**北京邮电大学软件学院、**

**2017-2018学年第1学期实验报告**

**课程名称： 操作系统**

**实验名称： 实验五 虚拟存储器管理**

**实验完成人：**

**姓名：**\_\_\_\_\_苏可欣\_\_\_**学号：**\_\_\_2016211954\_\_\_\_\_**成绩：**\_\_\_\_\_\_\_\_

**指导教师：**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_陈晋鹏\_\_\_\_\_\_\_\_\_\_

**日 期： 2018 年 6 月 20**日

**1 实验目的**

1. 了解虚拟存储技术的特点；

2. 掌握请求页式管理的页面置换算法。

**2 实验内容**

1. 通过随机数产生一个指令序列，共 320 条指令。其地址按下述原则生成：

（1）50%的指令是顺序执行的；

（2）50%的指令是均匀分布在前地址部分；

（3）50%的指令是均匀分布在后地址部分；

具体的实施方法是：

A. 在[0，319]的指令地址之间随机选取一起点 M；

B. 顺序执行一条指令，即执行地址为 M+1 的指令；

C. 在前地址[0，M+1]中随机选取一条指令并执行，该指令的地址为 M’；

D. 顺序执行一条指令，其地址为 M’+1；

E. 在后地址[M’+2，319]中随机选取一条指令并执行；

F. 重复 A—E，直到执行 320 次指令。

2. 指令序列变换成页地址流

设：

（1）页面大小为 1K；

（2）用户内存容量为 4 页到 32 页；

（3）用户虚存容量为 32K。

在用户虚存中，按每 K 存放 10 条指令排列虚存地址，即 320 条指令在虚存中的存放方式为：

第 0 条—第 9 条指令为第 0 页（对应虚存地址为[0，9]）；

第 10 条—第 19 条指令为第 1 页（对应虚存地址为[10，19]）；

……………………

第 310 条—第 319 条指令为第 31 页（对应虚存地址为[310，319]）；

按以上方式，用户指令可组成 32 页。

3. 计算并输出下述各种算法在不同内存容量下的命中率。

A. 先进先出（FIFO）页面置换算法

B. 最近最久未使用（LRU）页面置换算法--最近最少使用算法

C. 最少使用（LFR）页面置换算法

D. 最佳（Optimal）页面置换算法

**3. 实验准备**

1. 先进先出（FIFO）页面置换算法

该算法总是淘汰最新进入内存的页面，即选择在内存中驻留时间最久的页面予以淘汰。该算法实现简单，只需把一个进程已调入内存的页面，按先后次序链接成一个队列，并设置一个指针，称为替换指针，使它总是指向最老的页面。

2. 最近最久未使用（LRU）页面置换算法

最近最久未使用（LRU）页面置换算法，是根据页面调入内存后的使用情况进行决策的。由于无法预测各页面将来的使用情况，只能利用“最近的过去”作为“最近的将来”的近似，因此，LRU 置换算法是选择最近最久未使用的页面予以淘汰。该算法赋予每个页面一个访问字段，用来记录一个页面自上次被访问以来所经历的时间 t，当需淘汰一个页面时，选择现有页面中其 t 值最大的，即最近最久未使用的页面予以淘汰。

3. 最少使用（LFR）页面置换算法

在采用该算法时，应为在内存中的每个页面设置一个移位寄存器，用来记录页面被访问的频率。该置换算法选择在最近使其使用最少的页面作为淘汰页。

4. 最佳（Optimal）页面置换算法

该算法选择的被淘汰页面，将是以后永远不使用的，或许是在最长（未来）时间内不再被访问的页面。采用该算法，通常可保证获得最低的缺页率。但由于人们目前还无法预知一个进程在内存的若干个页面中，哪一个页面是未来最长时间内不再被访问的，因而该算法是无法实现的，但可以利用该算法去评价其他算法。

提示：A.命中率=1-页面失效次数/页地址流长度

B.本实验中，页地址流长度为 320，页面失效次数为每次访问相应指令时，该指令所对应的页不在内存的次数。

C.关于随机数产生方法，采用 VC 系统提供函数 RAND()和 RANDOMIZE()来产生

**4. 程序示例**

自己补充。

**5. 实验结果**

结果展示以及分析原因。

看以下输出结果不难发现

1. 总体上opt的命中率要高于其他页面置换算法，这是意料之中的结果。
2. 随着页框的增多，命中率在提高，不过遗憾的是在这组数据下没能观察到FIFO独有的belady’s anomaly。
3. 并且让我惊讶的是，FIFO这种看似原始粗暴的方法其实命中率相比其他算法也没有多低，甚至在大部分时候（特别是页框比较少的时候）的时候甚至一度大于其他算法。这一点我觉得不太正常，我觉得应该是指令序列产生的方式并没有很好地体现局部性原理。
4. 最终32个页框的时候，四个算法的结果都一样。因为总共就32页嘛。0.9主要是因为最初的时候都是空闲的，每一个新访问都不会命中。直到所有的都调入内存了，也就不会再置换了。

------------通过随机数产生一个指令序列，共 320 条指令------------

206 147 148 298

246 5 6 80

152 68 69 75

222 126 127 301

174 129 130 175

95 67 68 151

144 115 116 287

23 15 16 309

49 45 46 293

295 102 103 126

140 16 17 56

25 3 4 162

134 30 31 169

290 1 2 48

70 36 37 74

102 71 72 286

97 29 30 79

32 19 20 117

139 26 27 37

281 210 211 265

133 91 92 196

101 31 32 227

99 38 39 256

295 78 79 154

173 96 97 138

108 80 81 300

248 211 212 228

35 17 18 234

180 67 68 293

303 99 100 177

83 76 77 236

241 182 183 302

146 69 70 255

165 139 140 210

96 19 20 177

114 26 27 211

213 162 163 276

101 71 72 262

186 11 12 271

232 23 24 85

193 12 13 64

245 148 149 172

236 153 154 289

83 2 3 204

195 180 181 200

182 120 121 263

57 24 25 304

220 97 98 250

109 40 41 304

208 44 45 209

37 8 9 299

317 54 55 260

27 8 9 208

69 11 12 313

156 129 130 152

286 110 111 222

291 231 232 313

136 14 15 64

204 190 191 309

4 3 4 105

252 134 135 142

310 196 197 254

132 21 22 126

272 4 5 245

142 9 10 167

315 213 214 284

262 16 17 26

265 209 210 296

198 129 130 296

133 127 128 308

144 2 3 62

67 22 23 297

114 59 60 161

69 66 67 309

254 247 248 318

293 221 222 304

100 63 64 250

173 58 59 232

201 94 95 282

106 37 38 302

-----------2.指令序列变换成页地址流-------

20-14-14-29-24-0-0-8-15-6-6-7-22-12-12-30-17-12-13-17-9-6-6-15-14-11-11-28-2-1-1-30-4-4-4-29-29-10-10-12-14-1-1-5-2-0-0-16-13-3-3-16-29-0-0-4-7-3-3-7-10-7-7-28-9-2-3-7-3-1-2-11-13-2-2-3-28-21-21-26-13-9-9-19-10-3-3-22-9-3-3-25-29-7-7-15-17-9-9-13-10-8-8-30-24-21-21-22-3-1-1-23-18-6-6-29-30-9-10-17-8-7-7-23-24-18-18-30-14-6-7-25-16-13-14-21-9-1-2-17-11-2-2-21-21-16-16-27-10-7-7-26-18-1-1-27-23-2-2-8-19-1-1-6-24-14-14-17-23-15-15-28-8-0-0-20-19-18-18-20-18-12-12-26-5-2-2-30-22-9-9-25-10-4-4-30-20-4-4-20-3-0-0-29-31-5-5-26-2-0-0-20-6-1-1-31-15-12-13-15-28-11-11-22-29-23-23-31-13-1-1-6-20-19-19-30-0-0-0-10-25-13-13-14-31-19-19-25-13-2-2-12-27-0-0-24-14-0-1-16-31-21-21-28-26-1-1-2-26-20-21-29-19-12-13-29-13-12-12-30-14-0-0-6-6-2-2-29-11-5-6-16-6-6-6-30-25-24-24-31-29-22-22-30-10-6-6-25-17-5-5-23-20-9-9-28-10-3-3-30-

----------3. 计算并输出下述各种算法在不同内存容量下的命中率--------

Page FIFO LRU OPT LFR

4 0.294 0.297 0.481 0.325

5 0.309 0.319 0.525 0.337

6 0.353 0.356 0.566 0.350

7 0.372 0.366 0.603 0.375

8 0.409 0.397 0.628 0.400

9 0.438 0.428 0.653 0.425

10 0.469 0.459 0.675 0.444

11 0.475 0.488 0.694 0.453

12 0.531 0.519 0.712 0.488

13 0.572 0.538 0.731 0.512

14 0.578 0.559 0.747 0.547

15 0.587 0.603 0.762 0.562

16 0.613 0.613 0.778 0.572

17 0.637 0.644 0.794 0.600

18 0.656 0.659 0.806 0.625

19 0.697 0.694 0.819 0.650

20 0.706 0.716 0.831 0.684

21 0.712 0.738 0.844 0.709

22 0.716 0.747 0.853 0.741

23 0.769 0.759 0.863 0.766

24 0.762 0.778 0.869 0.800

25 0.806 0.812 0.875 0.803

26 0.816 0.825 0.881 0.806

27 0.831 0.837 0.887 0.831

28 0.856 0.850 0.891 0.837

29 0.875 0.863 0.894 0.847

30 0.884 0.866 0.897 0.869

31 0.887 0.891 0.900 0.884

32 0.900 0.900 0.900 0.900