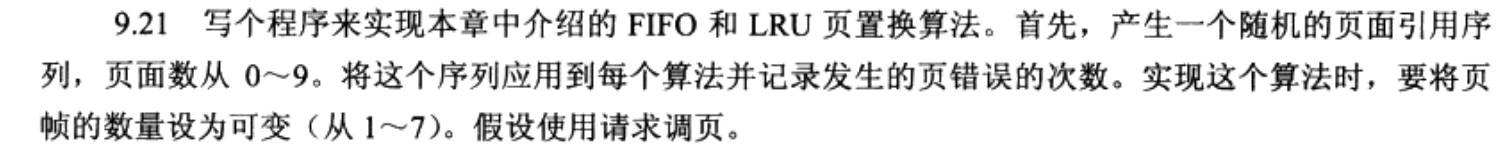
**操作系统实验四 页面置换算法**

班级：2017211314 学号：2017213508 学生：蒋雪枫

1. **任务：**



1. **分析：**

关于请求调页的页面置换算法，李老师已经在课堂上给我们讲了很多了。在完成本次实验的时候，学生直接想象上课时老师给出一个引用串与置换算法，我们做题的流程，所以本次实验还是完成地比较顺利的。由于C++ STL的queue并不支持遍历操作，这给我们的置换算法带来了些许不便，所以本次我们采用C++ STL vector来模拟Frame。

1. **变量说明：**

int Frame\_size; //页帧的数量，用户决定输入0~7作为其大小

int Reference\_string[]；//引用串

int page\_fault\_times;//缺页次数

vector<int> q(Frame\_size,-1);//页帧模拟，初始化为-1

1. **源程序：**

#include <iostream>

#include <vector>

using namespace std;

int main()

{

    int Frame\_size=5;

    cout<<"Firstly we simulate FIFO method,please input the initial Frames size:"<<endl;

    cin>>Frame\_size;

    vector<int> q(Frame\_size,-1);

    int Reference\_string[]={ 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1};

    int N=sizeof(Reference\_string)/sizeof(Reference\_string[0]);

    int page\_fault\_times=0;

    int victim=0;

    for(int i=0;i<N;i++)

    {

        int temp=Reference\_string[i];

        bool flag=false;*//means string[i] not in QUEUE*

        for(int j=0;j<Frame\_size;j++)

        {

            if(q[j]==temp)

            {

                flag=true;

                break;

            }

            if(q[j]==-1)

                break;

        }

        if(flag)

        {

            cout<<"TURN:"<<i<<" ,No page fault"<<endl;

        }

        else

        {

            q[victim]=temp;

            victim=(victim+1)%Frame\_size;

            cout<<"TURN:"<<i<<" ,page fault! ";

            cout<<"FRMAMES:"<<q[0]<<" "<<q[1]<<" "<<q[2]<<" "<<q[3]<<" "<<q[4]<<endl;

            page\_fault\_times++;

        }

    }

    cout<<"Sum of page fault is "<<page\_fault\_times<<endl;

    cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

    cout<<"Then we move to LRU method"<<endl;

*//7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1*

    vector<int> qq(Frame\_size,-1);

    victim=0;

    page\_fault\_times=0;

    N=sizeof(Reference\_string)/sizeof(Reference\_string[0]);

    for(int i=0;i<N;i++)

    {

        int temp=Reference\_string[i];

        bool flag=false;*//means string[i] not in Frames*

        for(int j=0;j<Frame\_size;j++)

        {

            if(qq[j]==temp)

            {

                flag=true;

                break;

            }

            if(qq[j]==-1)

                break;

        }

        if(flag)

        {

            cout<<"TURN:"<<i<<", "<<temp<<" ,No page fault"<<endl;

        }

        else

        {

*//stands for we need to refer to REFERENCE\_STRING to find a victim in FRAMES*

            page\_fault\_times++;

            int pointer=i;

            int counter=0;

            bool matcher[Frame\_size];

            for(int ite=0;ite<Frame\_size;ite++)

                matcher[ite]=false;

            while(pointer>=0&&counter<=Frame\_size-1)

            {

                if(pointer<=0) break;

                pointer--;

                int comp=Reference\_string[pointer];

                for(int it=0;it<Frame\_size;it++)

                {

                    if(comp==qq[it])

                    {

                        matcher[it]=true;

                        counter++;

                        break;

                    }

                }

            }

*//          cout<<"matcher:"<<matcher[0]<<matcher[1]<<matcher[2]<<matcher[3]<<matcher[4]<<endl;*

            for(int it=0;it<Frame\_size;it++)

            {

                if(matcher[it]==false)

                {

                    qq[it]=Reference\_string[i];

                    break;

                }

            }

            cout<<"TURN:"<<i<<", "<<temp<<" ,page fault! ";

            cout<<"FRMAMES:"<<qq[0]<<" "<<qq[1]<<" "<<qq[2]<<" "<<qq[3]<<" "<<qq[4]<<endl;

        }

    }

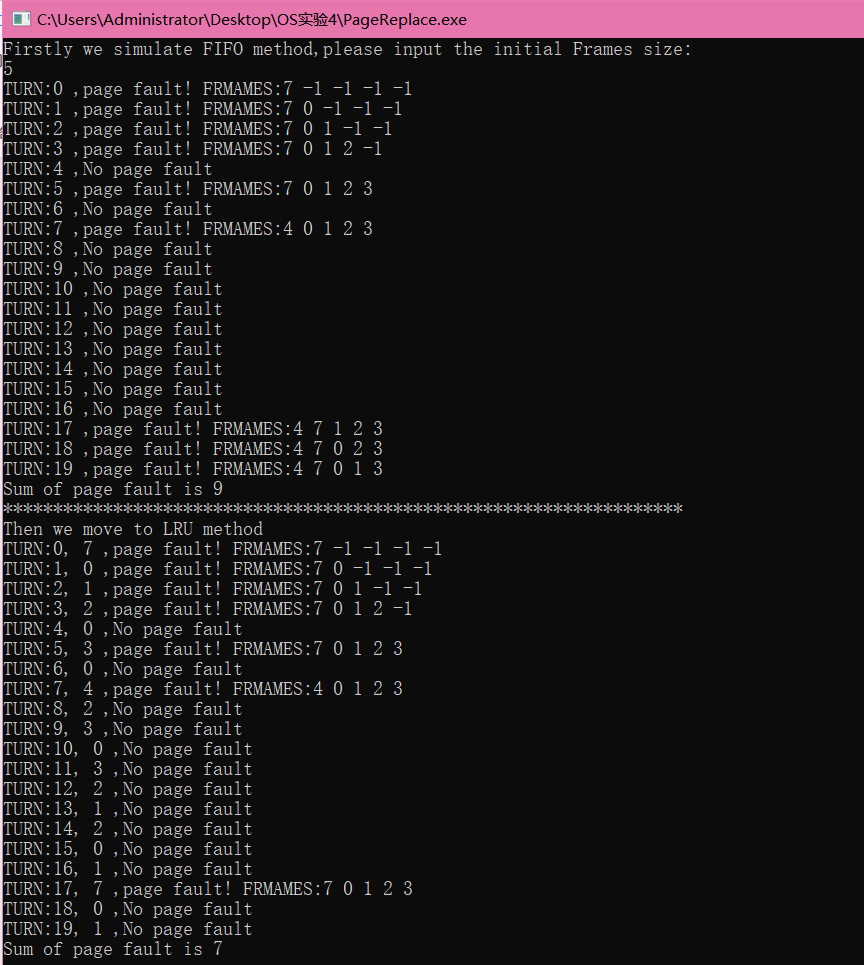
    cout<<"Sum of page fault is "<<page\_fault\_times<<endl;

}

1. **运行结果实例：**

**选择李老师上课讲LRU的例子作为20bit的引用串。**

**引用串：Reference\_string[]={ 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1};**



结果与实际一致。