模拟页面置换算法报告



班级：计科1505 学号：2015014407

张永德

北京化工大学

信息科学与技术学院计算机科学与技术专业

2017.7

**目录**

[一、功能描述 1](#_Toc486858111)

[1.1 系统实现的目的和意义 1](#_Toc486858112)

[1.2 系统实现的功能 1](#_Toc486858113)

[1.3 系统开发的环境 1](#_Toc486858114)

[二、总体结构 1](#_Toc486858115)

[2.1 文件清单 1](#_Toc486858116)

[2.2 系统功能结构图 2](#_Toc486858117)

[2.3 系统的总流程图 2](#_Toc486858118)

[三、详细设计 2](#_Toc486858119)

[3.1 数据结构 2](#_Toc486858120)

[四、系统实现 5](#_Toc486858122)

[4.1模块一（FIFO） 5](#_Toc486858123)

[4.2模块二（LRU） 5](#_Toc486858124)

[4.3模块三（OPT）…… 6](#_Toc486858125)

[4.4模块四（LFU）……………….………………………………………………………………6](#_4.4模块四（OPT）)

[4.5模块五（NRU）………………………………………………………………………………7](#_4.5模块四（OPT）)

[五、程序的过程展示 7](#_Toc486858126)

[5.1 程序运行过程展示 7](#_Toc486858127)

[5.2 程序结果展示及分析 9](#_Toc486858128)

5.3README使用界面…………………………………………………………………………..11

[六、软件的辅助说明 12](#_Toc486858129)

[6.1 本软件的优点 12](#_Toc486858130)

[6.2 本软件的缺点 12](#_Toc486858131)

[6.3 本软件的BUG 12](#_Toc486858132)

[七、课程设计结论 12](#_Toc486858133)

[八、参考文献（参考网站） 12](#_Toc486858134)

[九、附录——提交文档 12](#_Toc486858135)

[9.1 电子形式清单 12](#_Toc486858137)

9.2书面形式清单………………………………………………………………………………...12

# 一、功能描述

## 1.1 系统实现的目的和意义

本软件的目的：

第一：增强自身对于操作系统中的页面置换算法有了更深的体会，对软件的编写让自己对该类算法有了实际的操作意义

第二：可以锻炼自身的程序编写能力

本软件的意义：

第一：可以锻炼自身，给自己实际的实践经验

第二：所写软件在日后的其他地方有需要的时候可以使用

第三：该软件在QT类库上编写，可跨平台

## 1.2 系统实现的功能

1.实现了对于FIFO，OPT，LRU，LFU，NRU算法的模拟

2.实现了对目前物理块的页面分配情况的实时监控

3.实现了对于新到来页面是否缺页的判断

4.实现了缺页率的计算

## 1.3 系统开发的环境

硬件配置：MacBOOK Pro Early 2015

操作系统版本：MAC OS Sierra ; Version:10.12.3

开发环境：基于C++库的跨平台QT程序库；IDE：QT Creator

# 二、总体结构

## 2.1 文件清单（按文件系统树型方式排列）(Debug未发布版本)

Main.cpp

Page\_change.cpp

Page\_change

文件夹（主文件夹）

Page\_change.h

Page\_change.ui

Page\_change.pro

Page\_change.cpp.autosave

Page\_change pro.user

（为了方便程序演示时候的修改，暂时没有产生可发布的Release版本）

## 2.2 系统功能结构图

Page\_change.cpp

主要功能函数

结构体、类以及函数定义

Page\_change g.h

Main.cpp

主函数入口

图形界面文件

Page\_change.ui

## 2.3 系统的总流程图

Page\_change g.h

结构体，函数定义

Main.cpp主函数入口

程序进入初始界面

模式选择

3：OPT

5：NRU

4：LFU

2：LRU

1：FIFO

缺页率

# 三、详细设计

## 3.1 数据结构

1.宏定义

#define M\_count 4 //物理块的数量

#define page\_list 32 //分出页的数量

#define instruction\_number 10 //每页中指令的数量

#define wait\_time 10//程序暂停时间

#define all\_instryction 320//所有指令一共的数量

2.结构体

typedef struct ins

{

int order;//指令编号

int used; //是否使用的标记

}ins;

typedef struct PAGE

{

int pag; //show each unit containing the page number

int targe;//to show if this page is page-missing

ins instructions[instruction\_number];//each page will have 10 instructions

}pagest;

typedef struct MEM

{

int M\_number;//used to show this physic memory is now containing which page

int time; //is time

int using\_time;

int is\_get; //the flag that show if we have found it

int A;

int changed;

}mem; //物理块的结构体

3.类定义

class pro4

{

private:

pagest pagelist[page\_list];//pagelist means to the calling pages ,here ,we use to store the pages

mem M[M\_count];

int rand\_flag;

public:

int search\_oldest();//寻找一个最老的页面

void inputdata(Ui::page\_change \*k); //输入数据函数

int search\_blank\_mem();//寻找空白物理块的函数

int search\_get(int find); //寻找是否已经存在于物理块中的页

void FIFO(Ui::page\_change \*k); //先进先出算法FIFO主功能函数

void LRU(Ui::page\_change \*k); //LRU的主功能函数

void OPT(Ui::page\_change \*k); //OPT算法的主功能函数

void LFU(Ui::page\_change \*k); //LFU算法的主功能函数

void NRU(Ui::page\_change \*k); //NRU算法的主功能函数

void print(int count\_missing,Ui::page\_change \*k); //打印函数

int search\_latest();//找最新的函数

void change\_states(Ui::page\_change \*k); //改变UI状态的函数

void initial\_all\_instructrion(void); //初始化函数

int complete(void); //判断是否完成的函数

int get\_rand\_pos(int pos); //按照题目规则生成的随机数指令序列函数

void get\_order\_list(int order\_list[]);//生成指令序列函数

};

# 四、系统实现

## 4.1模块一（FIFO）

4.1.1 输入

4.1.1.1

void pro4::get\_order\_list(int order\_list[])//把产生的随机数放在序列中

4.1.1.2

void pro4::get\_rand\_pos(int pos)//按照题目原则产生随机数

4.1.2 输出

4.1.2.1

void pro4::change\_state(Ui::page\_change \*k)//输出页面状态

4.1.2.2

k->textBrowser->setText(showing);//输出控制台信息

4.1.2.3

k->misrate->setText(QString::number(misr))//最后输出缺页率

4.1.3主功能

主体函数

## 4.2模块二（LRU）

4.2.1 输入

4.1.1.1

void pro4::get\_order\_list(int order\_list[])//把产生的随机数放在序列中

4.1.1.2

void pro4::get\_rand\_pos(int pos)//按照题目原则产生随机数

4.2.2 输出

4.2.2.1

void pro4::change\_state(Ui::page\_change \*k)//输出页面状态

4.2.2.2

k->textBrowser->setText(showing);//输出控制台信息

4.2.2.3

k->misrate->setText(QString::number(misr))//最后输出缺页率

4.2.3主功能

主体函数

## 4.3模块三（OPT）

4.3.1 输入

4.1.1.1

void pro4::get\_order\_list(int order\_list[])//把产生的随机数放在序列中

4.1.1.2

void pro4::get\_rand\_pos(int pos)//按照题目原则产生随机数

4.3.2 输出

4.3.2.1

void pro4::change\_state(Ui::page\_change \*k)//输出页面状态

4.3.2.2

k->textBrowser->setText(showing);//输出控制台信息

4.3.2.3

k->misrate->setText(QString::number(misr))//最后输出缺页率

4.3.3主功能

主体函数

## 4.4模块四（LFU）

4.4.1 输入

4.4.1.1

void pro4::get\_order\_list(int order\_list[])//把产生的随机数放在序列中

4.4.1.2

void pro4::get\_rand\_pos(int pos)//按照题目原则产生随机数

4.4.2 输出

4.4.2.1

void pro4::change\_state(Ui::page\_change \*k)//输出页面状态

4.4.2.2

k->textBrowser->setText(showing);//输出控制台信息

4.4.2.3

k->misrate->setText(QString::number(misr))//最后输出缺页率

4.4.3主功能

主体函数

## 4.5模块五（NRU）

4.5.1 输入

4.5.1.1

void pro4::get\_order\_list(int order\_list[])//把产生的随机数放在序列中

4.5.1.2

void pro4::get\_rand\_pos(int pos)//按照题目原则产生随机数

4.5.2 输出

4.5.2.1

void pro4::change\_state(Ui::page\_change \*k)//输出页面状态

4.5.2.2

k->textBrowser->setText(showing);//输出控制台信息

4.5.2.3

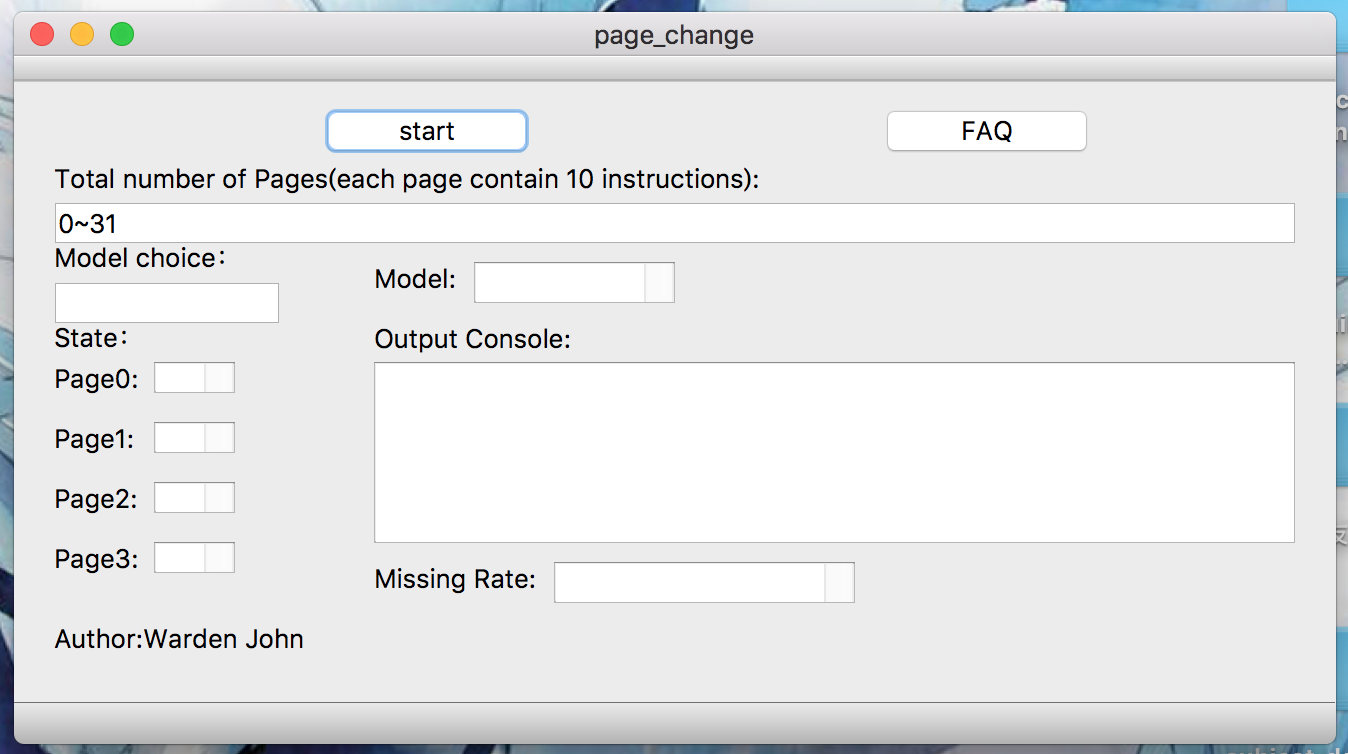
k->misrate->setText(QString::number(misr))//最后输出缺页率

4.5.3主功能

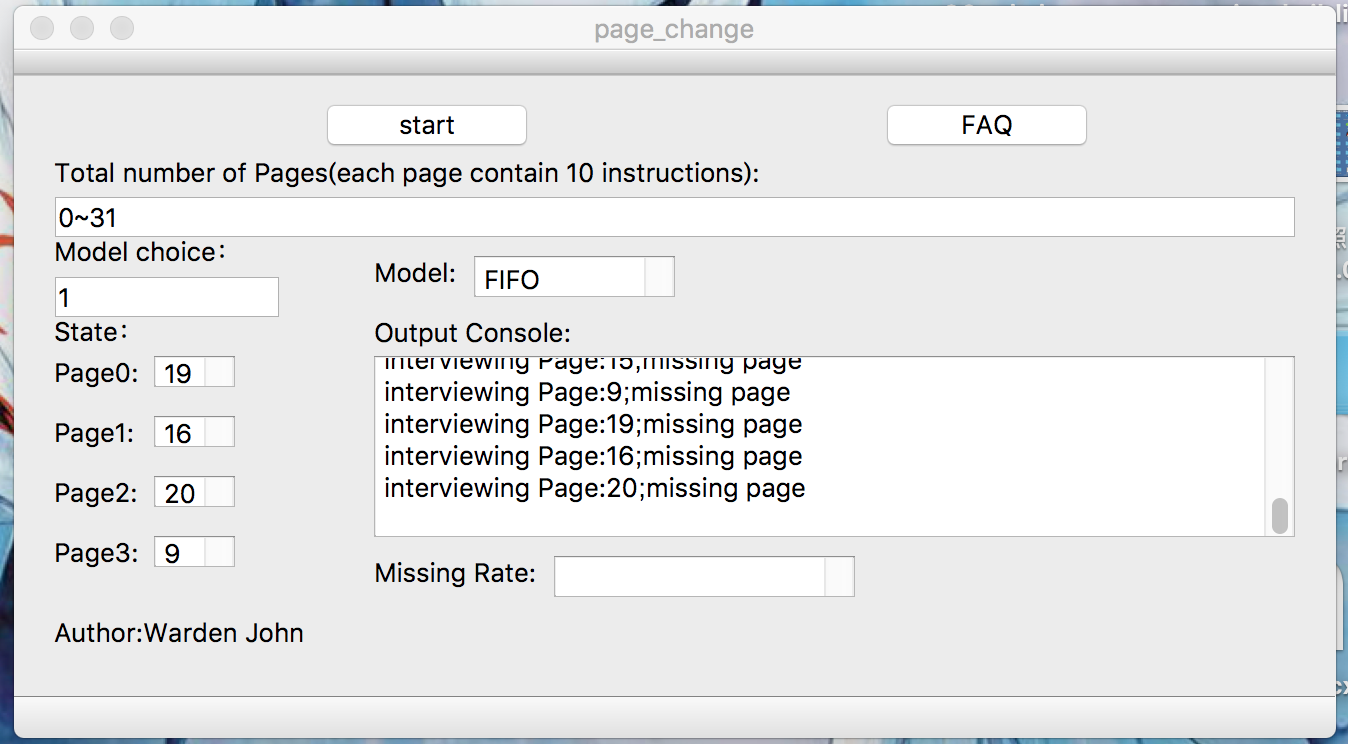
主体函数

# 五、程序的过程展示

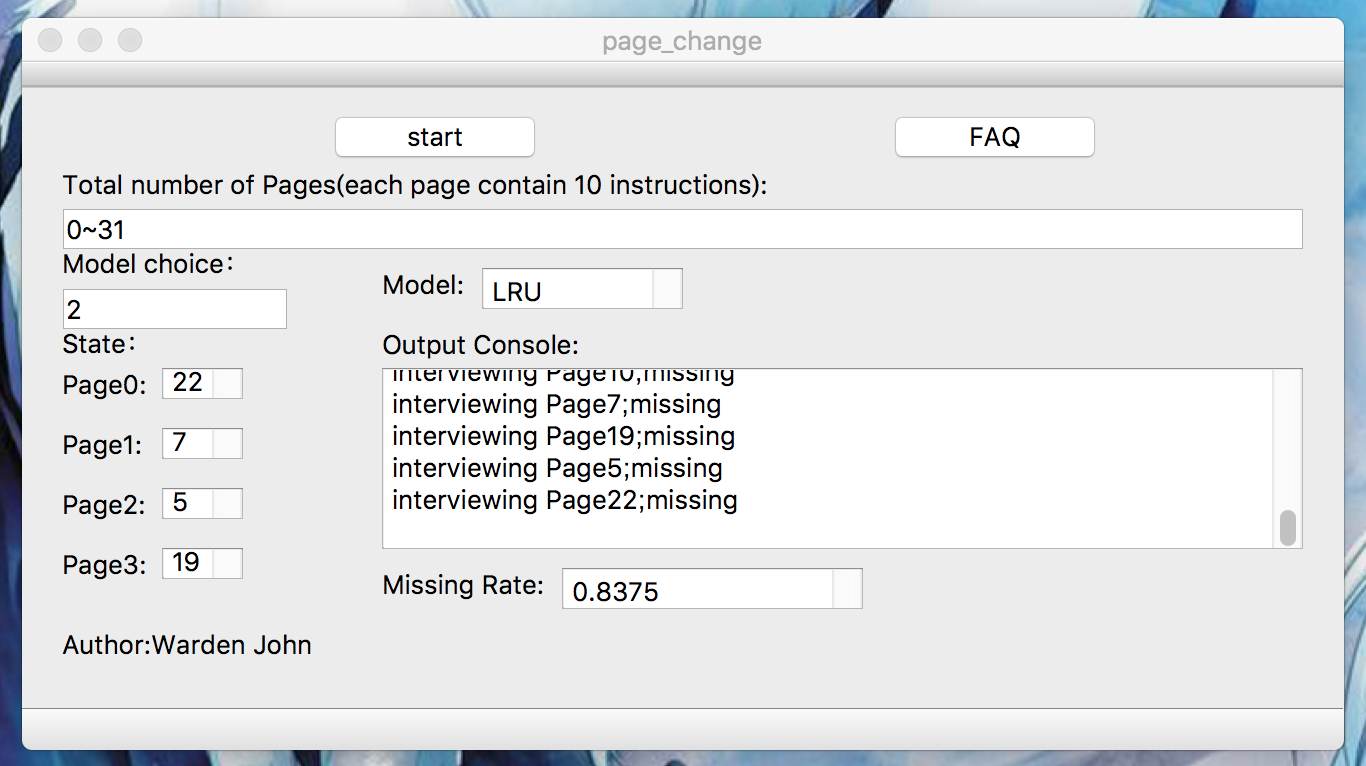
## 5.1 程序运行过程展示

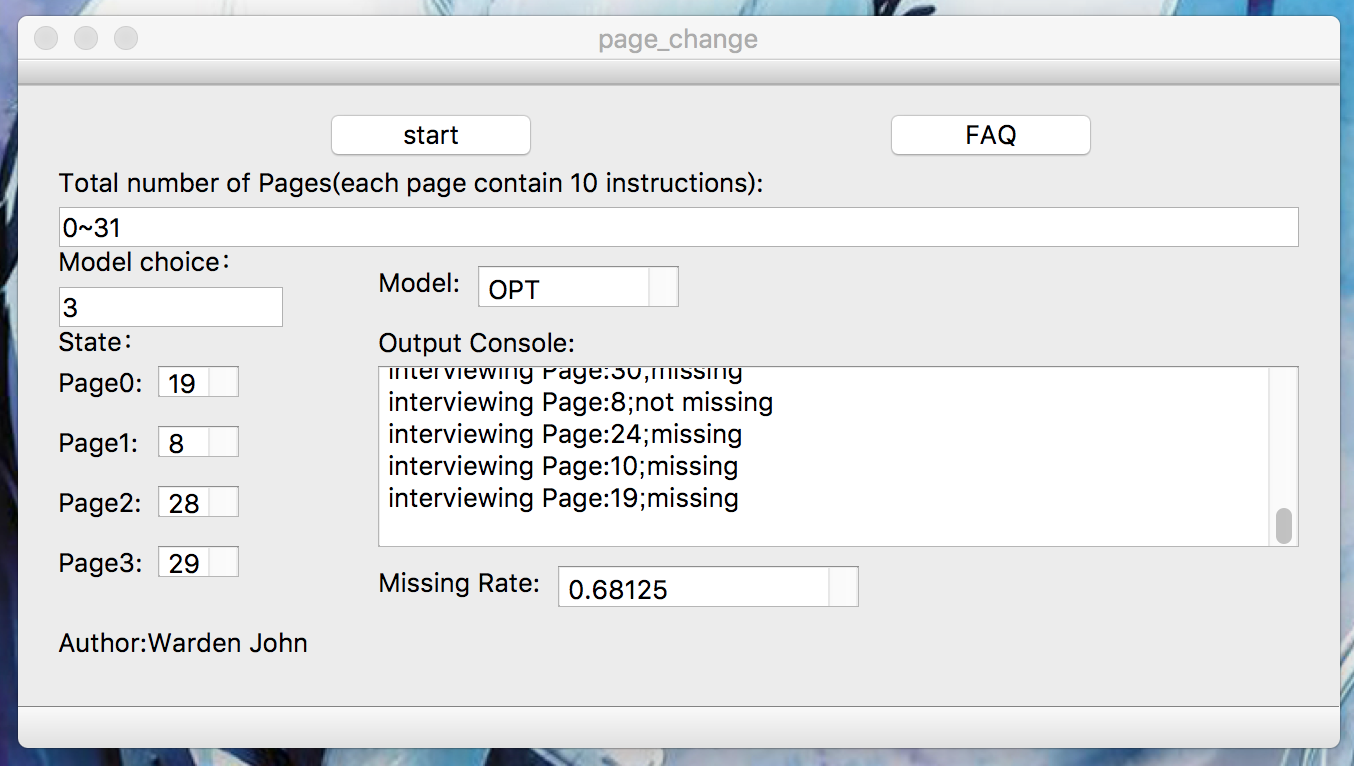
1.程序运行的初始界面

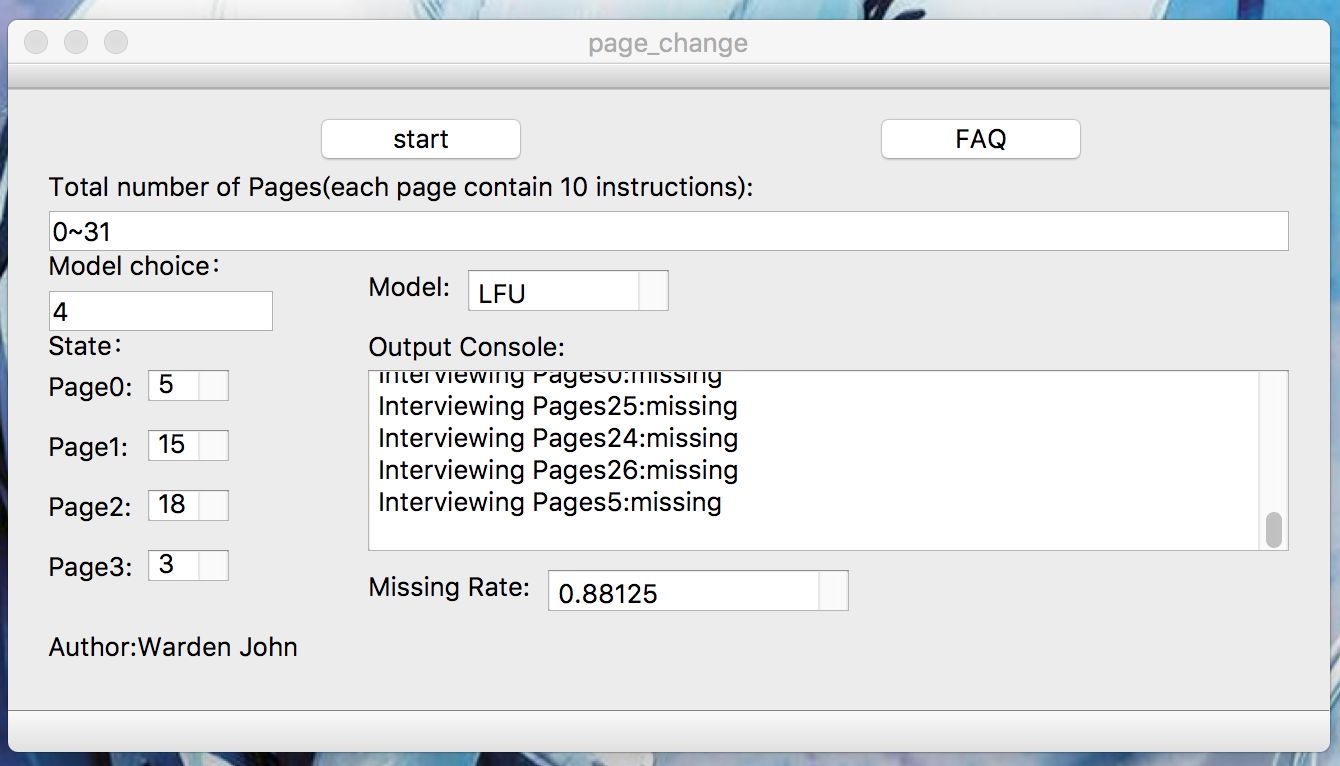
2.选择了模式后的运行状态

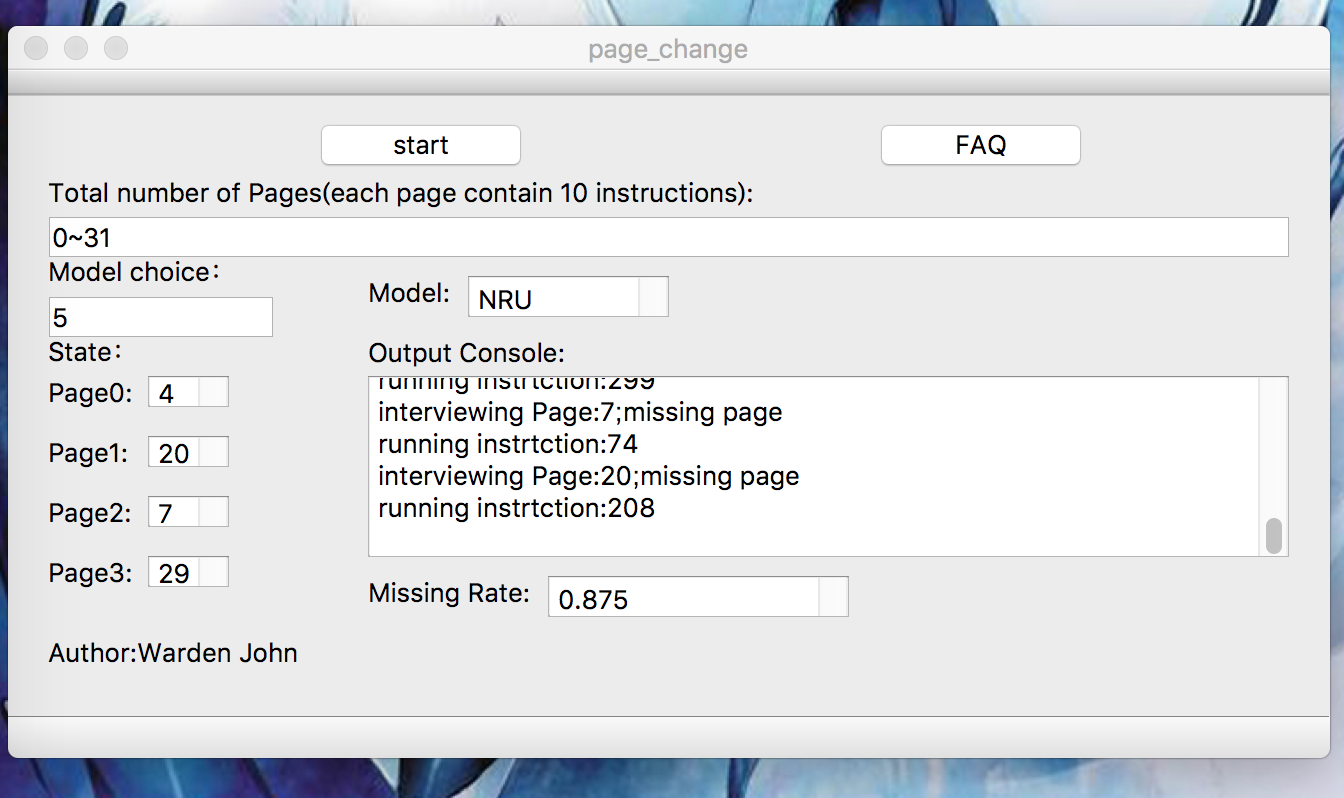


## 5.2 程序结果展示及分析../屏幕快照%202017-07-03%20下午8.46.20.png

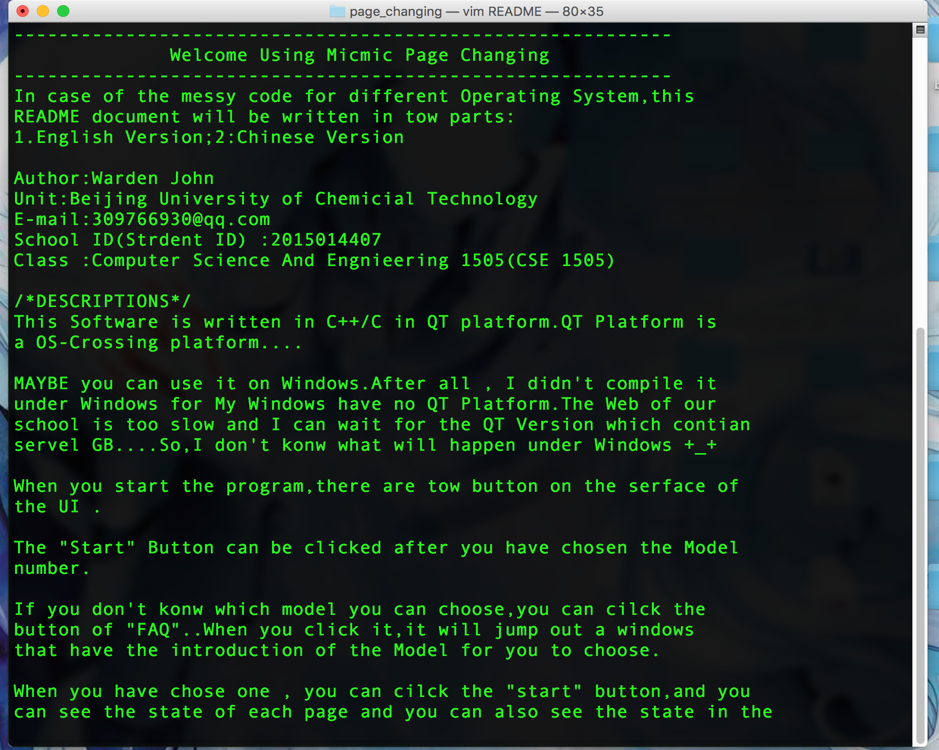








3.README使用界面



二：结果分析：

在程序运行结果中，发现除了OPT算法的缺页率明显减少外，其他的算法都差不多。但是观察在这过程中每个物理块的页面置换情况就会发现不一样。这个原因很有可能是因为由于随机数产生不规律，导致访问的序列离散，这就导致了这几种算法都的情况都有点趋近与每一次都是不同的页面而且每一次都会出现缺页的情况出现。但是不同的算法在置换的页面的时候是差别有点大的（即使在这种访问序列非常离散的情况下）

# 六、软件的辅助说明

## 6.1 本软件的优点

本软件可以让用户自由选择5种算法的实现。同时，用户在程序运行的过程中可以看到在页面置换过程中的每个物理块的状态，置换的状态，以及还可以在输出控制台中看到现在在访问那个页面，正在访问那个指令，是否缺页，缺页以后换出的是哪一个物理块的页面以及最后的缺页率等等的信息，本软件在信息上体现的比较完整，用户体验应该不错。

## 6.2 本软件的缺点

在设计的过程中的图形界面的设计不太好看，而且算法考虑的可能不是很完整或者有错漏的情况出现。而且某些功能由于时间原因还没有来得及实现，还是有较大的改善的空间。如果还有用户停止程序运行的按钮或方法满足用户和程序的交互就更好了。

## 6.3 本软件的BUG

本软件在编写的过程中难免会出现BUG。在不同函数使用的过程中，结构体内某些未使用的变量会出现没有初始化的现象。由于变量没有初始化，但是在函数执行过程中没有使用该变量，对程序运行结果没有太大的影响。BUG需要在日后的使用中慢慢发现。

# 七、课程设计结论

本次课程设计增强了自己的编程能力，而且对几个算法又有了实践性的理解。对一个小型软件的成型有了大致的认识与了解。发现在访问序列非常离散的情况下，由于经常性缺页，除了OPT以外许多算法会退化导致缺页率差别不太明显。但OPT算法是一个很好的算法。

# 八、参考文献（参考网站）

网上的各种博客，百度百科等

# 九、附录——提交文档

## 9.1 电子形式清单

* 使用说明书README
* 模拟页面置换算法报告
* subject\_design\_page\_changing文件夹（包含所有文件）（Debug非发布版）(main.cpp;page\_change.cpp;page\_change.h;page\_change.ui;page\_change.pro;page\_change.pro.user)