**VISHWAKARMA INSTITUTE OF TECHNOLOGY**

COMPUTER ENGINEERING

**Name: Hetan Hemant Nandre**

**Division: C**

**Roll Number: 33**

Subject: Operating System (OS) – Course Project

MULTIPROGRAMMING OPERATING SYSTEM (MOS) PROJECT

Second Phase

ASSUMPTIONS:

* Jobs may have program errors
* PI interrupt for program errors introduced
* No physical separation between jobs
* Job outputs separated in output file by 2 blank lines
* Paging introduced, page table stored in real memory
* Program pages allocated one of 30 memory block using random number generator
* Load and run one program at a time
* Time limit, line limit, out-of-data errors introduced
* TI interrupt for time-out error introduced
* 2-line messages printed at termination

NOTATION

M: memory

IR: Instruction Register (4 bytes)

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

IR [3, 4]: Bytes 3, 4 of IR/Operand Address M[&]: Content of memory location &

IC: Instruction Counter Register (2 bytes) R: General Purpose Register (4 bytes)

C: Toggle (1 byte)

PTR: Page Table Register (4 bytes)

PCB: Process Control Block (data structure) VA: Virtual Address

RA: Real Address

TTC: Total Time Counter

LLC: Line Limit Counter

TTL: Total Time Limit

TLL: Total Line Limit

EM: Error Message

¬ : Loaded/stored/placed into

**Program Code -**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*    AUTHOR:         HETAN HEMANT NANDRE  \*

\*    INSTITUITION:   VIT PUNE             \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#include<stdlib.h>

#include<iostream>

#include<fstream>

#include<string.h>

using namespace std;

int IC,PTR,VA,RA,kio=-1,flag[30],PI,TI,SI,TTC=0,LLC=0;

char M[300][4],IR[4],R[4],C;

char page[10][4];

fstream fin,fout;

char temp[100];

string line;

void MOS();

struct PCB

{

    char job[4],TTL[4],TLL[4];

}pcb;

int ttl,tll;

void endprogram()

{

    fout<<"\nSI = "<<SI<<"  TI ="<<TI<<"  PI="<<PI<<endl;

        fout<<"TTC = "<<TTC<<"  LLC="<<LLC<<endl;

            cout<<"\nSI = "<<SI<<"  TI ="<<TI<<"  PI="<<PI<<endl;

        cout<<"TTC = "<<TTC<<"  LLC="<<LLC<<endl;

    exit(1);

}

void allocate()

{

    int pos,i,j,k=0,check=0,len;

    char str[2];

    int level=0;

    while(check!=1)

    {

        cout<<"TEMP  "<<kio<<endl;

        kio++;

        cout<<"Cnt1  "<<level<<endl;

        pos=(rand()%29)\*10;

        cout<<"Pos1   "<<pos<<endl;

        while(flag[pos/10] != 0 )

        {

        cout<<"In while pos"<<endl;

        pos=(rand()%29)\*10;

        cout<<"Pos2  "<<pos<<endl;

        cout<<"cnt2  "<<level<<endl;

        }

        flag[pos/10]=1;

        itoa(pos,str,10);

        if(pos/100==0)

        {

            M[PTR+kio][2]='0';

             M[PTR+kio][3]=str[0];

        }

        else{

                M[PTR+kio][2]=str[0];

        M[PTR+kio][3]=str[1];

        cout<<"Cnt3  "<<level<<endl;

        }

        getline(fin,line);

        cout<<line<<"Line2"<<endl;

        level++;

        k=0;

        for(i=0;i<line.size()/4;i++)

        {

            for(j=0;j<4;j++)

            {

                M[pos+i][j]=line[k];

                k++;

                if(line[k]=='H')

                {

                    check=1;

                    M[pos+(i+1)][0]='H';

                    M[pos+(i+1)][1]='0';

                    M[pos+(i+1)][2]='0';

                    M[pos+(i+1)][3]='0';

                }

            }

        }

    }

    cout<<endl<<"MEMORY"<<endl;

    for(i=0;i<300;i++)

    {cout<<"["<<i<<"] = ";

        for(j=0;j<4;j++)

        {

            cout<<M[i][j];

        }

        cout<<endl;

    }

}

void AddMap()

{

    int add,pos;

    char str[2];

    RA = PTR+(VA/10);

    if(M[RA][3]== '#')

    {

        cout<<"\*\*\*\* Page fault occured \*\*\*\*\n";

        pos=(rand()%29)\*10;

        cout<<"POS1   "<<pos<<endl;

        while(flag[pos/10] != 0 )

        {

        cout<<"in while pos"<<endl;

        pos=(rand()%29)\*10;

        cout<<"POS2  "<<pos<<endl;

        }

        flag[pos/10]=1;

        itoa(pos,str,10);

        if(pos/100==0)

        {

            M[RA][2]='0';

            M[RA][3]=str[0];

        }

        else

        {

            M[RA][2]=str[0];

            M[RA][3]=str[1];

        }

        PI=3;

    }

    if( RA > PTR+10 )

    {

        PI=2;

        MOS();

    }

}

void read()

{

    int no,i,j,k,z,x;

    getline(fin,line);

    cout<<"\nLine  :"<<line;

    no=((M[RA][2]-48)\*10)+(M[RA][3]-48);

    no=no\*10;

    k=0;

    for(i=0; k<=line.size()  ;i++)

    {

        for(j=0;j<4 && k<=line.size();j++)

        {

            cout<<"\Count  :";

            cout<<no+i<<endl<<j<<"\n";

            M[no+i][j] = line[k];

            k++;

        }

    }

    for(i=0;i<300;i++)

    {

        cout<<"["<<i<<"] = ";

        for(j=0;j<4;j++)

        {

            cout<<M[i][j];

        }

        cout<<endl;

    }

}

void write()

{

    char buff[40];

    cout<<"\n In Write : \n";

    int no=0,i,j,k,z,x;

    no=((M[RA][2]-48)\*10)+(M[RA][3]-48);

    no=no\*10;

    k=0;

    while(1)

    {

        for(i=0; i<4 ;i++)

        {

            if(M[no][i] == '\_')

                break;

            buff[k] = M[no][i];

            k++;

        }

        if(M[no][i] == '\_')

                break;

        no++;

    }

    buff[k]='\0';

    cout<<"\n Line is : "<<buff<<endl;

    fout.write(buff,strlen(buff));

    cout<<"program terminated normally";

    fout<<endl<<"-----------------------------------------------------------------------------------"<<endl;

    fout<<endl<<"program terminated normally"<<endl;

}

void MOS()

{

    if(PI==1)

    {

        cout<<"\n\*\*\*\* Opcode Error : \*\*\*\*\n\*\*\*\* Program terminated abnormally. \*\*\*\*\n\n";

            fout<<"\n\*\*\*\* Opcode error : \*\*\*\*\n\*\*\*\* Program terminated abnormally. \*\*\*\*";

            endprogram();

    }

    else if(PI==2)

    {

        cout<<"\n\*\*\*\* Operand Error : \*\*\*\*\n\*\*\*\* Program terminated abnormally. \*\*\*\*\n\n";

            fout<<"\n\*\*\*\* Operand error : \*\*\*\*\n\*\*\*\* Program terminated abnormally. \*\*\*\*";

            endprogram();

    }

    if(SI==3)

        endprogram();

    else if(SI==1)

    {

        if(TI==0)

            read();

        else if(TI==2)

        {

            cout<<"\n\*\*\*\* Time Limit Exceeded : \*\*\*\*\n\*\*\*\* Program terminated abnormally. sss\*\*\*\*";

            fout<<"\n\*\*\*\* Time Limit Exceeded : \*\*\*\*\n\*\*\*\* Program terminated abnormally. \*\*\*\*";

            endprogram();

        }

    }

    else if(SI==2)

    {

        cout<<"\nIN SI=2";

        if(TI==0)

        {

            cout<<"\nIN TI=0";

            write();

        }

        else if(TI==2)

        {

             cout<<"\nIN TI=0";

            write();

            cout<<"\n\*\*\*\* Time Limit Exceeded : \*\*\*\*\n\*\*\*\* Program terminated abnormally. \*\*\*\*";

            fout<<"\n\*\*\*\* Time Limit Exceeded : \*\*\*\*\n\*\*\*\* Program terminated abnormally. \*\*\*\*";

            endprogram();

        }

        else if(TI==1)

        {

            cout<<"\n\*\*\*\* Line Limit Exceeded : \*\*\*\*\n\*\*\*\* Program terminated abnormally. \*\*\*\*";

            fout<<"\n\*\*\*\* Line Limit Exceeded : \*\*\*\*\n\*\*\*\* Program terminated abnormally. \*\*\*\*";

            endprogram();

        }

    }

}

void examine()

{

    char ch;

    ch=IR[0];

    PI=0;

    switch(ch)

    {

    case 'G':

        cout<<"\nSI = "<<SI<<"  TI ="<<TI<<"  PI="<<PI<<endl;

        cout<<"TTC = "<<TTC<<"  LLC="<<LLC<<endl;

        if(IR[1] !='D')

        {

            PI=1;

            MOS();

        }

        else

        {

            TTC = TTC+2;

            if(TTC <= ttl)

            {

                SI=1;

                MOS();

            }

            else

            {

                TI=2;

                MOS();

            }

        }

        cout<<"\nSI = "<<SI<<"  TI ="<<TI<<"  PI="<<PI<<endl;

        cout<<"TTC = "<<TTC<<"  LLC="<<LLC<<endl;

        break;

    case 'P':

        SI=2;

        cout<<"\nSI = "<<SI<<"  TI ="<<TI<<"  PI="<<PI<<endl;

        cout<<"TTC = "<<TTC<<"  LLC="<<LLC<<endl;

        if(IR[1] != 'D')

        {

            PI=1;

            MOS();

        }

        else

        {

            LLC=LLC+1;

            TTC=TTC+1;

            cout<<LLC<<" :llc"<<endl;

            cout<<TTC<<" :ttc"<<endl;

            if(LLC < tll)

            {

                TI=0;

                MOS();

            }

            if(TTC > ttl)

            {

                TI=1;

                MOS();

            }

            else

            {

                SI=2;

                MOS();

            }

        }

        cout<<"\nTTC = "<<TTC<<"  LLC="<<LLC<<endl;

        cout<<"SI = "<<SI<<"  TI ="<<TI<<"  PI="<<PI<<endl;

        break;

    case 'H':

        SI=3;

        MOS();

        break;

    }

}

void executeProgram()

{

    int no;

    char a[3];

    int i,j,k;

    for(i=0;i<=kio;i++)

    {

            cout<<"PTR  "<<PTR<<endl;

            a[0]=M[PTR+i][2];

            a[1]=M[PTR+i][3];

            a[2]='\0';

            no=((a[0]-48)\*10)+(a[1]-48);

            cout<<endl<<"no  "<<no;

            for(j=0;j<10;j++)

            {

                for(k=0;k<4;k++)

                {

                    IR[k]=M[(no\*10)+j][k];

                }

                cout<<"IR  "<<IR<<endl;

                VA=((IR[2]-48)\*10)+(IR[3]-48);

                AddMap();

                examine();

            }

            cout<<endl<<"A  "<<a<<endl;

    }

}

void startExecution()

{

    IC=0;

    executeProgram();

}

void init()

{

    int i,j;

    PTR=(rand()%29)\*10;

    cout<<"PTR   "<<PTR<<endl;

    for(i=0;i<30;i++)

    {

        flag[i]=0;

    }

    flag[PTR/10]=1;

    for(i=0;i<300;i++)

    {

        for(j=0;j<4;j++)

        {

            M[i][j]='\_';

        }

    }

    for(i=PTR;i<PTR+10;i++)

    {

        for(j=0;j<4;j++)

        {

            M[i][j]='#';

        }

    }

}

void load()

{

    printf("IN LOAD\n");

    int i,j,k;

    while(fin)

    {

        cout<<"In while"<<endl;

        getline(fin,line);

        cout<<line<<"line1"<<endl;

        if(line[0] =='$' && line[1]=='A' && line[2]=='M' && line[3]=='J')

        {

            SI=0;

            TI=0;

            TTC=0;LLC=0;

            for(i=4,j=0;i<8;i++,j++)

            {

                pcb.job[j]=line[i];

            }

            for(i=8,j=0;i<12;i++,j++)

            {

                pcb.TTL[j]=line[i];

            }

            for(i=12,j=0;i<16;i++,j++)

            {

                pcb.TLL[j]=line[i];

            }

            ttl= (pcb.TTL[0]-48)\*1000 + (pcb.TTL[1]-48)\*100 + (pcb.TTL[2]-48)\*10 + (pcb.TTL[3]-48);

            tll= (pcb.TLL[0]-48)\*1000 + (pcb.TLL[1]-48)\*100 + (pcb.TLL[2]-48)\*10 + (pcb.TLL[3]-48);

            cout<<" TTL jdfjks = "<<ttl<<endl;

            cout<<" TLL = "<<tll<<endl;

            TTC=0;

            LLC=0;

            init();

            allocate();

        }

        if(line[0] =='$' && line[1]=='D' && line[2]=='T' && line[3]=='A')

        {

            startExecution();

        }

    }

}

int main()

{

    fin.open("input.txt");

    fout.open("output.txt");

    load();

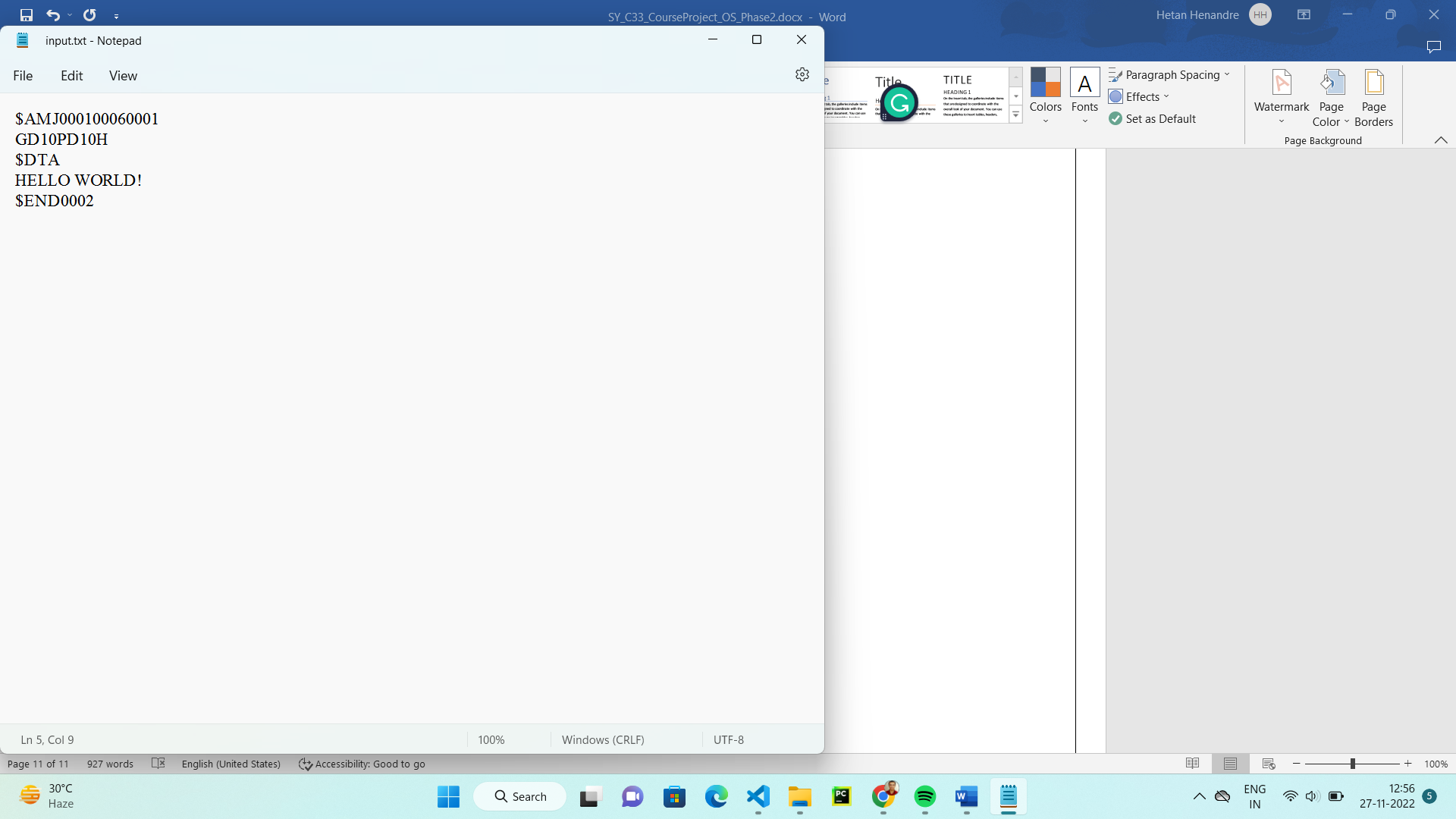
    fin.close();

    fout.close();

    return 0;

}

**Input -**



**Output –**

