

# 操作系统课程项目 二

## 内存管理/请求调页模拟

邹笑寒 1652746

### 1.项目背景

#### 内容

- 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。

#### 模拟过程

- 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。

#### 基本算法

- FIFO
- LRU

### 2. 需求分析

- 产生320条随机指令
  - 地址形成规则：
    - 50%的指令是顺序执行的，25%是均匀分布在前地址部分，25%是均匀分布在后地址部分
  - 随机指令产生规则：
    1. 在(0, 319)条指令之间，随机选取一个起始执行指令，序号为m
    2. 顺序执行下一条指令(m+1)
    3. 通过随机数，跳转到前地址部分(0, m-1)中的某个指令处，序号为m1
    4. 顺序执行下一条指令(m1+1)
    5. 通过随机数，跳转到后地址部分(m1+2, 319)中的某条指令处，序号为m2
    6. 顺序执行下一条指令(m2+1)
    7. 重复步骤1-7，直到执行完320条指令

- 当指令到来时，首先查询4个物理模块中是否含有当前指令，如果有则直接下一条指令，若没有就进行判定查找物理块中是否还有空闲盘块，有该指令则直接调入，无则要发生页面置换
- 根据用户的选择采用FIFO或LFU算法进行页面置换，完成后计算并显示缺页次数和缺页率

## 3.算法实现

### 3.1 FIFO `function FIFO()`

总是淘汰最先进入内存的页面（既在内存中驻留时间最久的页面）

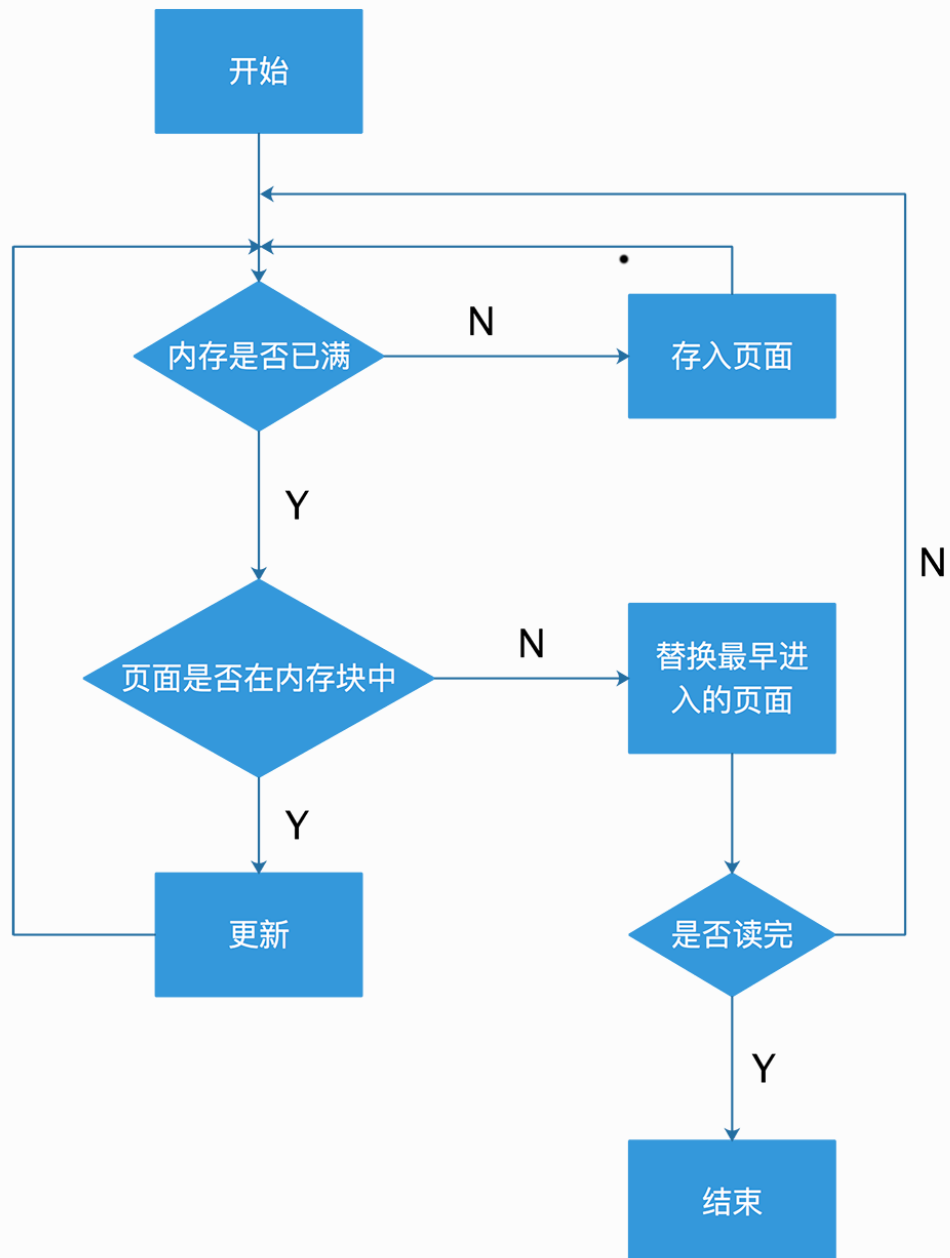
#### 1. 模拟过程：

- 当页面被置换进入内存时，将置换页面所在的物理块中访问标记设为-1;
- 并且每执行一次指令，都将物理块的访问标记自动加1;
- 需要置换时将访问标记最大的物理块中的页面置换出去;

#### 2. 缺陷：

- 性能较差，较早调入的页往往是经常被访问的页，这些页在FIFO算法下被反复调入和调出

#### 3. FIFO算法流程图



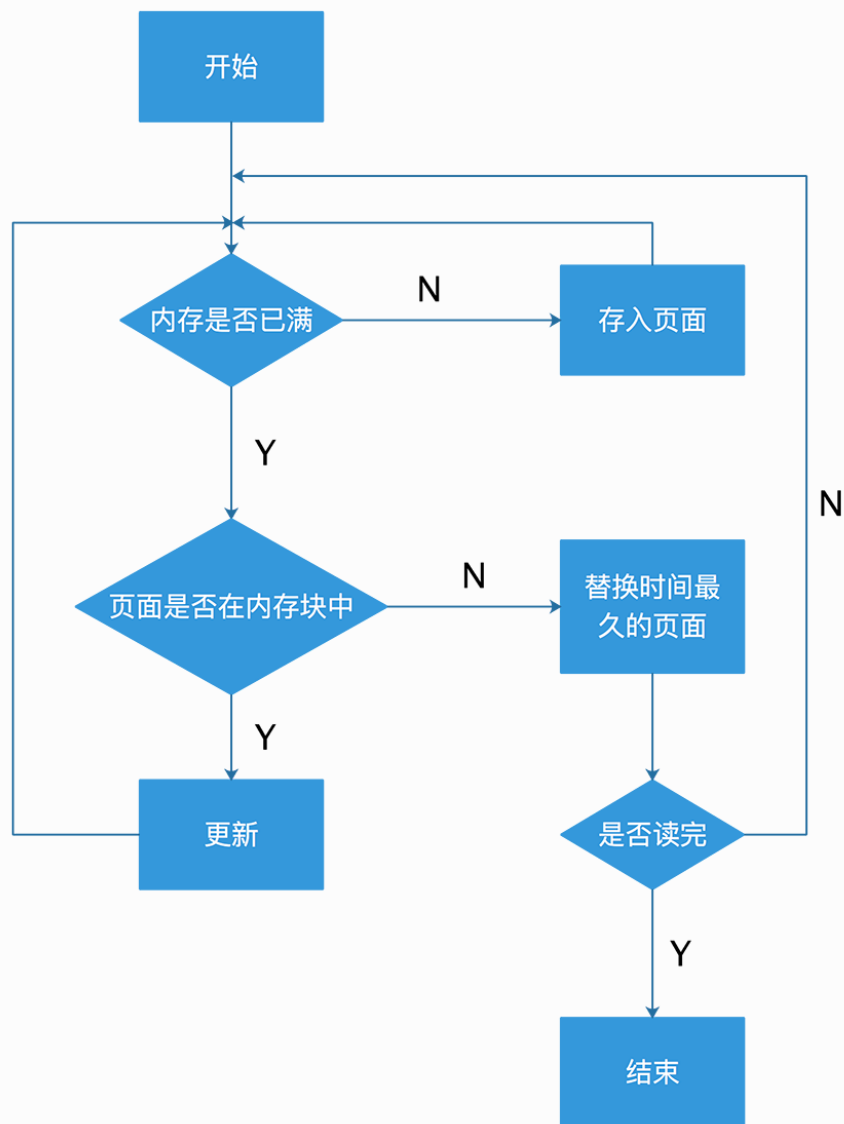
### 3.2 LRU function LRU()

以最近的过去作为不久将来的近似，将过去最长一段时间里不曾被使用的页面置换掉

#### 1. 模拟过程：

- 当物理块中的页面被访问时(包括原先存在的和后来置换进入的页面)，都将其物理块访问标记置为-1；
- 每执行一条指令，都将物理块中各页面的访问标记加1；
- 需要置换时将访问标记最大的物理块中的页面置换出去；

2. 缺陷：
- 由于需要记录页面使用时间的先后关系，硬件开销太大
3. LRU算法流程图



## 4.函数、变量解释

main.js

名称	说明
function flagInstructionExecuted()	判断选中的指令是否被运行过
function flagInstructionAvailable()	判断选中指令是否在内存中

function Pre()	初始化
function FIFO()	FIFO算法
function LRU()	LRU算法
function chooseAlgrithm()	选择使用哪个算法
var startBtn = document.getElementById("startBtn")	获取“开始”按钮
var memoryBlockNumber	内存块数
var instructionNumber	总指令数
var numberOfInstructionsInEachPage	每页存放指令数
var currentInstructionSpan	当前指令
var numberOfMissingPagesSpan	缺页数
var pageFaultRateSpan	缺页率
var memory	内存
var instructions	指令是否被执行
var insCount	记录执行的指令个数
var missingPage	缺页个数

## 5.系统实现

### 5.1 算法选择



点击对应按钮可以选择算法。

### 5.2 参数信息、模拟结果显示及“开始模拟”按钮

参数信息

内存块数: 4

总指令数: 320

每页存放指令数: 10

模拟结果

当前指令: 141

缺页次数: 191

缺页率: 0.596875

开始模拟

左边框显示参数信息，右边框在运行结束后将显示模拟结果。点击“开始模拟”按钮将开始执行对应算法。

### 5.3 过程显示

模拟运行信息

指令	内存块1	内存块2	内存块3	内存块4	说明
39	3	Empty	Empty	Empty	发生缺页，指令39不在内存中，将指令39所在的页调入内存，替换块2
40	3	4	Empty	Empty	发生缺页，指令40不在内存中，将指令40所在的页调入内存，替换块3
26	3	4	2	Empty	发生缺页，指令26不在内存中，将指令26所在的页调入内存，替换块4
27	3	4	2	Empty	指令27在内存块3中
234	3	4	2	23	发生缺页，指令234不在内存中，将指令234所在的页调入内存，替换块1
235	3	4	2	23	指令235在内存块4中
172	17	4	2	23	发生缺页，指令172不在内存中，将指令172所在的页调入内存，替换块2
173	17	4	2	23	指令173在内存块1中
229	17	22	2	23	发生缺页，指令229不在内存中，将指令229所在的页调入内存，替换块3
230	17	22	2	23	指令230在内存块4中

表格将显示运行中的信息，包括当前指令、当前4个内存块中的页号和当前所执行的步骤。

## 6.开发环境

语言：

html、css、js

## 7.提交内容

- 源代码 把 

index.html

 在浏览器中打开即可运行
- 说明文档