**Lab Assignment 8**

**Name: Shreya Pawar**

**TY CS C, roll no: 29**

**Problem statement**

Kindly upload OS phase  1 -  Slave (User) module code  which will execute LR,SR,CR,BT instructions of program

#include<iostream>

#include<vector>

#include<fstream>

#include <string>

#include<bits/stdc++.h>

using namespace std;

#define ROW 100

#define COL 4

// char memory[100][4];

vector<vector<string>> Memory(ROW, vector<string>(COL,"-"));

int c;

vector<string> R(4);

vector<string> IR(4);

// IR [1, 2]: Bytes 1, 2 of IR/Operation Code

// IR [3, 4]: Bytes 3, 4 of IR/Operand Address

int IC {0};

bool C {false};

// int count=0;

int SI=1;

vector<string> words {};

string line;

int IC1;

void READ(int);

void WRITE(int);

void LOAD();

void TERMINATE();

void printM(vector<vector<char>>&);

void startExecution();

void userMode();

void userMode(){

    string s = IR[2];

    string s1 = IR[3];

    s=s+s1;

    int n = stoi(s);

    if(IR[0]=="L")

    {

            for(int j=0;j<4;j++){

                R[j]=Memory[n][j];

            }

    }

    else if (IR[0]=="S")

    {

            for(int j=0;j<4;j++){

                Memory[n][j]=R[j];

            }

    }

    else if(IR[0]=="C")

    {

            for(int j=0;j<4;j++){

                if(R[j]==Memory[n][j])

                    C=true;

                else{

                    C=false;

                    break;

                }

            }

    }

    else if(IR[0]=="B"){

            if(C==true){

                IC1=n;

            }

    }

return;

}

void TERMINATE(){

    return;

}

void READ(int IC1){

            string s = IR[2];

            string s1 = IR[3];

            s=s+s1;

            int n = stoi(s);

            // int wordLen = words[3].length();

            int wordLen = line.length();

            int c=0;

            bool flag = false;

            for(int j=n;j<n+10;j++){

                for(int k=0;k<COL;k++){

                     if(c < wordLen)

                     {

                        Memory[j][k]=line[c++];

                     }

                    else

                     {

                        flag = true;

                        break;

                     }

                }

              if(flag==true)

              {

                break;

              }

            }

    return;

}

void WRITE(int IC1){

            string s = IR[2];

            string s1 = IR[3];

            s=s+s1;

            int n = stoi(s);

            int c=0;

            bool flag = false;

            string word="";

            vector <string> buffer;

     for(int j=n;j<n+10;j++){

                for(int k=0;k<COL;k++){

                     if(Memory[j][k]!="-")

                     {

                        // Memory[j][k]=line[c++];

                        // buffer[c++]=Memory[j][k];

                         word+=Memory.at(j).at(k);

                     }

                    else

                     {

                        flag = true;

                        break;

                     }

                }

              if(flag==true)

              {

                break;

              }

            }

            buffer.push\_back(word);

    ofstream fout("output1.txt",ios::app);

    string line1;

    int i=0;

    while (fout && i<buffer.size()) {

        fout << buffer[i]<< endl;

        i++;

    }

    fout.close();

return;

}

void startExecution(){

int cnt=0;

while(IC1!=IC){

            for(int i=0;i<4;i++){

                IR[i]=Memory[IC1][i];

            }

            if(IR[0]=="G"){

                SI=1;

                cnt++;

                // return;

            }

            else if(IR[0]=="P")

                SI=2;

            else if(IR[0]=="H")

                SI=3;

            else

                SI=4;

    switch(SI){

        case 1:

                if(cnt>1){

                    return;

                }

                READ(IC1);

                break;

        case 2:

                WRITE(IC1);

                break;

        case 3:

                TERMINATE();

                break;

        case 4:

                userMode();

                break;

    }

    if(SI==4 && C==true && IR[0]=="B")

            continue;

    else

        ++IC1;

 }

 return;

}

void LOAD(){

    ifstream fin;

    fin.open("input1.txt");

    int temp=0;

    string s;

    bool flag = false;

    int count=0;

     while(fin){

         getline(fin, line);

        int c=0;

         if(line[0]=='$'){

            count++;

            continue;

         }

         else if(count==1){

         int wordLen = line.length();

            for(int j=0;j<ROW;j++){

                for(int k=0;k<COL;k++){

                    if(c < wordLen){

                         Memory[j][k]=line[c++];

                        //  c++;

                    }

                    else

                    {

                    flag = true;

                    break;

                    }

                }

                IC++;

             if(flag==true)

             {

                break;

             }

          }

         }

         else if(count>1){

            startExecution();

         }

     }

     fin.close();

}

void printM(vector<vector<string>> &M){

    int cnt = 0;

    int block = 0;

    for(int i=0; i<ROW; i++)

    {

        if(cnt==10 || cnt == 0)

        {

            cout<<"\nBlock "<< block <<endl;

            block++;

            cnt=0;

        }

        cout<<i<<" =>  ";

        for(int j=0; j<COL; j++)

        {

            cout<<M.at(i).at(j)<<"  ";

        }

        cnt++;

        cout<<endl;

    }

}

int main(){

    ofstream fout;

    fout.open("output1.txt");

    fout<<"";

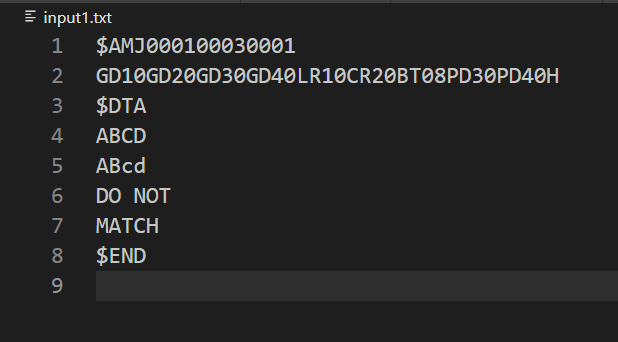
    fout.close();

    LOAD();

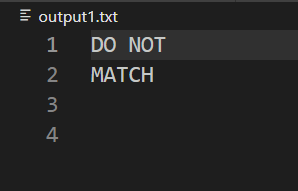
    printM(Memory);

}

**Input1.txt file:**



**Output1.txt file:**



**Output:**

Block 0

0 =>  G  D  1  0

1 =>  G  D  2  0

2 =>  G  D  3  0

3 =>  G  D  4  0

4 =>  L  R  1  0

5 =>  C  R  2  0

6 =>  B  T  0  8

7 =>  P  D  3  0

8 =>  P  D  4  0

9 =>  H  -  -  -

Block 1

10 =>  A  B  C  D

11 =>  -  -  -  -

12 =>  -  -  -  -

13 =>  -  -  -  -

14 =>  -  -  -  -

15 =>  -  -  -  -

16 =>  -  -  -  -

17 =>  -  -  -  -

18 =>  -  -  -  -

19 =>  -  -  -  -

Block 2

20 =>  A  B  c  d

21 =>  -  -  -  -

22 =>  -  -  -  -

23 =>  -  -  -  -

24 =>  -  -  -  -

25 =>  -  -  -  -

26 =>  -  -  -  -

27 =>  -  -  -  -

28 =>  -  -  -  -

29 =>  -  -  -  -

Block 3

30 =>  D  O     N

31 =>  O  T  -  -

32 =>  -  -  -  -

33 =>  -  -  -  -

34 =>  -  -  -  -

35 =>  -  -  -  -

36 =>  -  -  -  -

37 =>  -  -  -  -

38 =>  -  -  -  -

39 =>  -  -  -  -

Block 4

40 =>  M  A  T  C

41 =>  H  -  -  -

42 =>  -  -  -  -

43 =>  -  -  -  -

44 =>  -  -  -  -

45 =>  -  -  -  -

46 =>  -  -  -  -

47 =>  -  -  -  -

48 =>  -  -  -  -

49 =>  -  -  -  -

Block 5

50 =>  -  -  -  -

51 =>  -  -  -  -

52 =>  -  -  -  -

53 =>  -  -  -  -

54 =>  -  -  -  -

55 =>  -  -  -  -

56 =>  -  -  -  -

57 =>  -  -  -  -

58 =>  -  -  -  -

59 =>  -  -  -  -

Block 6

60 =>  -  -  -  -

61 =>  -  -  -  -

62 =>  -  -  -  -

63 =>  -  -  -  -

64 =>  -  -  -  -

65 =>  -  -  -  -

66 =>  -  -  -  -

67 =>  -  -  -  -

68 =>  -  -  -  -

69 =>  -  -  -  -

Block 7

70 =>  -  -  -  -

71 =>  -  -  -  -

72 =>  -  -  -  -

73 =>  -  -  -  -

74 =>  -  -  -  -

75 =>  -  -  -  -

76 =>  -  -  -  -

77 =>  -  -  -  -

78 =>  -  -  -  -

79 =>  -  -  -  -

Block 8

80 =>  -  -  -  -

81 =>  -  -  -  -

82 =>  -  -  -  -

83 =>  -  -  -  -

84 =>  -  -  -  -

85 =>  -  -  -  -

86 =>  -  -  -  -

87 =>  -  -  -  -

88 =>  -  -  -  -

89 =>  -  -  -  -

Block 9

90 =>  -  -  -  -

91 =>  -  -  -  -

92 =>  -  -  -  -

93 =>  -  -  -  -

94 =>  -  -  -  -

95 =>  -  -  -  -

96 =>  -  -  -  -

97 =>  -  -  -  -

98 =>  -  -  -  -

99 =>  -  -  -  -