**Lab Assignment 7**

Name: Shreya Pawar

TY CS C, roll no:29

**Problems statement**

Kindly upload OS phase 1 - MOS module code which will execute GD,PD and H 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 SI=1;

vector<string> words {};

string line;

int IC1;

void READ(int);

void LOAD();

void TERMINATE();

void WRITE(int);

void startExecution();

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

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("output.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;

}

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

continue;

else

++IC1;

}

return;

}

void LOAD(){

ifstream fin;

fin.open("input.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("output.txt");

fout<<"";

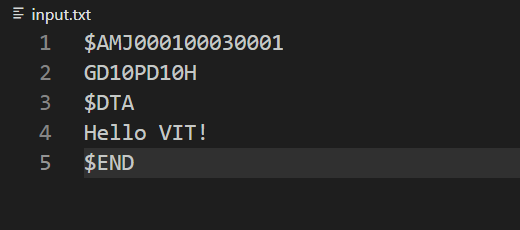
fout.close();

LOAD();

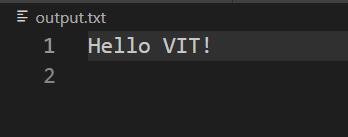
printM(Memory);

}

**Input.txt file:**

****

**Output.txt file:**

****

**Output:**

