1. 在银行家算法中，若出现下述资源分配情况：

|  |
| --- |
| Allocation Need Available |
| P0 0 0 3 2 0 0 1 2 1 6 2 2  P1 1 0 0 0 1 7 5 0  P2 1 3 5 4 2 3 5 6  P3 0 3 3 2 0 6 5 2  P4 0 0 1 4 0 6 5 6 |

试问：（1）该状态是否安全？（2）如果进程P2提出请求Request（1，2，2，2）后，系统能否将资源分配它？（请列出这两个问题的解题过程）

**答**：（1）利用安全性算法对该时刻的资源分配情况进行分析可知（如下图所示），存在一个安全序列｛P0,P3,P1,P2,P4｝,故系统是安全的。

|  |
| --- |
| Work Need Allocation Work+Allocation Finish |
| P0 1 6 2 2 0 0 1 2 0 0 3 2 1 6 5 4 true  P3 1 6 5 4 0 6 5 2 0 3 3 2 1 9 8 6 true  P1 1 9 8 6 1 7 5 0 1 0 0 0 2 9 8 6 true  P2 2 9 8 6 2 3 5 6 1 3 5 4 3 12 13 10 true  P4 3 12 13 10 0 6 5 6 0 0 1 4 3 12 14 14 true |

（安全序列不唯一）

（2）首先，检查Request(1,2,2,2)≤Need(1,2,2)

Request(1,2,2,2)≤Available(1,6,2,2)

其次，系统先假定可为它分配资源，并修改相关数据，由此得到新的系统资源情况，如下图所示：

|  |
| --- |
| Allocation Need Available |
| P0 0 0 3 2 0 0 1 2 0 4 0 0  P1 1 0 0 0 1 7 5 0  P2 2 5 7 6 1 1 3 4  P3 0 3 3 2 0 6 5 2  P4 0 0 1 4 0 6 5 6 |

由于可用资源已不能满足任何进程的需要，故系统进入不安全状态，此时系统不分配资源。

1. 用银行家算法考虑下列系统状态：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 进程 | Allocation | | | | Max | | | | 资源总数 | | | |
| A | 3 | 0 | 1 | 1 | 4 | 1 | 1 | 1 | 6 | 3 | 4 | 2 |
| B | 0 | 1 | 0 | 0 | 0 | 2 | 1 | 2 |  |  |  |  |
| C | 1 | 1 | 1 | 0 | 4 | 2 | 1 | 0 |  |  |  |  |
| D | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |  |  |  |  |
| E | 0 | 0 | 0 | 0 | 2 | 1 | 1 | 0 |  |  |  |  |

请回答：

（1）该时刻系统是否安全？（5分）

（2）若进程B请求（0，0，1，0），可否分配？请分析说明。（5分）

**答**：（1）Need=Max-Allocation

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 1 | 1 | 1 |  | 3 | 0 | 1 | 1 |  | 1 | 1 | 0 | 0 |
| 0 | 2 | 1 | 2 | - | 0 | 1 | 0 | 0 | = | 0 | 1 | 1 | 2 |
| 4 | 2 | 1 | 0 |  | 1 | 1 | 1 | 0 |  | 3 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |  | 1 | 1 | 0 | 1 |  | 0 | 0 | 1 | 0 |
| 2 | 1 | 1 | 0 |  | 0 | 0 | 0 | 0 |  | 2 | 1 | 1 | 0 |

Available=（6,3,4,2）-(5,3,2,2)=(1,0,2,0)

Work=Available（1,0,2,0）

系统安全性分析：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Work | | | | Need | | | | Work+Allocation | | | | Finish |
| D | 1 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 2 | 1 | True |
| A | 2 | 1 | 2 | 1 | 1 | 1 | 0 | 0 | 5 | 1 | 3 | 2 | True |
| B | 5 | 1 | 3 | 2 | 0 | 1 | 1 | 2 | 5 | 2 | 3 | 2 | True |
| C | 5 | 2 | 3 | 2 | 3 | 1 | 0 | 0 | 6 | 3 | 4 | 2 | True |
| E | 6 | 3 | 4 | 2 | 2 | 1 | 1 | 0 | 6 | 3 | 4 | 2 | True |

因为存在一个安全序列, D,A,B,C,E,所以系统处于安全。（序列不唯一）

（2）RequestB（0,0,1,0）<=Need(0,1,1,2)

RequestB（0,0,1,0）<=Available(1,0,2,0)

试着分配B请求的资源，则：

NeedB(0,1,0,2)

AllocationB（0,2,2,2）

Available(1,0,1,0)

安全性算法检测：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | work | | | | need | | | | w+a | | | | finish |
| D | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 2 | 1 | True |
| A | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 5 | 1 | 3 | 2 | True |
| B | 5 | 1 | 3 | 2 | 0 | 1 | 1 | 2 | 5 | 2 | 3 | 2 | True |
| C | 5 | 2 | 3 | 2 | 3 | 1 | 0 | 0 | 6 | 3 | 4 | 2 | True |
| E | 6 | 3 | 4 | 2 | 2 | 1 | 1 | 0 | 6 | 3 | 4 | 2 | True |

因为存在安全序列D，A，B，C，E，所以B请求能满足。

1. 假定系统中有五个进程P1、P2、P3、P4和P5，以及四类资源A、B、C、D。某一时刻的资源分配图如下图所示：

|  |
| --- |
| Allocation Max Available  A B C D A B C D A B C D |
| P1 0 0 1 2 0 0 1 2 2 1 0 0  P2 2 0 0 0 2 7 5 0  P3 0 0 3 4 6 6 5 6  P4 2 3 5 4 4 3 5 6  P5 0 3 3 2 0 6 5 2 |

试问：（1）当前系统是否安全？（2）当进程P3提出请求Request（0，1，0，0）时，系统能否将资源分配给它？（请列出这两个问题的详细解题过程）

答：（1）Need=Max-Allocation 资源分配表如下：

|  |
| --- |
| Allocation Need Available  A B C D A B C D A B C D |
| P1 0 0 1 2 0 0 0 0 2 1 0 0  P2 2 0 0 0 0 7 5 0  P3 0 0 3 4 6 6 2 2  P4 2 3 5 4 2 0 0 2  P5 0 3 3 2 0 3 2 0 |

利用安全性算法对该时刻的资源分配情况进行分析可知（如下图所示），存在一个安全序列｛P1,P4,P5,P2,P3｝,故系统是安全的。

|  |
| --- |
| Work Need Allocation Work+Allocation Finish |
| P1 2 1 0 0 0 0 0 0 0 0 1 2 2 1 1 2 true  P4 2 1 1 2 2 0 0 2 2 3 5 4 4 4 6 6 true  P5 4 4 6 6 0 3 2 0 0 3 3 2 4 7 9 8 true  P2 4 7 9 8 0 7 5 0 2 0 0 0 6 7 9 8 true  P3 6 7 9 8 6 6 2 2 0 0 3 4 6 7 12 12 true |

（安全序列不唯一）

（2）首先，检查Request(0,1,0,0)≤Need(6,6,3,2)

Request(0,1,0,0)≤Available(2,1,0,0)

其次，系统先假定可为它分配资源，并修改相关数据，由此得到新的系统资源情况，如下图所示：

|  |
| --- |
| Allocation Need Available  A B C D A B C D A B C D |
| P1 0 0 1 2 0 0 0 0 2 0 0 0  P2 2 0 0 0 0 7 5 0  P3 0 1 3 4 6 5 2 2  P4 2 3 5 4 2 0 0 2  P5 0 3 3 2 0 3 2 0 |

利用安全性算法对该时刻的资源分配情况进行分析可知（如下图所示）:

|  |
| --- |
| Work Need Allocation Work+Allocation Finish |
| P1 2000 0000 0012 2012 true  P4 2012 2002 2354 4366 true  P5 4366 0320 0332 4698 true  4698 |

此时资源已不能满足P2或P3的请求，系统进入不安全状态，因此系统拒绝此次资源申请。

（查找过程不唯一）

1. 假定系统中有5个进程P0、P1、P2、P3、P4和4种资源A、B、C、D，假设出现下表所示的进程资源分配情况，请回答下列问题。

|  |
| --- |
| 已分配到的资源 尚需资源数 当前可用资源数 |
| P0 1，1，1，0 0，3，3，1 0，3，2，2  P1 0，2，3，1 0，3，4，2  P2 0，2，1，2 1，0，3，4  P3 0，3，1，0 0，3，2，0  P4 1，0，2，1 0，4，2，3 |

（1）该状态是否安全？为什么？

（2）如果进程P0提出资源请求Request（0，0，0，1），系统能否将资源分配它？为什么？（请列出这两个问题的解题过程）

答：（1）

|  |
| --- |
| Work Need Allocation Work+Allocation Finish |
| P3 0，3，2，2 0，3，2，0 0，3，1，0 0，6，3，2 TRUE  P0 0，6，3，2 0，3，3，1 1，1，1，0 1，7，4，2 TRUE  P1 1，7，4，2 0，3，4，2 0，2，3，1 1，9，7，3 TRUE  P4 1，9，7，3 0，4，2，3 1，0，2，1 2，9，9，4 TRUE  P2 2，9，9，4 1，0，3，4 0，2，1，2 2，11，10，6 TRUE |

利用安全性算法对该时刻的资源分配情况进行分析可知，存在一个安全序列P3,P0,P1,P4,P2，因此，该状态是安全的。

（2）由于Request(0，0，0，1)≤Need(0，3，3，1)

Request(0，0，0，1)≤Available(0，3，2，2)

固可尝试将资源分配给P0，并修改相关数据。P0对应的Need为（0，3，3，0），Allocation为（1，1，1，1）；系统对应的Available为（0，3，2，1）。由此得到新的系统资源情况：

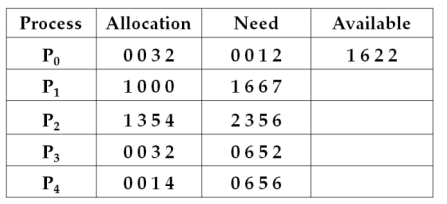
|  |
| --- |
| Work Need Allocation Work+Allocation Finish |
| P3 0，3，2，1 0，3，2，0 0，3，1，0 0，6，3，1 TRUE  P0 0，6，3，1 0，3，3，0 1，1，1，1 1，7，4，2 TRUE  P1 1，7，4，2 0，3，4，2 0，2，3，1 1，9，7，3 TRUE  P4 1，9，7，3 0，4，2，3 1，0，2，1 2，9，9，4 TRUE  P2 2，9，9，4 1，0，3，4 0，2，1，2 2，11，10，6 TRUE |

存在一个安全序列P3,P0,P1,P4,P2，由于该状态是安全的，因此可实施资源分配。

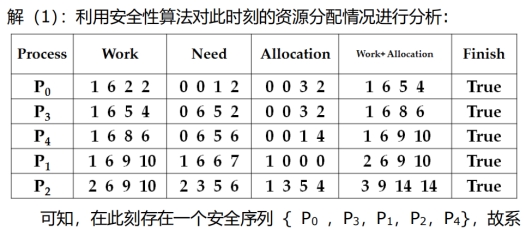
5. 系统状态如图表所示，试用银行家算法解决：

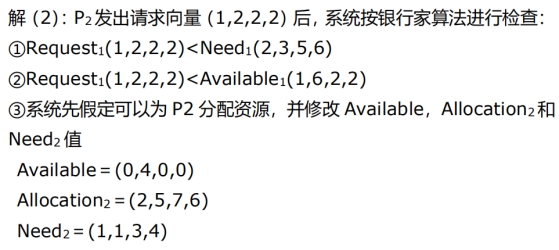
（1）该状态是否安全?

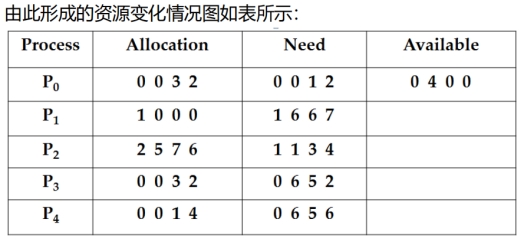
（2）若进程P2提出请求Request(1,2,2,2)后，系统能否将资源分配给它?

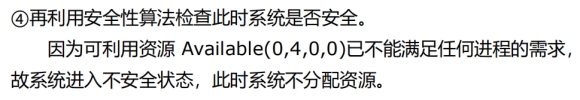


答：







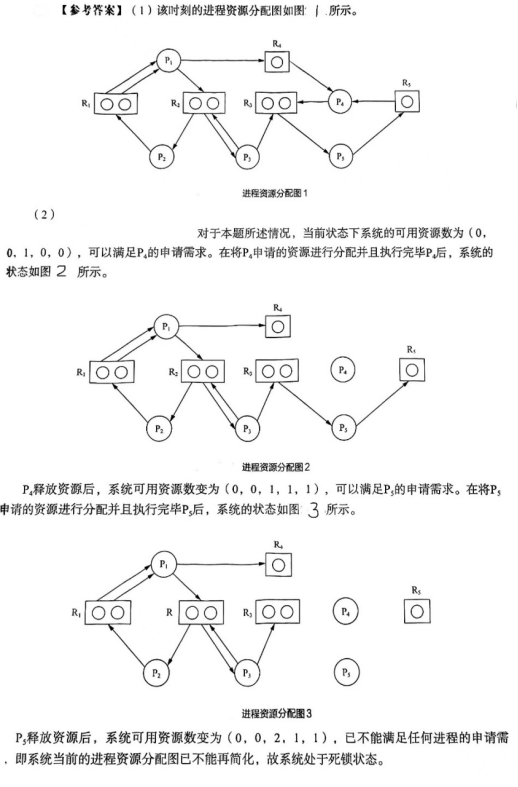


6. 假设系统有5类独占资源：R1、R2、R3、R4、R5。各类资源分别有2、2、2、1、1个。系统有5个进程：P1、P2、P3、P4、P5。其中P1已占有2个R1，且申请1个R2和1个R4；P2已占有1个R2，且申请1个R1；P3已占有1个R2，且申请1个R2和1个R3；P4已占有1个R4和1个R5，且申请1个R3；P5已占有1个R3，且申请1个R5。

（1）试画出该时刻的资源分配图。

（2）判断（1）中的资源分配图有无死锁，给出判断过程和结果。

答：



7. 在一个请求分页系统中，假定一个作业的运行页面走向为：1，2，3，6，4，7，3，2，1，4，7，5，6，5，2，1。当分配给此作业的物理块数为4时，请用最近最久未使用置换算法（LRU置换算法）画出页面置换图，并计算访问过程中的页面置换次数和缺页率。

答：

LRU：页面置换次数10，缺页中断次数为14；缺页率为14/16=87.5%

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 6 | 4 | 7 | 3 | 2 | 1 | 4 | 7 | 5 | 6 | 5 | 2 | 1 |
| 1 | 1 | 1 | 1 | 4 | 4 |  | 4 | 1 | 1 | 1 | 1 | 6 |  | 6 | 6 |
|  | 2 | 2 | 2 | 2 | 7 |  | 7 | 7 | 4 | 4 | 4 | 4 |  | 2 | 2 |
|  |  | 3 | 3 | 3 | 3 |  | 3 | 3 | 3 | 7 | 7 | 7 |  | 7 | 1 |
|  |  |  | 6 | 6 | 6 |  | 2 | 2 | 2 | 2 | 5 | 5 |  | 5 | 5 |

8. 在一个采用页式虚拟存储管理的系统中，有一用户作业，它依次要访问的字地址序列是：115，228，120，88，446，102，321，432，260，167，若该作业的第0页已经装入主存，现分配给该作业的主存共300字，页的大小为100字，请回答下列问题：

（1）按FIFO调度算法将产生多少次缺页中断，依次淘汰的页号为多少，缺页中断率为多少？

（2）按LRU调度算法将产生多少次缺页中断，依次淘汰的页号为多少，缺页中断率为多少？

答：页号=地址/页的大小，访问页面序列为：1,2,1,0,4,1,3,4,2,1

FIFO：发生缺页的次数5次，依次淘汰的页号为0,1,2，缺页率为5/10=50%

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 1 | 0 | 4 | 1 | 3 | 4 | 2 | 1 |
| 0 | 0 |  |  | 4 |  | 4 |  |  | 4 |
| 1 | 1 |  |  | 1 |  | 3 |  |  | 3 |
|  | 2 |  |  | 2 |  | 2 |  |  | 1 |

LRU：发生缺页的次数6次，依次淘汰的页号为2,0,1,3，缺页率为6/10=60%

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 1 | 0 | 4 | 1 | 3 | 4 | 2 | 1 |
| 0 | 0 |  |  | 0 |  | 3 |  | 3 | 1 |
| 1 | 1 |  |  | 1 |  | 1 |  | 2 | 2 |
|  | 2 |  |  | 4 |  | 4 |  | 4 | 4 |

9. 一个页式存储管理系统使用FIFO 、OPT 和LRU 页面替换算法，如果一个作业的页面走向为：1、8、1、7、8、2、7、2、1、8、3、8、2、1、3、1、7、1、3、7。当分配给该作业的物理块数为3时，试分别计算访问过程中发生的缺页中断次数和缺页中断率，并说明哪个算法更适合？

**答**：

Fifo：11次，11/20=55%

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 8 | 1 | 7 | 8 | 2 | 7 | 2 | 1 | 8 | 3 | 8 | 2 | 1 | 3 | 1 | 7 | 1 | 3 | 7 |
| 1 | 1 |  | 1 |  | 2 |  |  | 2 | 2 | 3 |  | 3 | 3 |  |  | 7 |  | 7 |  |
|  | 8 |  | 8 |  | 8 |  |  | 1 | 1 | 1 |  | 2 | 2 |  |  | 2 |  | 3 |  |
|  |  |  | 7 |  | 7 |  |  | 7 | 8 | 8 |  | 8 | 1 |  |  | 1 |  | 1 |  |

Lru：11次，11/20=55%

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 8 | 1 | 7 | 8 | 2 | 7 | 2 | 1 | 8 | 3 | 8 | 2 | 1 | 3 | 1 | 7 | 1 | 3 | 7 |
| 1 | 1 |  | 1 |  | 2 |  |  | 2 | 2 | 3 |  | 3 | 1 | 1 |  | 1 |  |  |  |
|  | 8 |  | 8 |  | 8 |  |  | 1 | 1 | 1 |  | 2 | 2 | 2 |  | 7 |  |  |  |
|  |  |  | 7 |  | 7 |  |  | 7 | 8 | 8 |  | 8 | 8 | 3 |  | 3 |  |  |  |

Opt：8次，8/20=40%

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 8 | 1 | 7 | 8 | 2 | 7 | 2 | 1 | 8 | 3 | 8 | 2 | 1 | 3 | 1 | 7 | 1 | 3 | 7 |
| 1 | 1 |  | 1 |  | 1 |  |  |  | 1 | 3 |  |  | 3 |  |  | 3 |  |  |  |
|  | 8 |  | 8 |  | 2 |  |  |  | 2 | 8 |  |  | 1 |  |  | 1 |  |  |  |
|  |  |  | 7 |  | 7 |  |  |  | 8 | 2 |  |  | 2 |  |  | 7 |  |  |  |

OPT算法最好，但不易实现，就本题而言LRU和FIFO一样，但实际中LRU更好。

10. 一个请求磁盘I/O的磁盘调度队列，分别在下列磁道上阻塞：40，90，170，38，110，20，144，48，59。磁头初始在100号磁道上，沿磁道号增加方向移动。

（1）按照扫描算法（SCAN算法）计算平均寻道长度。（5分）

（2）按照先来先服务算法（FCFS算法）计算平均寻道长度。（5分）

【参考答案】

（1）SCAN算法：

|  |  |  |
| --- | --- | --- |
| 到达序列 | 访问序列 | 移动距离 |
| 40 | 110 | 10 |
| 90 | 144 | 34 |
| 170 | 170 | 26 |
| 38 | 90 | 80 |
| 110 | 59 | 31 |
| 20 | 48 | 11 |
| 144 | 40 | 8 |
| 48 | 38 | 2 |
| 59 | 20 | 18 |

平均寻道距离=（10+34+26+80+31+11+8+2+18）/9=24.4

（2）FCFS算法：

|  |  |
| --- | --- |
| 100 | FCFS |
| 40 | 60 |
| 90 | 50 |
| 170 | 80 |
| 38 | 132 |
| 110 | 72 |
| 20 | 90 |
| 144 | 124 |
| 48 | 96 |
| 59 | 11 |

平均寻道距离=（60+50+80+132+72+90+124+96+11）/9=79.44

11. 一个页式存储管理系统使用FIFO、OPT和LRU页面置换算法，如果一个作业的页面走向为：4、3、2、l、4、3、5、4、3、2、l、5。当分配给该作业的物理块数为4时，试分别计算访问过程中发生的缺页中断次数和缺页中断率，并说明哪个算法更适合？

答：

（1）FIFO，缺页中断率为10 / 12 =83.3 ％

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 3 | 2 | 1 | 4 | 3 | 5 | 4 | 3 | 2 | 1 | 5 |
| 4 | 4 | 4 | 4 |  |  | 5 | 5 | 5 | 5 | 1 | 1 |
|  | 3 | 3 | 3 |  |  | 3 | 4 | 4 | 4 | 4 | 5 |
|  |  | 2 | 2 |  |  | 2 | 2 | 3 | 3 | 3 | 3 |
|  |  |  | 1 |  |  | 1 | 1 | 1 | 2 | 2 | 2 |

（2）LRU 为8 次，缺页中断率为8 / 12 = 67.7% 。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 3 | 2 | 1 | 4 | 3 | 5 | 4 | 3 | 2 | 1 | 5 |
| 4 | 4 | 4 | 4 |  |  | 4 |  |  | 4 | 4 | 5 |
|  | 3 | 3 | 3 |  |  | 3 |  |  | 3 | 3 | 3 |
|  |  | 2 | 2 |  |  | 5 |  |  | 5 | 1 | 1 |
|  |  |  | 1 |  |  | 1 |  |  | 2 | 2 | 2 |

（3）OPT 为6 次，缺页中断率为6/ 12 = 50% 。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 3 | 2 | 1 | 4 | 3 | 5 | 4 | 3 | 2 | 1 | 5 |
| 4 | 4 | 4 | 4 |  |  | 4 |  |  |  | 4 |  |
|  | 3 | 3 | 3 |  |  | 3 |  |  |  | 3 |  |
|  |  | 2 | 2 |  |  | 2 |  |  |  | 1 |  |
|  |  |  | 1 |  |  | 5 |  |  |  | 5 |  |

经过比较：OPT最好，但不实际，LRU实现效果最好。

12. 一个页式存储管理系统使用FIFO、OPT和LRU页面替换算法，如果一个作业的页面走向为：2、3、2、l、5、2、4、5、3、2、5、2。当分配给该作业的物理块数为3时，试分别计算访问过程中发生的缺页次数和缺页率，并说明哪个算法更适合？

答：

FIFO：9次，缺页率=9/12=75%

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 3 | 2 | l | 5 | 2 | 4 | 5 | 3 | 2 | 5 | 2 |
| 2 | 2 |  | 2 | 5 | 5 | 5 |  | 3 |  | 3 | 3 |
|  | 3 |  | 3 | 3 | 2 | 2 |  | 2 |  | 5 | 5 |
|  |  |  | 1 | 1 | 1 | 4 |  | 4 |  | 4 | 2 |
| LRU：7次，7/12=58.3% | | | | | | | | | | | |
| 2 | 3 | 2 | l | 5 | 2 | 4 | 5 | 3 | 2 | 5 | 2 |
| 2 | 2 |  | 2 | 2 |  | 2 |  | 3 | 3 |  |  |
|  | 3 |  | 3 | 5 |  | 5 |  | 5 | 5 |  |  |
|  |  |  | 1 | 1 |  | 4 |  | 4 | 2 |  |  |
| OPT：6次，6/12=50% | | | | | | | | | | | |
| 2 | 3 | 2 | l | 5 | 2 | 4 | 5 | 3 | 2 | 5 | 2 |
| 2 | 2 |  | 2 | 2 |  | 4 |  |  | 2 |  |  |
|  | 3 |  | 3 | 3 |  | 3 |  |  | 3 |  |  |
|  |  |  | 1 | 5 |  | 5 |  |  | 5 |  |  |

OPT最好，但不实际，LRU实现效果最好。

13. 请求页式存储管理中，进程访问地址序序列为：10, 11, 104, 170, 73, 305, 180, 240, 235, 485, 467, 366。试问：

（1）如果页面大小为100，给出页面访问序列。

（2）进程分3个页框，采用FIFO算法，求缺页率。（注：初始页面为空，第一次页面调入认为发生一次缺页）

（3）进程分3个页框，采用LRU替换算法，求缺页率。（注：初始页面为空，第一次页面调入认为发生一次缺页）

答：

（1）页面访问序列为0 , 0 , 1 , 1 , 0 , 3 , 1 , 2 ,2 , 4 , 4 , 3

（2）FIFO ：页面调入 5 次，缺页率为5 / 12=41.7%。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 0 | 3 | 1 | 2 | 2 | 4 | 4 | 3 |
| 0 |  | 0 |  |  | 0 |  | 2 |  | 2 |  |  |
|  |  | 1 |  |  | 1 |  | 1 |  | 4 |  |  |
|  |  |  |  |  | 3 |  | 3 |  | 3 |  |  |

（3）LRU ：页面调入6 次，缺页率为6 / 12 = 50% 。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 0 | 3 | 1 | 2 | 2 | 4 | 4 | 3 |
| 0 |  | 0 |  |  | 0 |  | 2 |  | 2 |  | 2 |
|  |  | 1 |  |  | 1 |  | 1 |  | 1 |  | 3 |
|  |  |  |  |  | 3 |  | 3 |  | 4 |  | 4 |

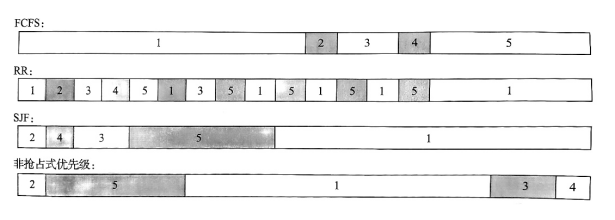
14. 假定要在一台处理机上执行下表所示的作业，且假定这些作业在时刻0以1，2，3，4，5的顺序到达。

（1）请说明分别采用FCFS、RR(时间片为1)、SJF及非抢占式优先级调度算法时，这些作业的执行情况(优先级的高低顺序依次为1到5)。

（2）针对上述每种调度算法，给出平均周转时间和平均带权周转时间。

|  |  |  |
| --- | --- | --- |
| 作业 | 执行时间 | 优先级 |
| 1 | 10 | 3 |
| 2 | 1 | 1 |
| 3 | 2 | 3 |
| 4 | 1 | 4 |
| 5 | 5 | 2 |

答：（1）作业执行顺序如图所示：



（2）各个作业对应于各个算法的周转时间和带权周转时间如下表：



15. 某虚拟存储器的用户空间共有32个页面，每页1KB，内存16KB。假定某时刻系统为用户的第0、1、2、3页分配的物理块号分别为5、10、4、7，而该用户作业的长度为6页，试将十六进制逻辑地址0A5C、103C、1A5C转换成物理地址。

答：由题目所给条件可知，该系统的逻辑地址有15位，其中高5位为页号，低10位为页内地址；物理地址有14位，其中高4位为块号，低10位为块内地址。另外，由于题目中给出的逻辑地址是十六进制数，故可先将其转换成二进制数以直接获得页号和页内地址，再完成地址的转换。

（1）逻辑地址0A5CH转化为二进制为000 1010 0101 1100，页号为00010，即2号页，页号合法。从页表中找到对应的内存物理块号为4，即0100;与页内地址10 0101 1100拼接而形成物理地址01 0010 0101 1100，即125CH。（4分）

（2）逻辑地址103CH的页号为4，页号合法，但该页未装入内存，故产生缺页中断。

（3）逻辑地址1A5CH的页号为6，页号非法，故产生越界中断。

16. 在某个文件系统中，每个盘块为2K字节，文件控制块占64个字节，其中文件名占14个字节。如果索引节点编号占2个字节，对一个存放在磁盘上的1024个目录项的目录，试比较引入索引结点前后，为找到其中一个文件的FCB，平均启动磁盘的次数。

答：

在引入索引结点前，每个目录项中存放的是对应文件的FCB，故1024个目录项的目录总共需要占用1024\*64/2048=32个盘块。因此，在该目录中检索到一个文件，平均启动磁盘的次数为(1+32)/2=16.5次。

在引入索引结点后，目录项中只需存储文件名和对应文件索引节点编号，而该文件的其他信息存储在索引节点中，因此目录占用盘块数：1024\*(14+2)/2048=8块。

按名查找到目录中的索引节点编号平均启动磁盘次数：(1+8)/2=4.5次

得到索引结点编号后，要将对应文件的索引结点内信息读入内存，需启动磁盘1次，故平均需要启动磁盘4.5+1=5.5次。

17. 假定磁盘的盘块大小为1KB（1K=1024），若采用FAT文件系统进行管理，FAT表项大小要求为半字节的整数倍，且FAT占用的空间要求尽可能小，请回答以下问题：（1）当硬盘容量为800MB时，其FAT占用多少存储空间？

（2）当硬盘容量为2GB时，其FAT占用多少存储空间？

【参考答案】（1）800M/1K=800K个盘块=800╳210个盘块，近似等于1024╳210个=210╳210个=220个。因此，用20位二进制位表示盘块号。由于1个字节是8位二进制位，因此半个字节是4位二进制位，20正好是4的整数倍。用20/8=2.5个字节表示一个盘块，即每个FAT表项的大小为2.5B，因此FAT占用的存储空间为25╳800╳210=2000KB。

（2）2GB=231，231/210=221个盘块。21不是4的整数倍，因此，用24位二进制位表示盘块，即FAT的表项占24/8=3B。因此FAT占用的存储空间为221╳3=6MB。

18. 在某个文件系统中，每个盘块为1KB，文件控制块占64个字节，其中文件名占8个字节。如果索引节点编号占2个字节，对一个存放在磁盘上的256个目录项的目录，试比较引入索引结点前后，为找到其中一个文件的FCB，平均启动磁盘的次数。

答：在引入索引结点前，每个目录项中存放的是对应文件的FCB，故256个目录项的目录总共需要占用256\*64/1024=16个盘块。因此，在该目录中检索到一个文件，平均启动磁盘的次数为(1+16)/2=8.5次。

在引入索引结点后，目录项只存储文件名和索引节点编号，而该文件的其他信息存储在索引节点中，因此目录占用盘块数：256\*(8+2)/1024=2.5

按名查找到目录中的索引节点编号平均启动磁盘次数：(1\*1/2.5+2\*1/2.5+3\*0.5/2.5)=1.8，说明：文件在第一块的概率是1/2.5，需1次磁盘读取，文件在第二块的概率是1/2.5，需2次磁盘读取，文件在第三块的概率是0.5/2.5，需3次磁盘读取，按离散型概率来算。

得到索引结点编号后，要将对应文件的索引结点内信息读入内存，需启动磁盘1次，故平均需要启动磁盘1.8+1=2.8次。

19. 某文件系统采用混合索引方式组织文件的存储空间，其盘块大小是4KB，每个目录项中包括13个地址项，其中0~9是直接地址，10为一级间址，11为二级间址，12为三级间址。已知1个地址占4B，现有文件A、B、C，它们的大小分别为5KB、40.5KB、4138KB，若不计目录项，则请问这些文件分别占用多大的磁盘空间？

答：

盘块的大小为4KB，地址的大小为4B，每个盘块可存储4KB/4B=1024个地址。最大的文件长度为(10+1 024+10 242+10243) ╳4KB=40KB+4MB+4GB+4TB。长度为5K、40.5KB、4138KB的文件A、B、C所占的磁盘空间如下。

（1）A文件：因为每个物理块的大小是4KB，所以5KB需要2个物理块，使用2个直接地址，所需磁盘空间为2╳4KB=8KB。

（2）B文件：40.5KB/4KB=10.125，需要11个盘块，因此要用到10个直接地址和1个一级间址索引块。一级间址索引块是单独的地址，因此所需磁盘空间为40.5KB+4KB=44.5KB。

（3）C文件：4138KB/4KB=1035，而(1035-10)/1024=2，故需要2个二级间址索引块。由于索引块需要(1+2) ╳4KB=12KB，因此所需磁盘空间为4 138KB+12KB=4150KB。

20. 某文件系统采用索引节点存放文件的属性和地址信息，簇的大小为4KB。每个文件索引节点占64B，有11个地址项，其中直接地址项8个，一级、二级和三级间址项各1个，每个地址项的长度均为4B。请回答下列问题。

（1）该文件系统能支持的最大文件长度是多少（给出计算表达式即可）？

（2）若文件F1的大小为6KB，文件F2的大小为40KB，则该文件系统获取F1和F2最后一个簇的簇号需要的时间是否相同?为什么?

答：

（1）由题意可知,该文件系统在外存中存放的基本分配单位是簇,大小为4K。

直接地址:指向的是文件数据簇,因此支持的文件长度为: 8x4K= 32K ;

一级间址:簇作为索引块,每个地址项长度为4B ,则每个簇可存放的地址项个数为: 4KB/4B=1K个数据簇，因此支持文件长度为: 1Kx4K=4M ;

二级间址:簇作为索引块,每个簇可存放的依旧是索引块,则存储1Kx1K=1M个数据簇;因此支持文件长度为: 1Mx4K=4G ;

三级间址:存储的数据簇个数为1Kx1Kx1K=1G个,支持文件长度为: 1Gx4K=4T ;

由此，该文件系统能支持的最大文件长度为32KB+4MB+4GB+4TB.

（2）若文件F1的大小为6KB，文件F2的大小为40KB，则该文件系统获取F1和F2最后一个簇的簇号需要的时间不相同;

因为F1有6KB，它的最后一个簇的簇号可从索引项的直接地址项中得到。因为，直接地址支持的文件长度为32K;

而F2有40KB，要获得其最后一个簇的簇号就要访问一级间接地址索引表。

直接地址可以从索引指针直接得到其所在的磁盘块地址，故访盘1次即可。一级间址，先访盘1次得到文件所在的磁盘块地址，再访盘1次读出内容，因此共须访盘2次。