

第八次实验：存储管理模拟

Hollow Man

一、实验目的与要求：

通过模拟实现请求页式存储管理的几种基本页面置换算法，了解虚拟存储技术的特点，掌握虚拟存储请求页式存储管理中几种基本页面置换算法的基本思想和实现过程并比较它们的效率。

二、实验环境：（硬件环境、软件环境）

1. LINUX 操作系统和文本编辑器

三、实验内容：（原理、操作步骤等）

设计一个虚拟存储区和内存工作区，并使用下述算法计算访问命中率。

1、最佳淘汰算法（OPT）

2、先进先出的算法（FIFO）

3、最近最久未使用算法（LRU）

```
#include <stdlib.h>
#include <stdio.h>
#define pagesize 8 //页面尺寸大小
typedef struct BLOCK //声明一种新类型--物理块类型
{
    int pagenum; //页号
    int accessed; //访问量，其值表示多久未被访问
} BLOCK;
int pc; //程序计数器，用来记录对应的页号
int n; //缺页计数器，用来记录缺页的次数
static int num[320]; //用来存储 320 条随机数
BLOCK block[pagesize]; //定义一大小为 8 的物理块数组
void init(); //程序初始化函数
int findExist(int curpage); //查找物理块中是否有该页面
int findSpace(); //查找是否有空闲物理块
int findReplace(); //查找应予置换的页面
void display(); //显示
void random(); //产生 320 条随机数,显示并存储到 num[320]
void pagestring(); //显示调用的页面队列
void OPT();
void LRU();
void FIFO();
void main()
{
    int select;
    printf("请输入第一个随机数(0~320):");
    random();
    printf("*****对应的调用页面队列*****\n");
```

```

    pagestring();
do
{
    printf("*****\n");
    printf("-----1:OPT 2:LRU 3:FIFO 4:退出-----\n");
    printf("*****\n");
    printf("请选择一种页面置换算法:");
    scanf("%d", &select);
    printf("*****\n");
    init();
    switch (select)
    {
        case 1:
            printf("最佳置换算法 OPT:\n");
            printf("*****\n");
            OPT();
            break;
        case 2:
            printf("最近最久未使用置换算法 LRU:\n");
            printf("*****\n");
            LRU();
            break;
        case 3:
            printf("先进先出置换算法 FIFO:\n");
            printf("*****\n");
            FIFO();
            break;
    }
} while (select != 4);
}
void init()
{
    for (int i = 0; i < pagesize; i++)
    {
        block[i].pagenum = -1;
        block[i].accessed = 0;
        pc = n = 0;
    }
}
int findExist(int curpage)
{
    for (int i = 0; i < pagesize; i++)
    {
        if (block[i].pagenum == curpage)

```

```

        return i; //检测到内存中有该页面，返回 block 中的位置
    }
    return -1;
}
int findSpace()
{
    for (int i = 0; i < pagesize; i++)
    {
        if (block[i].pagenum == -1)
            return i; //找到空闲的 block，返回 block 中的位置
    }
    return -1;
}
int findReplace()
{
    int pos = 0;
    for (int i = 0; i < pagesize; i++)
    {
        if (block[i].accessed > block[pos].accessed)
            pos = i; //找到应该置换页面，返回 BLOCK 中位置
    }
    return pos;
}
void display()
{
    for (int i = 0; i < pagesize; i++)
    {
        if (block[i].pagenum != -1)
        {
            printf(" %02d", block[i].pagenum);
        }
    }
    printf("\n");
}
void random()
{
    int flag = 0;
    scanf("%d", &pc);
    printf("*****按照要求产生的 320 个随机数: *****\n");
    for (int i = 0; i < 320; i++)
    {
        num[i] = pc;
        if (flag % 2 == 0)
            pc = ++pc % 320;
    }
}

```

```

        if (flag == 1)
            pc = rand() % (pc - 1);
        if (flag == 3)
            pc = pc + 1 + (rand() % (320 - (pc + 1)));
        flag = ++flag % 4;
        printf(" %03d", num[i]);
        if ((i + 1) % 10 == 0)
            printf("\n");
    }
}

void pagestring() //显示调用的页面队列,页面号取法为随机数除 10 取整
{
    for (int i = 0; i < 320; i++)
    {
        printf(" %02d", num[i] / 10);
        if ((i + 1) % 10 == 0)
            printf("\n");
    }
}

void OPT() //最佳替换算法
{
    int exist,
        space, position;
    int curpage;
    for (int i = 0; i < 320; i++)
    {
        pc = num[i];
        curpage = pc / 10;
        exist = findExist(curpage);
        if (exist == -1)
        {
            space = findSpace();
            if (space != -1)
            {
                block[space].pagenum = curpage;
                display();
                n = n + 1;
            }
            else
            {
                for (int k = 0; k < pagesize; k++)
                {
                    for (int j = i; j < 320; j++)
                    {

```

```

        if (block[k].pagenum != num[j] / 10)
        {
            block[k].accessed = 1000;
        } //将来不会用，设置为一个很大数
        else
        {
            block[k].accessed = j;
            break;
        }
    }
    position = findReplace();
    block[position].pagenum = curpage;
    display();
    n++;
}
}
printf("缺页次数:%d\n", n);
printf("缺页率:%f%%\n", (n / 320.0) * 100);
}
void LRU() //最近最久未使用算法
{
    int exist, space, position;
    int curpage;
    for (int i = 0; i < 320; i++)
    {
        pc = num[i];
        curpage = pc / 10;
        exist = findExist(curpage);
        if (exist == -1)
        {
            space = findSpace();
            if (space != -1)
            {
                block[space].pagenum = curpage;
                display();
                n = n + 1;
            }
        }
        else
        {
            position = findReplace();
            block[position].pagenum = curpage;
            display();
        }
    }
}

```

```

        n++;
    }
}
else
    block[exist].accessed = -1; //恢复存在的并刚访问过的 BLOCK 中页面
accessed 为-1
    for (int j = 0; j < pagesize; j++)
    {
        block[j].accessed++;
    }
}
printf("缺页次数:%d\n", n);
printf("缺页率:%f%%\n", (n / 320.0) * 100);
}
void FIFO() //先进先出算法
{
    int exist, space, position;
    int curpage;
    for (int i = 0; i < 320; i++)
    {
        pc = num[i];
        curpage = pc / 10; //转换为页面号
        exist = findExist(curpage); //查找物理块中是否有该页面,没有的话,置为
-1
        if (exist == -1)
        {
            space = findSpace();
            //查找是否有空的物理块,没有的话,置为-1;有的话,把位置返
回
            if (space != -1)
            {
                block[space].pagenum = curpage;
                display();
                n = n + 1;
            }
            else
            {
                position = findReplace(); //没有空闲物理块,进行置换
                block[position].pagenum = curpage;
                display();
                n++;
                block[position].accessed--;
            }
        }
    }
}

```

```

        for (int j = 0; j < pagesize; j++) //把所有在页面里的页面号的访问次数加 1
            block[j].accessed++;
    }
    printf("缺页次数:%d\n", n);
    printf("缺页率:%f%%\n", (n / 320.0) * 100);
}

```

运行结果:

[hollow-virtualbox 文档]# ./a.out

请输入第一个随机数(0~320):34

*****按照要求产生的 320 个随机数: *****

```

034 035 027 028 150 151 027 028 051 052
032 033 269 270 229 230 297 298 168 169
291 292 050 051 291 292 182 183 203 204
202 203 314 315 228 229 266 267 100 101
266 267 143 144 313 314 234 235 245 246
177 178 274 275 198 199 243 244 199 200
297 298 032 033 228 229 150 151 290 291
149 150 207 208 091 092 201 202 163 164
182 183 001 002 159 160 063 064 139 140
081 082 252 253 230 231 252 253 191 192
238 239 087 088 150 151 091 092 249 250
206 207 273 274 011 012 149 150 037 038
244 245 145 146 297 298 257 258 273 274
248 249 255 256 061 062 251 252 209 210
221 222 181 182 301 302 058 059 205 206
189 190 197 198 023 024 179 180 122 123
278 279 017 018 085 086 066 067 166 167
010 011 296 297 267 268 271 272 007 008
310 311 072 073 272 273 236 237 238 239
231 232 288 289 034 035 182 183 032 033
087 088 027 028 243 244 128 129 218 219
117 118 254 255 125 126 217 218 059 060
061 062 044 045 262 263 219 220 250 251
036 037 201 202 021 022 253 254 030 031
091 092 012 013 227 228 178 179 317 318
031 032 135 136 111 112 169 170 080 081
101 102 067 068 263 264 162 163 221 222
217 218 286 287 120 121 169 170 152 153
262 263 182 183 263 264 035 036 239 240
086 087 123 124 003 004 113 114 061 062
293 294 093 094 269 270 090 091 139 140
035 036 181 182 179 180 235 236 146 147

```

*****对应的调用页面队列*****

03 03 02 02 15 15 02 02 05 05
03 03 26 27 22 23 29 29 16 16
29 29 05 05 29 29 18 18 20 20
20 20 31 31 22 22 26 26 10 10
26 26 14 14 31 31 23 23 24 24
17 17 27 27 19 19 24 24 19 20
29 29 03 03 22 22 15 15 29 29
14 15 20 20 09 09 20 20 16 16
18 18 00 00 15 16 06 06 13 14
08 08 25 25 23 23 25 25 19 19
23 23 08 08 15 15 09 09 24 25
20 20 27 27 01 01 14 15 03 03
24 24 14 14 29 29 25 25 27 27
24 24 25 25 06 06 25 25 20 21
22 22 18 18 30 30 05 05 20 20
18 19 19 19 02 02 17 18 12 12
27 27 01 01 08 08 06 06 16 16
01 01 29 29 26 26 27 27 00 00
31 31 07 07 27 27 23 23 23 23
23 23 28 28 03 03 18 18 03 03
08 08 02 02 24 24 12 12 21 21
11 11 25 25 12 12 21 21 05 06
06 06 04 04 26 26 21 22 25 25
03 03 20 20 02 02 25 25 03 03
09 09 01 01 22 22 17 17 31 31
03 03 13 13 11 11 16 17 08 08
10 10 06 06 26 26 16 16 22 22
21 21 28 28 12 12 16 17 15 15
26 26 18 18 26 26 03 03 23 24
08 08 12 12 00 00 11 11 06 06
29 29 09 09 26 27 09 09 13 14
03 03 18 18 17 18 23 23 14 14

-----1:OPT 2:LRU 3:FIFO 4:退出-----

请选择一种页面置换算法:1

最佳置换算法 OPT:

03

03 02

03 02 15

03 02 15 05

03 02 15 05 26

03 02 15 05 26 27
03 02 15 05 26 27 22
03 02 15 05 26 27 22 23
03 29 15 05 26 27 22 23
03 29 16 05 26 27 22 23
03 29 16 18 26 27 22 23
03 29 16 20 26 27 22 23
03 29 31 20 26 27 22 23
03 29 31 20 26 27 10 23
03 29 31 20 26 27 14 23
03 29 24 20 26 27 14 23
03 29 24 20 17 27 14 23
03 29 24 20 19 27 14 23
22 29 24 20 19 27 14 23
15 29 24 20 19 27 14 23
15 09 24 20 19 27 14 23
15 09 24 20 19 16 14 23
15 09 24 18 19 16 14 23
15 09 24 00 19 16 14 23
15 09 24 06 19 16 14 23
15 09 24 06 19 13 14 23
15 09 24 06 19 08 14 23
15 09 24 25 19 08 14 23
15 20 24 25 19 08 14 23
15 20 24 25 19 08 14 27
15 20 24 25 19 01 14 27
03 20 24 25 19 01 14 27
03 20 24 25 19 01 29 27
03 20 06 25 19 01 29 27
03 20 06 21 19 01 29 27
03 20 06 22 19 01 29 27
03 20 06 18 19 01 29 27
30 20 06 18 19 01 29 27
05 20 06 18 19 01 29 27
05 20 06 18 02 01 29 27
05 17 06 18 02 01 29 27
05 12 06 18 02 01 29 27
08 12 06 18 02 01 29 27
08 12 16 18 02 01 29 27
08 12 16 18 02 01 26 27
08 12 00 18 02 01 26 27
08 12 31 18 02 01 26 27
08 12 07 18 02 01 26 27
08 12 23 18 02 01 26 27

08 12 23 18 02 01 26 28
08 12 03 18 02 01 26 28
08 12 03 24 02 01 26 28
08 12 03 21 02 01 26 28
08 12 03 21 02 01 26 11
25 12 03 21 02 01 26 11
25 05 03 21 02 01 26 11
25 06 03 21 02 01 26 11
25 04 03 21 02 01 26 11
25 22 03 21 02 01 26 11
25 22 03 20 02 01 26 11
09 22 03 20 02 01 26 11
09 22 03 17 02 01 26 11
09 22 03 17 31 01 26 11
09 22 03 17 13 01 26 11
09 22 03 17 13 16 26 11
09 22 03 17 08 16 26 11
10 22 03 17 08 16 26 11
06 22 03 17 08 16 26 11
06 21 03 17 08 16 26 11
06 28 03 17 08 16 26 11
06 12 03 17 08 16 26 11
06 12 03 17 08 15 26 11
06 12 03 17 08 18 26 11
06 12 03 23 08 18 26 11
06 12 03 24 08 18 26 11
06 00 03 24 08 18 26 11
29 00 03 24 08 18 26 11
09 00 03 24 08 18 26 11
09 27 03 24 08 18 26 11
13 27 03 24 08 18 26 11
14 27 03 24 08 18 26 11
14 17 03 24 08 18 26 11
14 23 03 24 08 18 26 11

缺页次数:83

缺页率:25.937500%

-----1:OPT 2:LRU 3:FIFO 4:退出-----

请选择一种页面置换算法:2

最近最久未使用置换算法 LRU:

* 03

03 02
03 02 15
03 02 15 05
03 02 15 05 26
03 02 15 05 26 27
03 02 15 05 26 27 22
03 02 15 05 26 27 22 23
03 02 15 05 29 27 22 23
03 02 15 05 29 16 22 23
03 02 15 05 29 16 18 23
03 02 15 05 29 16 18 20
03 02 31 05 29 16 18 20
03 22 31 05 29 16 18 20
26 22 31 05 29 16 18 20
26 22 31 05 29 10 18 20
26 22 31 14 29 10 18 20
26 22 31 14 23 10 18 20
26 22 31 14 23 10 24 20
26 22 31 14 23 10 24 17
26 27 31 14 23 10 24 17
26 27 31 14 23 19 24 17
20 27 31 14 23 19 24 17
29 27 31 14 23 19 24 17
29 27 31 03 23 19 24 17
29 27 22 03 23 19 24 17
29 27 22 03 15 19 24 17
29 27 22 03 15 19 24 14
29 27 22 03 15 19 24 20
29 09 22 03 15 19 24 20
29 09 22 03 15 19 16 20
29 09 22 03 15 18 16 20
29 09 22 00 15 18 16 20
29 09 06 00 15 18 16 20
13 09 06 00 15 18 16 20
14 09 06 00 15 18 16 20
08 09 06 00 15 18 16 20
08 25 06 00 15 18 16 20
08 25 06 00 15 18 16 23
08 25 06 00 15 19 16 23
08 25 06 09 15 19 16 23
08 25 06 09 15 19 24 23
08 25 06 09 15 19 20 23
08 25 27 09 15 19 20 23
08 25 27 09 15 01 20 23

08 25 27 09 15 01 20 14
08 25 27 09 15 01 20 03
24 25 27 09 15 01 20 03
24 25 27 14 15 01 20 03
24 29 27 14 15 01 20 03
24 29 27 14 15 01 25 03
24 29 27 14 15 06 25 03
24 29 27 14 20 06 25 03
24 29 27 14 21 06 25 03
24 29 27 14 22 06 25 03
24 29 27 14 22 06 25 18
24 29 27 30 22 06 25 18
24 05 27 30 22 06 25 18
24 05 20 30 22 06 25 18
19 05 20 30 22 06 25 18
19 05 20 30 22 02 25 18
19 05 20 30 22 02 17 18
19 05 20 30 22 02 12 18
19 05 20 30 27 02 12 18
19 05 20 01 27 02 12 18
19 08 20 01 27 02 12 18
19 08 06 01 27 02 12 18
16 08 06 01 27 02 12 18
16 08 06 01 27 29 12 18
16 08 06 01 27 29 12 26
16 08 06 01 27 29 00 26
16 31 06 01 27 29 00 26
16 31 07 01 27 29 00 26
23 31 07 01 27 29 00 26
23 31 07 28 27 29 00 26
23 31 07 28 27 03 00 26
23 31 07 28 27 03 00 18
23 31 07 28 27 03 08 18
23 02 07 28 27 03 08 18
23 02 24 28 27 03 08 18
23 02 24 28 12 03 08 18
21 02 24 28 12 03 08 18
21 02 24 11 12 03 08 18
21 02 24 11 12 03 08 25
21 02 24 11 12 05 08 25
21 02 24 11 12 06 08 25
21 02 24 11 12 06 04 25
21 26 24 11 12 06 04 25
21 26 22 11 12 06 04 25

21 26 03 11 12 06 04 25
21 26 03 20 12 06 04 25
21 26 03 20 02 06 04 25
21 26 03 20 02 09 04 25
21 26 03 20 02 09 01 25
21 22 03 20 02 09 01 25
17 22 03 20 02 09 01 25
17 22 03 31 02 09 01 25
17 22 03 31 13 09 01 25
17 22 03 31 13 09 01 11
17 22 03 31 13 16 01 11
17 22 03 31 13 08 01 11
17 22 03 31 13 08 10 11
17 06 03 31 13 08 10 11
17 06 03 26 13 08 10 11
17 06 16 26 13 08 10 11
17 06 16 26 22 08 10 11
17 06 16 26 22 08 10 21
28 06 16 26 22 08 10 21
28 06 16 26 22 12 10 21
28 06 16 26 22 12 17 21
28 06 16 26 22 12 15 21
28 18 16 26 22 12 15 21
28 18 16 26 03 12 15 21
28 18 16 26 03 12 15 23
28 18 16 26 03 12 15 24
28 18 16 26 03 12 15 08
00 18 16 26 03 12 15 08
00 18 11 26 03 12 15 08
00 18 11 26 03 12 06 08
00 29 11 26 03 12 06 08
00 29 11 09 03 12 06 08
00 29 11 09 26 12 06 08
00 29 11 09 27 12 06 08
00 29 11 09 13 12 06 08
00 29 11 09 14 12 06 08
00 29 11 09 03 12 06 08
00 29 11 09 03 12 06 18
00 29 11 09 03 17 06 18
00 29 11 09 03 23 06 18
14 29 11 09 03 23 06 18

缺页次数:130

缺页率:40.625000%

-----1:OPT 2:LRU 3:FIFO 4:退出-----

请选择一种页面置换算法:3

先进先出置换算法 FIFO:

03

03 02

03 02 15

03 02 15 05

03 02 15 05 26

03 02 15 05 26 27

03 02 15 05 26 27 22

03 02 15 05 26 27 22 23

29 02 15 05 26 27 22 23

29 16 15 05 26 27 22 23

29 16 18 05 26 27 22 23

29 16 18 20 26 27 22 23

29 16 18 20 31 27 22 23

29 16 18 20 31 26 22 23

29 16 18 20 31 26 10 23

29 16 18 20 31 26 10 14

23 16 18 20 31 26 10 14

23 24 18 20 31 26 10 14

23 24 17 20 31 26 10 14

23 24 17 27 31 26 10 14

23 24 17 27 19 26 10 14

23 24 17 27 19 20 10 14

23 24 17 27 19 20 29 14

23 24 17 27 19 20 29 03

22 24 17 27 19 20 29 03

22 15 17 27 19 20 29 03

22 15 14 27 19 20 29 03

22 15 14 09 19 20 29 03

22 15 14 09 16 20 29 03

22 15 14 09 16 18 29 03

22 15 14 09 16 18 00 03

22 15 14 09 16 18 00 06

13 15 14 09 16 18 00 06

13 08 14 09 16 18 00 06

13 08 25 09 16 18 00 06

13 08 25 23 16 18 00 06

13 08 25 23 19 18 00 06

13 08 25 23 19 15 00 06

13 08 25 23 19 15 09 06
13 08 25 23 19 15 09 24
20 08 25 23 19 15 09 24
20 27 25 23 19 15 09 24
20 27 01 23 19 15 09 24
20 27 01 14 19 15 09 24
20 27 01 14 03 15 09 24
20 27 01 14 03 29 09 24
20 27 01 14 03 29 25 24
20 27 01 14 03 29 25 06
21 27 01 14 03 29 25 06
21 22 01 14 03 29 25 06
21 22 18 14 03 29 25 06
21 22 18 30 03 29 25 06
21 22 18 30 05 29 25 06
21 22 18 30 05 20 25 06
21 22 18 30 05 20 19 06
21 22 18 30 05 20 19 02
17 22 18 30 05 20 19 02
17 12 18 30 05 20 19 02
17 12 27 30 05 20 19 02
17 12 27 01 05 20 19 02
17 12 27 01 08 20 19 02
17 12 27 01 08 06 19 02
17 12 27 01 08 06 16 02
17 12 27 01 08 06 16 29
26 12 27 01 08 06 16 29
26 00 27 01 08 06 16 29
26 00 31 01 08 06 16 29
26 00 31 07 08 06 16 29
26 00 31 07 27 06 16 29
26 00 31 07 27 23 16 29
26 00 31 07 27 23 28 29
26 00 31 07 27 23 28 03
18 00 31 07 27 23 28 03
18 08 31 07 27 23 28 03
18 08 02 07 27 23 28 03
18 08 02 24 27 23 28 03
18 08 02 24 12 23 28 03
18 08 02 24 12 21 28 03
18 08 02 24 12 21 11 03
18 08 02 24 12 21 11 25
05 08 02 24 12 21 11 25
05 06 02 24 12 21 11 25

05 06 04 24 12 21 11 25
05 06 04 26 12 21 11 25
05 06 04 26 22 21 11 25
05 06 04 26 22 03 11 25
05 06 04 26 22 03 20 25
05 06 04 26 22 03 20 02
25 06 04 26 22 03 20 02
25 09 04 26 22 03 20 02
25 09 01 26 22 03 20 02
25 09 01 17 22 03 20 02
25 09 01 17 31 03 20 02
25 09 01 17 31 13 20 02
25 09 01 17 31 13 11 02
25 09 01 17 31 13 11 16
08 09 01 17 31 13 11 16
08 10 01 17 31 13 11 16
08 10 06 17 31 13 11 16
08 10 06 26 31 13 11 16
08 10 06 26 22 13 11 16
08 10 06 26 22 21 11 16
08 10 06 26 22 21 28 16
08 10 06 26 22 21 28 12
16 10 06 26 22 21 28 12
16 17 06 26 22 21 28 12
16 17 15 26 22 21 28 12
16 17 15 18 22 21 28 12
16 17 15 18 26 21 28 12
16 17 15 18 26 03 28 12
16 17 15 18 26 03 23 12
16 17 15 18 26 03 23 24
08 17 15 18 26 03 23 24
08 12 15 18 26 03 23 24
08 12 00 18 26 03 23 24
08 12 00 11 26 03 23 24
08 12 00 11 06 03 23 24
08 12 00 11 06 29 23 24
08 12 00 11 06 29 09 24
08 12 00 11 06 29 09 26
27 12 00 11 06 29 09 26
27 13 00 11 06 29 09 26
27 13 14 11 06 29 09 26
27 13 14 03 06 29 09 26
27 13 14 03 18 29 09 26
27 13 14 03 18 17 09 26

27 13 14 03 18 17 23 26

缺页次数:127

缺页率:39.687500%

-----1:OPT 2:LRU 3:FIFO 4:退出-----

请选择一种页面置换算法:4
