolTable[1][idx1] = lc

print("symbol table=", symbolTable)

print("Literal table=", literal\_table)

print("pool table=", pool\_table)

Output:



1. Week – 1 Part – 3

Program:

file = open('w2\_p4.txt', 'r')

lines = file.readlines()

count1 = 1

# lc=100

lcList = []

allSubstrings = []

emot = [['STOP', 'ADD', 'SUB', 'MULT', 'MOVER', 'MOVEM', 'COMP', 'BC', 'DIV', 'READ', 'PRINT', 'START', 'END', 'ORIGIN', 'EQU', 'LTORG', 'DS', 'DC', 'AREG', 'BREG', 'CREG', 'EQ', 'LT', 'GT', 'NE', 'ANY'], [1, 1, 1, 1, 1,

                                                                                                                                                                                                              1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 2, 2, 4, 4, 4, 5, 5, 5, 5, 5], ['00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '01', '02', '03', '04', '05', '01', '02', '01', '02', '03', '01', '02', '03', '04', '05']]

symbolTable = []

sym\_lc = []

symbls = []

literal\_table = []

literal\_add = []

literals = []

pool\_table = []

classfield = ['IS', 'DL', 'AD', 'RG', 'CC']

lc = 0

pool\_no = 0

sys = 0

lits = 0

litlc = []

litrls = 0

for line in lines:

    if 'START' in line:

        line = line.replace(',', ' ')

        line = line.replace('\t', ' ')

        line = line.strip()

        lc = int(line.split(' ')[1])-1

lit\_count = 0

for line in lines:

    line = line.replace(',', ' ')

    line = line.replace('\t', ' ')

    line = line.strip()

    if '=' in line:

        literal = line.split('=')

        literals.append("="+literal[1])

        lit\_count += 1

        litlc.append(lc)

    substrings = line.split(' ')

    allSubstrings.append(substrings)

    if 'ORIGIN' in line:

        line = line.replace('+', ' ')

        ori = line.split(' ')

        idx = symbls.index(ori[1])

        lc = sym\_lc[idx]+int(ori[2])-1

        print("origin wala lc=", lc)

    if 'LTORG' in line:

        for i in range(lit\_count):

            literal\_add.append(lc)

            lc += 1

        pool\_table.append(pool\_no)

        pool\_no += lit\_count

        lc -= 1

        litlc = []

    lcList.append(lc)

    if substrings[0] not in emot[0]:

        sym\_lc.append(lc)

        symbls.append(substrings[0])

    tokens = line.split(' ')

    string = ""

    for i in tokens:

        if i not in emot[0] and tokens[0] == i:

            pass

        else:

            string += '('

            if i == 'LTORG':

                for i in range(lit\_count):

                    string += "DL,02)"

                    if lit\_count > 1:

                        string += "( C,"+str(literals[litrls])+")"

                    else:

                        string += "( C,"+str(literals[litrls])

                    litrls += 1

                lit\_count = 0

            elif i in emot[0]:

                idx = emot[0].index(i)

                string += classfield[emot[1][idx]-1]+" "

                string += emot[2][idx]

            elif i in literals:

                string += " L,"

                string += str(lits)

                lits += 1

            elif i in symbls and i == tokens[-1] or i.isalpha():

                string += " S,"

                string += str(sys)

                sys += 1

            else:

                string += " C,"

                string += str(lc+1)

            string += ' )'

    print(string)

    lc += 1

symbolTable.append(symbls)

symbolTable.append(sym\_lc)

length = []

for i in range(len(symbolTable[0])):

    length.append(1)

symbolTable.append(length)

literal\_table.append(literals)

if lit\_count > 0:

    for i in range(lit\_count):

        literal\_add.append(lc-1)

        lc += 1

literal\_table.append(literal\_add)

for line in lines:

    if 'EQU' in line:

        line = line.strip()

        find\_lit = line.split(' ')

        idx = symbolTable[0].index(find\_lit[2])

        lc = symbolTable[1][idx]

        idx1 = symbolTable[0].index(find\_lit[0])

        symbolTable[1][idx1] = lc

print("symbol table=", symbolTable)

print("Literal table=", literal\_table)

print("pool table=", pool\_table)

for line in lines:

    line = line.replace(',', ' ')

    line = line.replace('\t', ' ')

    line = line.strip()

Output:

