# 操作系统内存管理项目二

## 说明文档

1452716

张尹嘉

## **一.项目分析**

### **1.题目要求**

假设每个页面可存放10条指令，分配给一个作业的内存块为4。模拟一个作业的执行过程，该作业有320条指令，即它的地址空间为32页，目前所有页还没有调入内存。

在模拟过程中，如果所访问指令在内存中，则显示其物理地址，并转到下一条指令；如果没有在内存中，则发生缺页，此时需要记录缺页次数，并将其调入内存。如果4个内存块中已装入作业，则需进行页面置换。

所有320条指令执行完成后，计算并显示作业执行过程中发生的缺页率。置换算法可以选用FIFO或者LRU算法

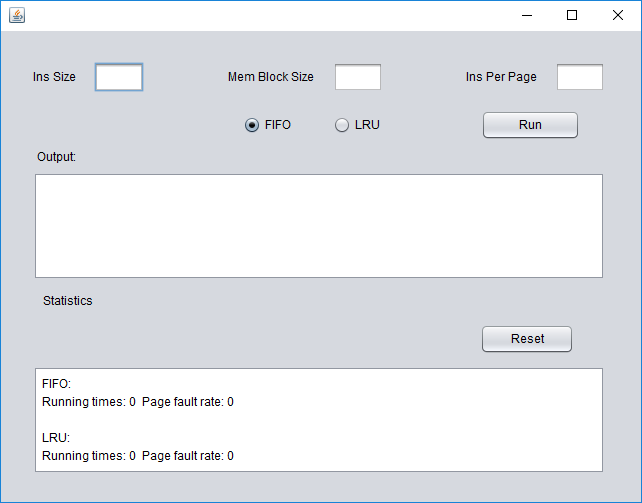
### 2.题目分析

根据题目要求，要求在指定指令条数，作业内存块数和每块内存指令数的情况下，运用FIFO和LRU两种算法进行缺页情况下的页面置换，并记录缺页率。

## **二.项目环境：**

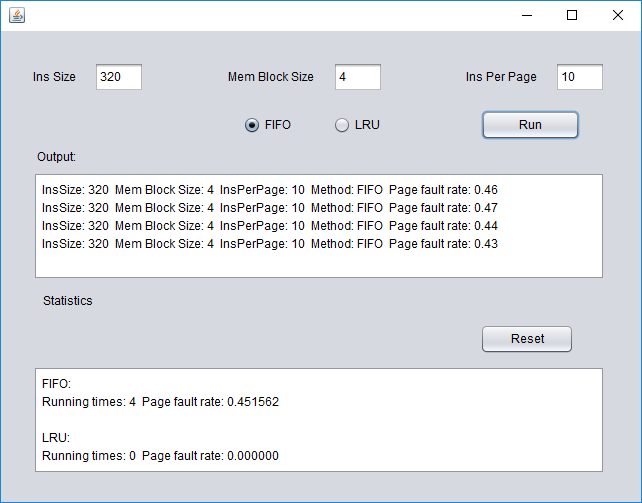
## 本程序使用java语言进行开发，并在NetBeans上进行界面的设计制作。开发时使用的JDK版本是JDK8，但在生成时同时生成了JDK8和JDK7两个版本。具体程序在VersionJDK7和VersionJDK8两个文件夹中。源代码保存在SourceCode文件夹中，以.java文件格式形成。

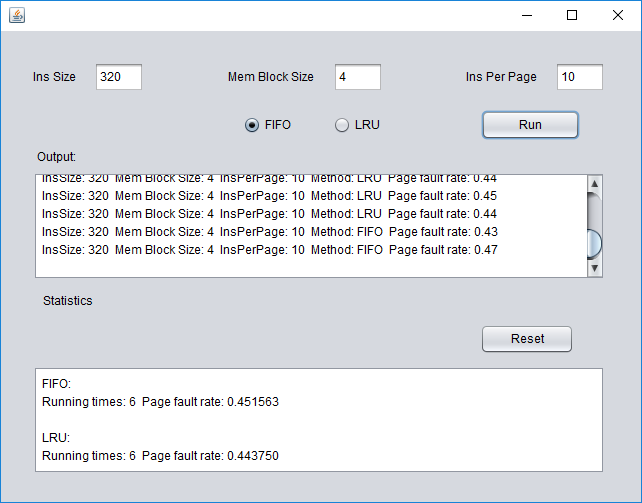
## 三.使用说明



以上是程序界面，程序共由三个部分组成，最上边的是控制部分，Ins Size代表指令的条数， Men Block代表每个作业的内存块数， Ins Per Page代表每块内存的指令条数。使用者可以通过下方FIFO和LRU选择算法，输入完参数后按Run确认。中间的是输出部分，会根据输入的参数进行模拟，并输出该次执行完毕后的缺页率。最后一个部分是统计窗口，会记录FIFO和LRU算法各执行了几次以及平均的缺页率。在输出窗口上方有Reset按钮，用来归零统计数字。建议每次改变控制部分的参数后都归零一次。

以下是程序运行截图：





## 四.算法实现

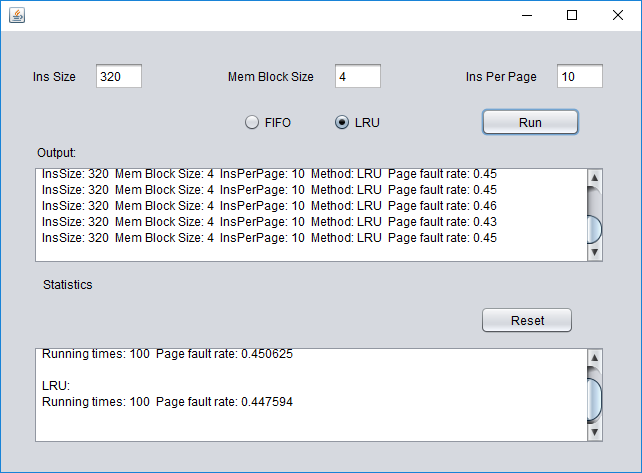
程序的目的是验证LRU和FIFO算法，缺页替换的算法就是按照这两个算法的进行实现的。FIFO循环依次替换每个页面。而LRU则是通过栈来进行实现的，即把最近使用的页面压入栈顶，替换栈底的页面。

另一方面，在指令分布上，采用了每两个指令一组，每组之间顺序执行，组与组之间运用随机数跳转执行。这种做法可以保证顺序执行的指令占到50%，且均匀分布在前后地址部分。项目课件上的方法存在一些错误，假如在执行第m条指令后随机跳转到m-1条指令，那么再顺序执行会重复执行。如果采用忽略该指令已经被执行过而继续执行该指令，320条指令全部被执行过一遍需要执行的次数在1500到2200次之间，而如果不执行之前被执行过的指令顺序向下查找，不能保证50%的顺序执行。当运行次数少时，该问题不是很明显，当大部分指令都被运行过后，这种问题会比较突出。因此该程序并没有采用课件上的指令分配算法。

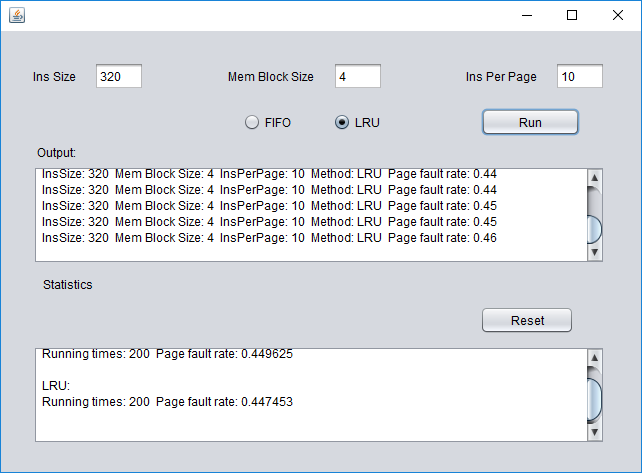
另外，运用该程序的随机数找到下一组指令（2条）时，如果该指令已经被执行过，就会再随机寻找，直到找到了下一组未被执行的指令。但是当大部分指令都被执行过后，这种算法会耗费很多时间去找下一个指令。因此在 已被执行过的指令数/总指令数 大于一定比例后，我会从头顺序寻找第一个未被访问的指令做为下一个指令。经过实验， 该比例在0.93~0.95之间基本不会出现时间复杂度过高的情况。

下面是分别执行100次和200次的结果图和统计比较：

100次：



200次：



## 五.代码实现

程序的代码由前端类，控制器接口类以及两个控制器实现类组成。前段类在接收到确认信息后，会调用控制类接口的函数，函数再根据指定的控制器实现类来执行指定数目的指令，并返回缺页率。

Fronter.java文件中是前端类，主要负责界面的布局统计数字的运算以及调用控制器接口。

Controller.java类是控制器接口，提供了供前段类调用和控制器实现类实现的函数接口。

FIFOConreoller.java和LRUController.java是两个控制器实现类，分别用不同算法实现了Controller.java中的接口。

public class Fronter：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 权限 | 返回值 | 函数名 | 参数 | 说明 |
| public | 无 | Fronter | 无 | 构造函数 |
| public | void | main | String args[] | 主类入口 |
| private | void | FIFORadioButtonActionPerformed | java.awt.event.ActionEvent evt | 设置FIFO方法 |
| private | void | LRURadioButtonActionPerformed | java.awt.event.ActionEvent evt | 设置LRU方法 |
| private | void | RunActionPerformed | java.awt.event.ActionEvent evt | 调用控制类接口 |
| private | void | ResetButtonActionPerformed | java.awt.event.ActionEvent evt | 统计数据清零 |

public interface Controller:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 权限 | 返回值 | 函数名 | 参数 | 说明 |
| public | double | runInstructions | 无 | 运行指令的接口 |

public class LRUController implements Controller：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 权限 | 返回值 | 函数名 | 参数 | 说明 |
| public | 无 | LRUController | int paraInstructionSize, int paraPageSize, int paraIPP | 构造函数 |
| public | double | runInstructions |  | 实现Controller接口，调用私有方法 |
| private | double | privateRunInstructions |  | 执行指令的私有方法 |
| private | int | getNextIns | int VisitTimes, int CurIns | 获取下一条指令 |
| private | int | checkExistInPageTable | int InsIndex | 检查该页号是否在作业的内存块中,如果存在返回页数，不存在返回-1 |
| private | void | swap | int curBlock | 当不缺页时修改栈的内容 |

public classFIFOController implements Controller：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 权限 | 返回值 | 函数名 | 参数 | 说明 |
| public | 无 | FIFOController | int paraInstructionSize, int paraPageSize, int paraIPP | 构造函数 |
| public | double | runInstructions |  | 实现Controller接口，调用私有方法 |
| private | double | privateRunInstructions |  | 执行指令的私有方法 |
| private | int | getNextIns | int VisitTimes, int CurIns | 获取下一条指令 |
| private | boolean | checkExistInPageTable | int InsIndex | 检查该页号是否在作业的内存块中 |