**操作系统2021课后应用题作业1**

**姓名：\_\_\_\_刘晓旭\_\_\_\_ 学号：\_\_\_\_\_\_\_201250123\_\_\_\_\_\_\_\_**

**提醒：直接在本文档填写解题答案（不要另建word文件），  
提交作业的文件名命名规范为【学号\_姓名\_作业1.doc】**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **题序** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **总分** |
| **分值** | **满分**10分 | **满分**10分 | **满分**10分 | **满分**10分 | **满分10**分 | **满分10**分 | **满分**8分 | **满分**8分 | **满分**24分 | **满分**100 |
| **分值** |  |  |  |  |  |  |  |  |  |  |

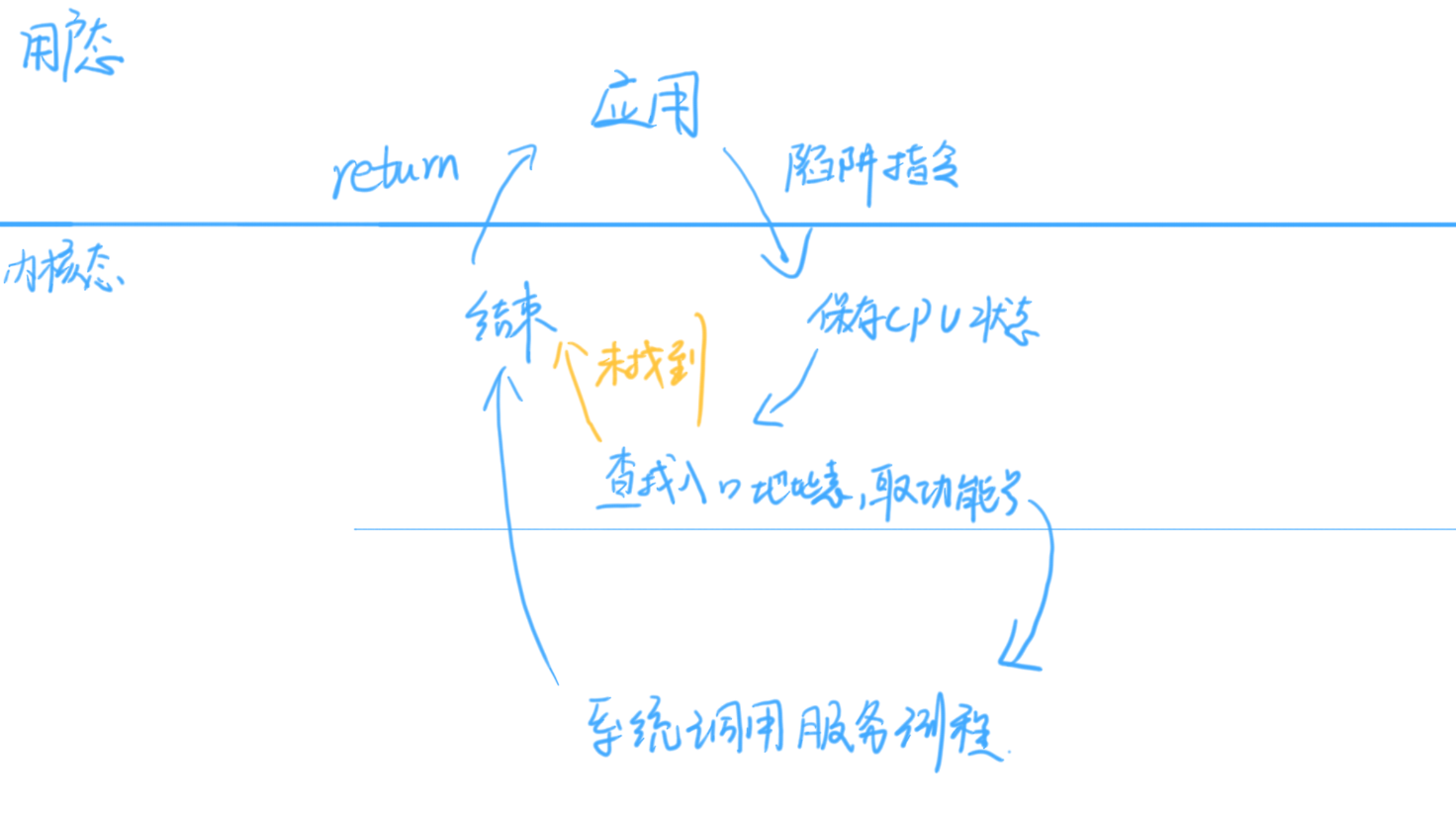
1. 试述系统调用的实现原理，陷阱机制和绘制系统调用的处理过程，并阐述系统调用的处理逻辑。（满分10分）

答：计算机有两种运行模式：用户态与内核态。其中应用程序运行在用户态，操作系统运行在内核态。在用户态中，软件只能使用少数指令，它们不具备直接访问硬件的权限；在内核态中，操作系统拥有对所有硬件的完全的访问权限，可以使机器运行任何指令。

系统调用的实现原理：操作系统的系统调用的机制称为系统陷阱或系统异常处理机制。应用程序产生陷阱指令，能够使处理器由用户态转换到内核态。陷阱指令应该指明系统调用的功能号和传递的参数。有了功能号和参数后，需要设计系统调用服务例程和系统调用入口地址表，每个入口和例程一一对应。找到例程后，执行系统服务例程，结束处理，并返回。

陷阱机制：在陷阱机制中，需要开辟现场保护区来保存系统调用时的处理器状态。便于在结束处理的时候返回原来的状态。

系统调用的处理过程：



系统调用的处理逻辑：实际上是函数调用，但是调用的是处于内核态的系统函数。用户给内核传递一个系统调用号，系统找到内核函数执行，最后返回。

1. 某个计算机系统有一台输入机和一台打印机，现有两道程序投入运行，且程序A先开始运行，程序B后开始运行。程序A的运行轨迹为：计算50ms、打印100ms、再计算50ms、打印100ms，结束。程序B的运行轨迹为：计算50ms、输入80ms、再计算100ms，结束。试说明：  
    ①两道程序运行时，CPU是否存在空闲等待？若是，在哪段时间内等待？为什么等待？  
    ②程序A、B是否有等待CPU的情况？若有，指出发生等待的时刻。（满分10分）

答：

①：以下为处理器、输入机、打印机的执行时间流程：

处理器：

0-50ms A执行计算；

50-100ms A要打印、B执行计算；

100-150ms A没打印完、B没输入完、空闲；

150-200ms A执行计算；

200-300ms B执行计算。

输入机：

0-100ms 空闲

100-180ms B输入

180-300ms 空闲

打印机：

0-50ms 空闲

50-150ms A打印

150-200ms 空闲

200-300ms A打印

CPU存在空闲等待，在100-150ms之间，因为A没打印完成、B没输入完成。

②：A不等待，线性进行；B在0-50ms等待A计算完毕，在180-200ms等待A计算完毕。

1. 若内存中有3道程序A、B、C，按照A、B、C的优先次序运行。各程序的计算轨迹为：

A：计算（20ms），I/O（30ms），计算（10ms）

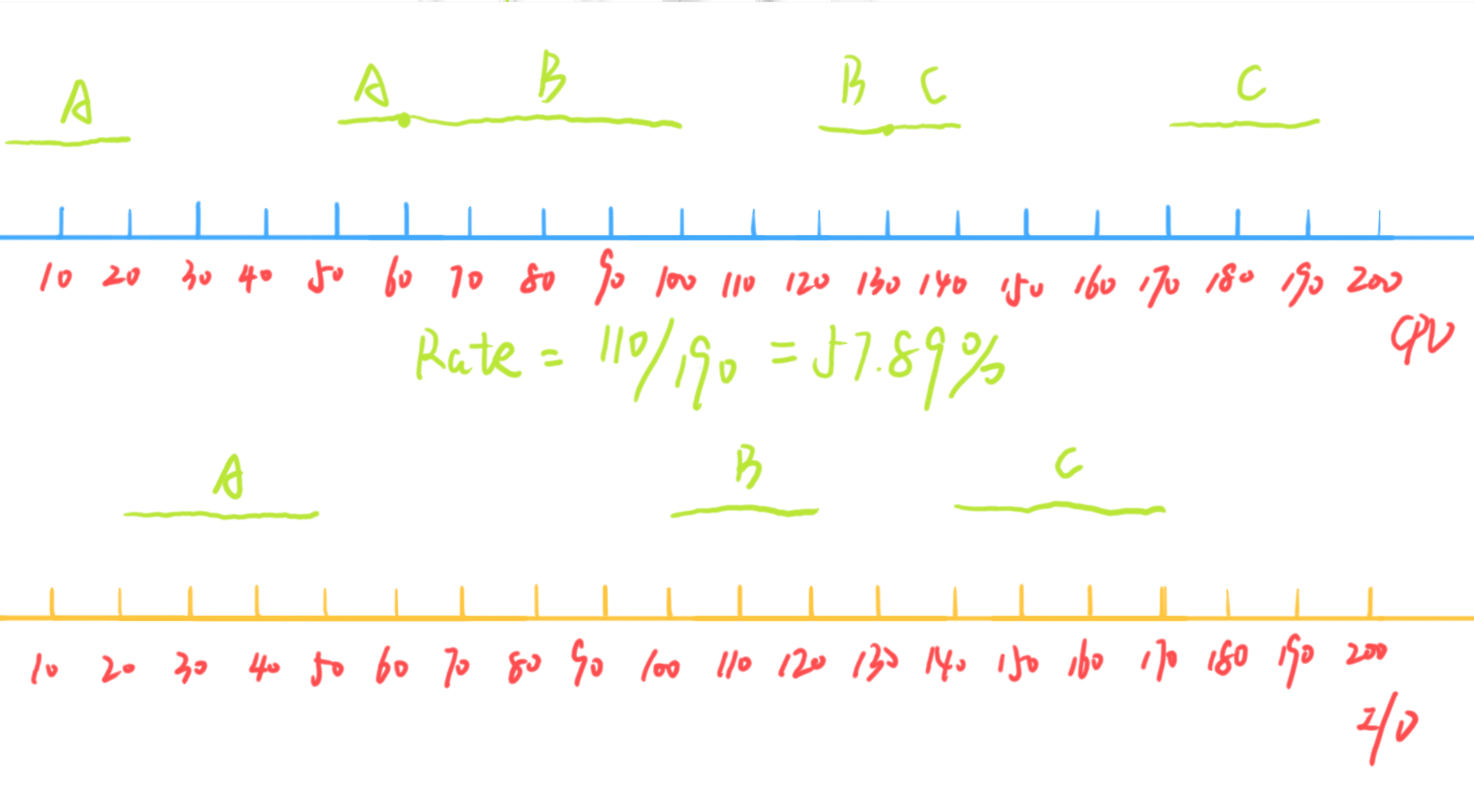
B：计算（40ms），I/O（20ms），计算（10ms）

C：计算（10ms），I/O（30ms），计算（20ms）

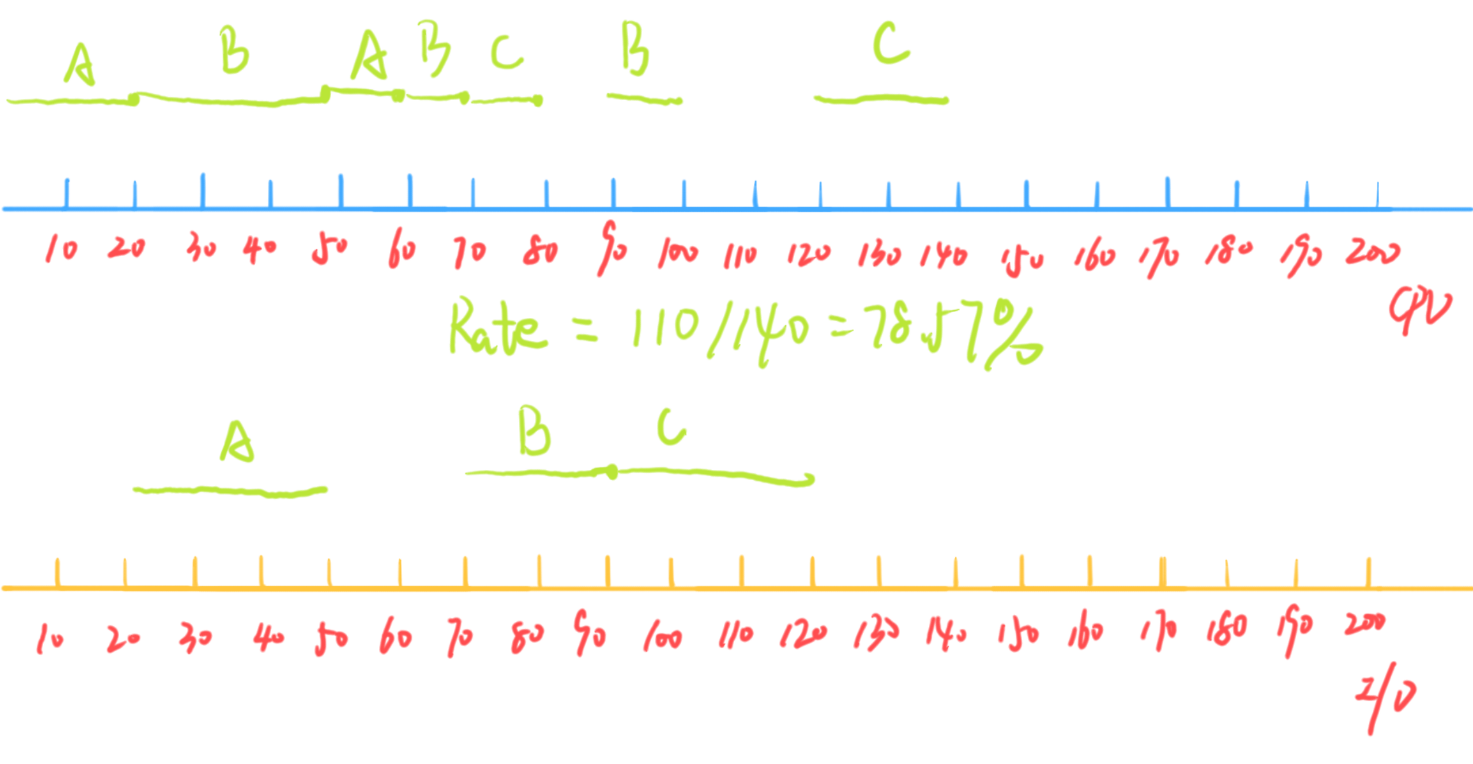
如果3道程序都使用相同的设备进行I/O操作（即程序以串行方式使用设备，调度开销忽略不计），试分别画出①单道和②多道运行的时间关系图。在两种情况下，CPU的平均利用率各是多少？（满分12分）

答：

单道&利用率：



多道&利用率：

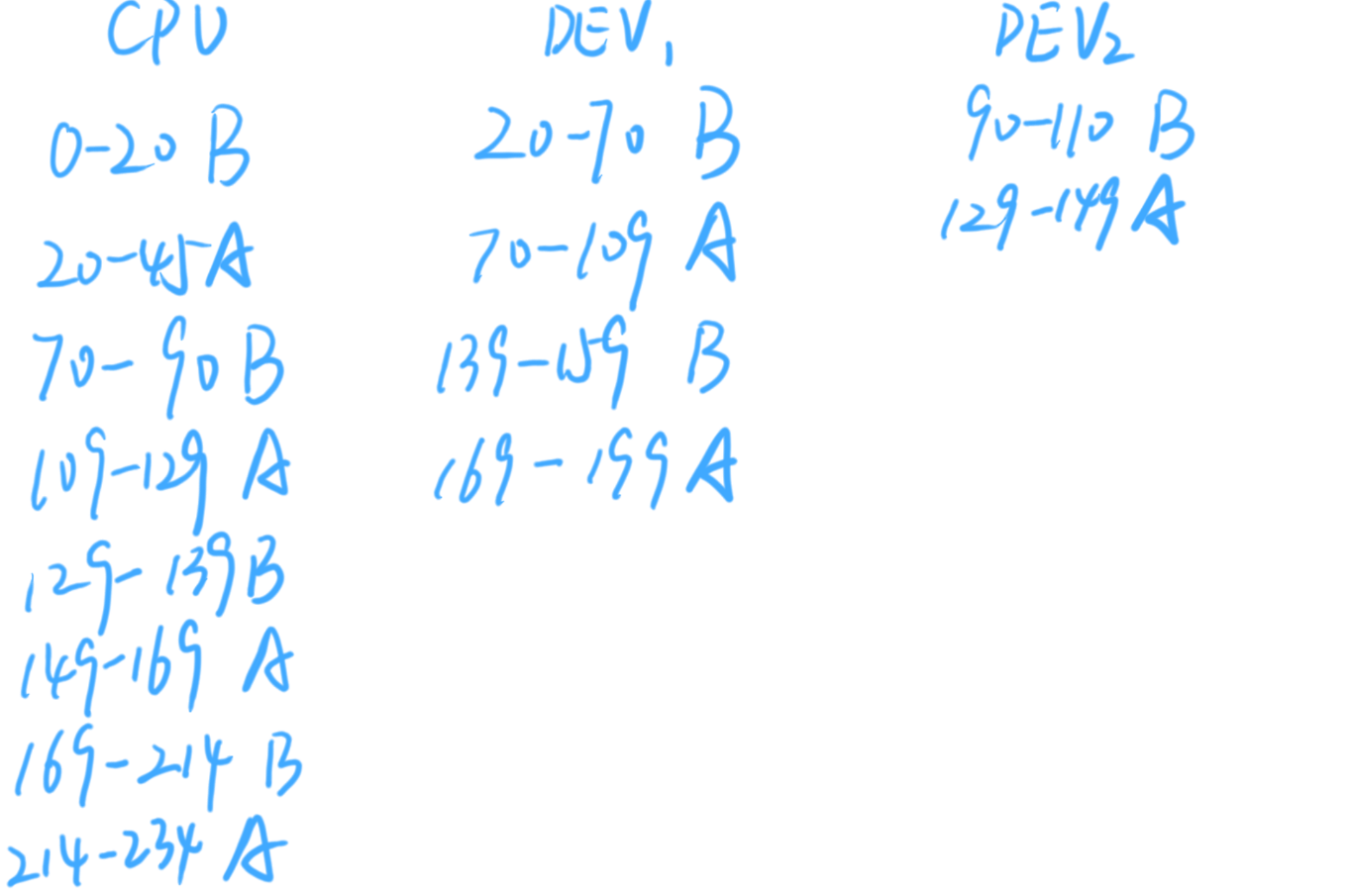


1. 在单机系统中，有CPU和两个设备DEV1、DEV2，它们能够同时工作。现有两个程序A、B同时到达，程序B的优先级高于程序A，但当程序A占用CPU时，程序B不能抢占。程序在CPU与IO设备之间的切换开销忽略不计。如果这两个程序使用CPU、DEV1、DEV2的顺序和时间如下表所示。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 程序 | 运行情况/ms | | | | | | |
| A | CPU | DEV1 | CPU | DEV2 | CPU | DEV1 | CPU |
| 25 | 39 | 20 | 20 | 20 | 30 | 20 |
| B | CPU | DEV1 | CPU | DEV2 | CPU | DEV1 | CPU |
| 20 | 50 | 20 | 20 | 10 | 20 | 45 |

试解答下列问题：  
 ①哪个程序先结束？  
 ②程序全部执行结束需要多长时间？  
 ③程序全部执行完毕时，CPU的利用率是多少？  
 ④程序A等待CPU的累计时间是多少？  
 ⑤程序B等待CPU的累计时间是多少？（满分10分）

答：示意图如下：



①：B先结束。

②：234ms。

③：180/234=76.92%。

④：0-20 199-214 共35ms

⑤：110-129 159-169 共25ms

1. 在一个只支持四道程序同时运行的多道程序系统中，若在一段时间内先后到达6个作业，其提交时刻和估计运行时间由下表给出。

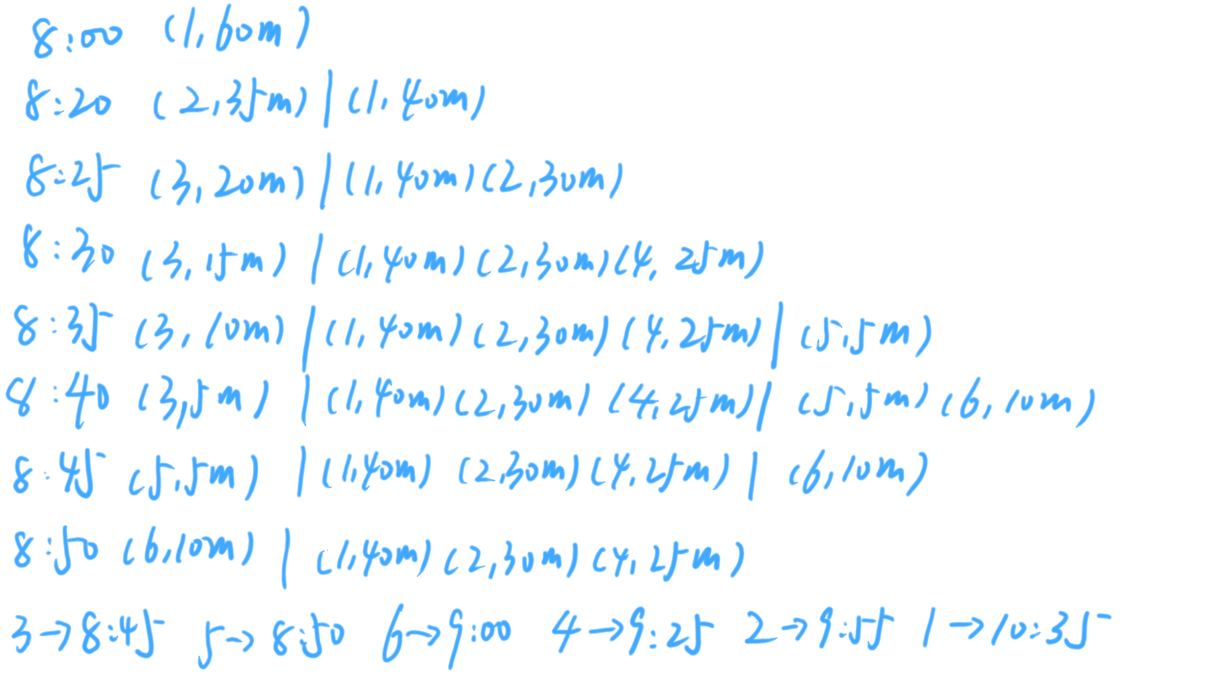
|  |  |  |
| --- | --- | --- |
| 作业 | 提交时刻 | 估计运行时间/min |
| 1 | 8:00 | 60 |
| 2 | 8:20 | 35 |
| 3 | 8:25 | 20 |
| 4 | 8:30 | 25 |
| 5 | 8:35 | 5 |
| 6 | 8:40 | 10 |

系统采用SRTF调度算法，作业被调度进入系统后中途不会退出，但作业运行时可被剩余时间更短的作业所抢占。

①分别给出6个作业的开始执行时间、作业完成时间、作业周转时间。

②计算平均作业周转时间。（满分10分）

答：作业示意图如下：



①：

作业1： 8：00开始运行 10：35完成 周转时间155分钟

作业2： 8：20开始运行 9：55完成 周转时间95分钟

作业3： 8：25开始运行 8：45完成 周转时间20分钟

作业4： 9：00开始运行 9：25完成 周转时间55分钟

作业5： 8：45开始运行 8：50完成 周转时间15分钟

作业6： 8：50开始运行 9：00完成 周转时间20分钟

②：（155+95+20+55+15+20）/6=60分钟

1. 设有4个进程P1、P2、P3、P4，它们到达就绪队列的时刻、运行时间及优先级（优先数越大优先级越高）如下表：

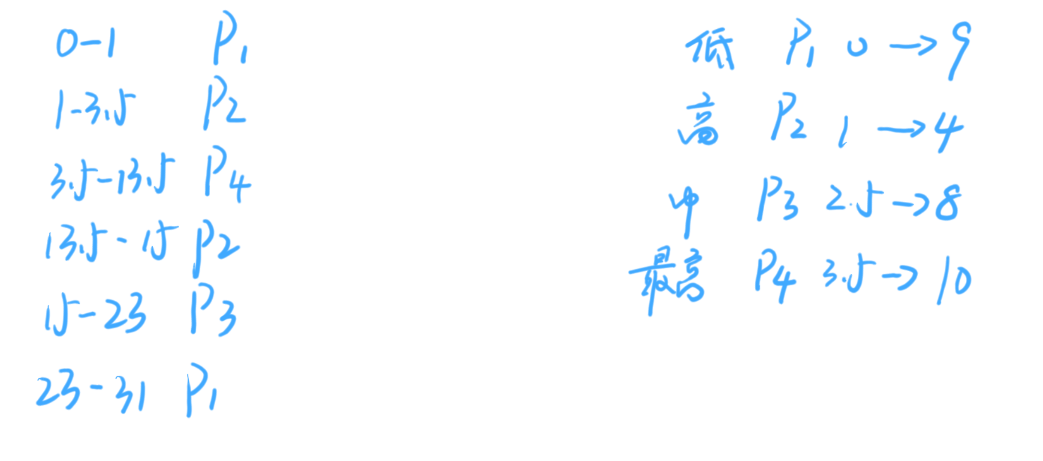
|  |  |  |  |
| --- | --- | --- | --- |
| 进程 | 到达就绪队列的时刻 | 运行时间/ms | 优先级 |
| P1 | 0 | 9 | 1 |
| P2 | 1 | 4 | 3 |
| P3 | 2.5 | 8 | 2 |
| P4 | 3.5 | 10 | 4 |

①若采用抢占式优先数调度算法（抢占的时间点为高优先级进程达到就绪队列的时刻），试给出各个进程的调度次序以及进程的平均周转时间和平均等待时间。

②若采用时间片轮换调度算法，且时间片长度取2ms，试给出各个进程的调度次序以及进程的平均周转时间和平均等待时间。（满分10分）

答：

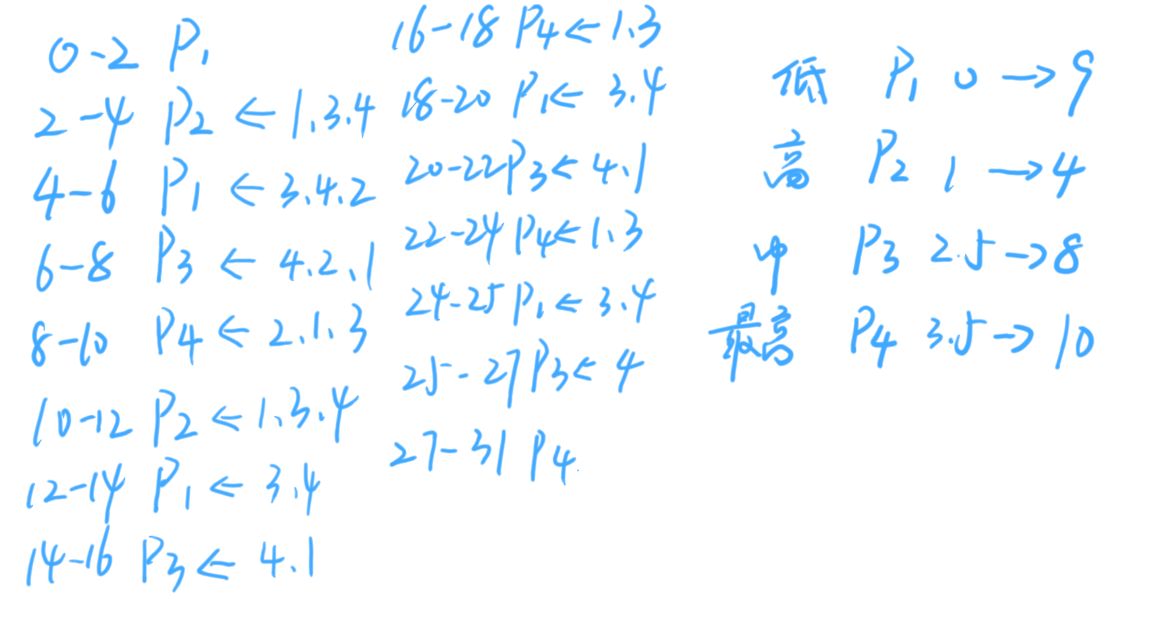
①：调度次序如下：



平均周转时间：（31+14+20.5+10）/4=18.875

平均等待时间：（31-9+14-4+20.5-8+10-10）/4=11.125

②：调度次序如下：



平均周转时间：（25+11+24.5+27.5）/4=22

平均等待时间：22-（9+4+8+10）/4=14.25

1. 某一页式存储管理系统，假设其页表全部存放在内存中。  
    ①若访问内存的时间为120ns，那么访问一个数据的时间是多少？  
    ②若增加一个快表，无论命中与否均需20ns的开销，假设快表的命中率为80％，则此时访问一个数据的时间是多少？（满分8分）

答：

①：120ns\*2=240ns

②：140\*0.8+260\*0.2=164ns

1. 13.在一页式存储管理系统中，逻辑地址长度为16位，页面大小为4096B，已知第0、1、2页依次存放在第10、12、14号物理块中，现有逻辑地址2F6AH，请问其相应的物理地址是多少？（地址以十六进制表示）（满分8分）

答：页面大小4096=2^12，所以前四位表示页号，页号为2，对应14号物理块，物理地址为EF6AH。

1. 假设一个物理存储器有4个页框，一个程序运行的页面走向是：1-2-3-1-4-5-1-2-1-4-5-3-4-5。假定所有页框最初都是空的，分别使用OPT、FIFO、LRU、CLOCK、MIN(滑动窗口τ＝3)、WS(工作集窗口尺寸△＝2)。算法，计算访问过程中所发生的缺页中断次数和缺页中断率。（满分24分）

答：

①最优置换算法OPT：缺页\_\_6\_\_次。中断率42.86%。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **页面序列** | **1** | **2** | **3** | **1** | **4** | **5** | **1** | **2** | **1** | **4** | **5** | **3** | **4** | **5** |
| 页框1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 |
| 页框2 |  | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 页框3 |  |  | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 页框4 |  |  |  |  | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 缺页标记 | F | F | F |  | F | F |  |  |  |  |  | F |  |  |

②先进先出算法FIFO：缺页\_\_\_10\_\_次。中断率71.43%。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **页面序列** | **1** | **2** | **3** | **1** | **4** | **5** | **1** | **2** | **1** | **4** | **5** | **3** | **4** | **5** |
| 页框1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 4 |
| 页框2 |  | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 5 |
| 页框3 |  |  | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 页框4 |  |  |  |  | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 |
| 缺页标记 | F | F | F |  | F | F | F | F |  |  |  | F | F | F |

③最近最少使用算法（LRU）：缺页\_\_\_7\_\_\_次。中断率50%。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **页面序列** | **1** | **2** | **3** | **1** | **4** | **5** | **1** | **2** | **1** | **4** | **5** | **3** | **4** | **5** |
| 页框1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 页框2 |  | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 页框3 |  |  | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 3 | 3 | 3 |
| 页框4 |  |  |  |  | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 缺页标记 | F | F | F |  | F | F |  | F |  |  |  | F |  |  |

④Clock调度算法：缺页\_\_\_10\_\_次。中断率71.43%。

注：解题时需要标记每个页号的引用位(&引用，\*指针)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **页面序列** | **1** | **2** | **3** | **1** | **4** | **5** | **1** | **2** | **1** | **4** | **5** | **3** | **4** | **5** |
| 页框1 | 1& | 1& | 1& | 1& | \*1& | 5& | 5& | 5& | 5& | 5& | 5& | \*5& | 4& | 4& |
| 页框2 | \* | 2& | 2& | 2& | 2& | \*2 | 1& | 1& | 1& | 1& | 1& | 1& | \*1 | 5& |
| 页框3 |  | \* | 3& | 3& | 3& | 3 | \*3 | 2& | 2& | 2& | 2& | 2& | 2 | \*2 |
| 页框4 |  |  | \* | \* | 4& | 4 | 4 | \*4 | \*4 | \*4& | \*4& | 3& | 3 | 3 |
| 缺页标记 | F | F | F |  | F | F | F | F |  |  |  | F | F | F |

⑤MIN(滑动窗口τ＝3)：缺页\_\_\_9\_\_\_次。中断率64.29%。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **页面序列** | **1** | **2** | **3** | **1** | **4** | **5** | **1** | **2** | **1** | **4** | **5** | **3** | **4** | **5** |
| 引用串 | P1 | P2 | P3 | P1 | P4 | P5 | P1 | P2 | P1 | P4 | P5 | P3 | P4 | P5 |
| P1 | √ | √ | √ | √ | √ | √ | √ | √ | √ |  |  |  |  |  |
| P2 |  | √ |  |  |  |  |  | √ |  |  |  |  |  |  |
| P3 |  |  | √ |  |  |  |  |  |  |  |  | √ |  |  |
| P4 |  |  |  |  | √ |  |  |  |  | √ | √ | √ | √ |  |
| P5 |  |  |  |  |  | √ |  |  |  |  | √ | √ | √ | √ |
| IN | P1 | P2 | P3 |  | P4 | P5 |  | P2 |  | P4 | P5 | P3 |  |  |
| OUT |  |  | P2 | P3 |  | P4 | P5 |  | P2 | P1 |  |  | P3 | P4 |

⑥工作集算法（WS）(△＝2)：缺页\_\_\_9\_\_\_次。中断率64.29%。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **页面序列** | **1** | **2** | **3** | **1** | **4** | **5** | **1** | **2** | **1** | **4** | **5** | **3** | **4** | **5** |
| 引用串 | P1 | P2 | P3 | P1 | P4 | P5 | P1 | P2 | P1 | P4 | P5 | P3 | P4 | P5 |
| P1 | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |  |  |  |
| P2 |  | √ | √ | √ |  |  |  | √ | √ | √ |  |  |  |  |
| P3 |  |  | √ | √ | √ |  |  |  |  |  |  | √ | √ | √ |
| P4 |  |  |  |  | √ | √ | √ |  |  | √ | √ | √ | √ | √ |
| P5 |  |  |  |  |  | √ | √ | √ |  |  | √ | √ | √ | √ |
| IN | P1 | P2 | P3 |  | P4 | P5 |  | P2 |  | P4 | P5 | P3 |  |  |
| OUT |  |  |  |  | P2 | P3 |  | P4 | P5 |  | P2 | P1 |  |  |