

操作系统第四次作业

BobAnkh

April 2021

1. 5个作业等待运行，预计它们的运行时间为9、6、3、5和X。以什么次序运行这些作业会使平均响应时间最小？(你的答案将依赖于X。)

答：采用最短作业优先 (SJF) 算法可以获得最小的平均响应时间。以下以各自的作业时间来指代作业本身：

如果 $0 < X \leq 3$ ，作业次序为 X、3、5、6、9

如果 $3 < X \leq 5$ ，作业次序为 3、X、5、6、9

如果 $5 < X \leq 6$ ，作业次序为 3、5、X、6、9

如果 $6 < X \leq 9$ ，作业次序为 3、5、6、X、9

如果 $X > 9$ ，作业次序为 3、5、6、9、X

2. 就绪队列里按先后顺序排列有5个作业A、B、C、D、E，它们所需的运行时间分别为10、1、2、1、5分钟，它们的优先级 $B > E > A > C > D$ ，对下述调度算法计算平均周转时间：

- 先来先服务
- 时间片轮转 (时间片为1分钟)
- 短作业优先
- 最高优先级优先

答：

- 先来先服务

如下图所示进行调度，平均周转时间为： $\frac{1}{5} \times (10 + 11 + 13 + 14 + 19) = 13.4$ (min)

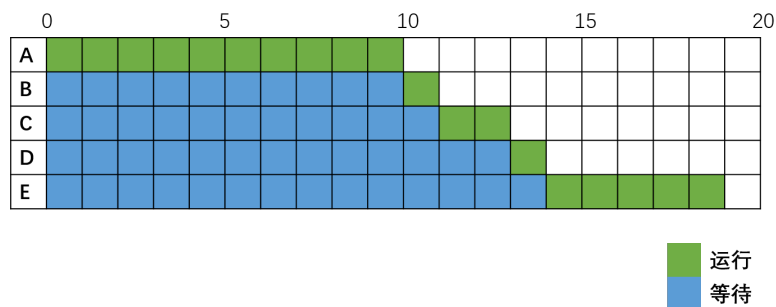


图 1: 先来先服务

• 时间片轮转

如下图所示进行调度，平均周转时间为： $\frac{1}{5} \times (19 + 2 + 7 + 4 + 14) = 9.2$ (min)

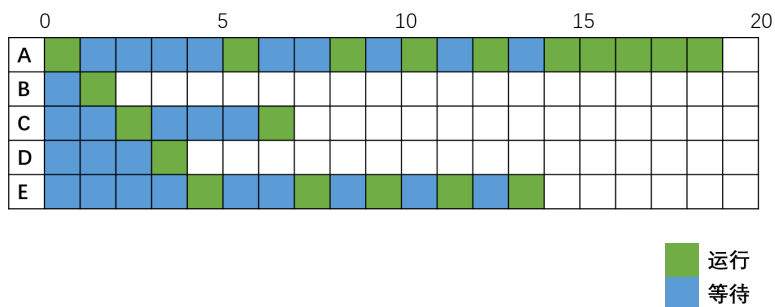


图 2: 时间片轮转

• 短作业优先

如下图所示进行调度，平均周转时间为： $\frac{1}{5} \times (19 + 1 + 4 + 2 + 9) = 7$ (min)

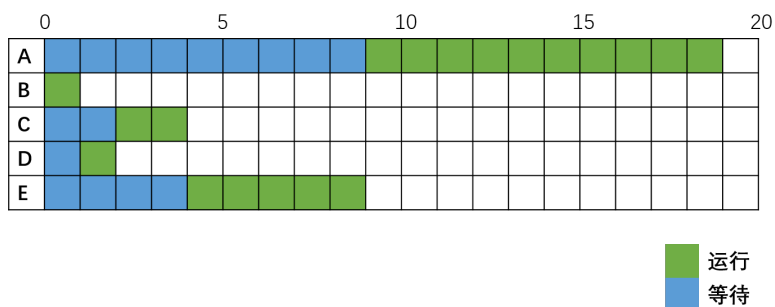


图 3: 短作业优先

• 最高优先级优先

如下图所示进行调度，平均周转时间为： $\frac{1}{5} \times (16 + 1 + 18 + 19 + 6) = 12$ (min)

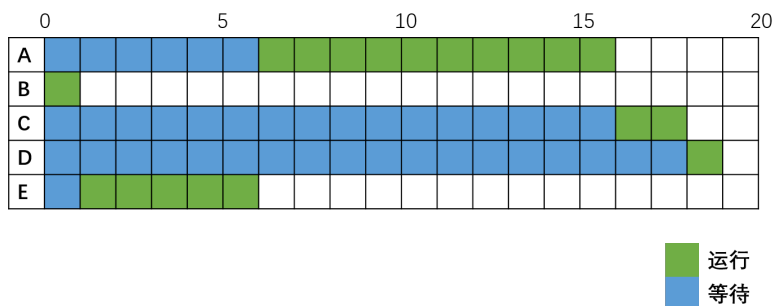


图 4: 最高优先级优先

3. 在一个实时系统中，有三个周期性实时任务：任务 A 要求每 20ms 执行一次，执行时间为 10ms；任务 B 要求 30ms 执行一次，执行时间为 10ms；任务 C 要求 40ms 执行一次，执行时间为 5ms。请在 0 200ms 范围内对 RMS、EDF 和 LLF 三种算法进行比较。

答：如下图所示，每一格表示 5ms，可以看到 RMS 算法会失败，而 EDF 算法和 LLF 算法可以成功调度：

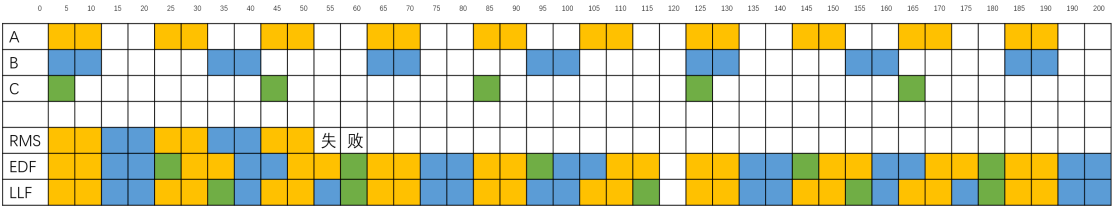


图 5: RMS、EDF、LLF 比较

4. 某时刻，系统的资源分配状态如下。系统是否安全？如果安全，请给出安全序列。

进程	已分配资源			仍需分配			可用资源		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
P_0	2	0	0	0	0	1	0	2	1
P_1	1	2	0	1	3	2			
P_2	0	1	1	1	3	1			
P_3	0	0	1	2	0	0			

答：不安全，在这样的情况下，可用资源只有可能在分配给进程 P_0 的情况下才能够有进程完成任务，回收资源后，可用资源变为 (2,2,1)，此时只有可能给进程 P_3 分配资源才能够有进程完成任务，回收资源后，可用资源变为 (2,2,2)，此时无论分配给进程 P_1 还是进程 P_2 都无法使它们完成任务，无法回收到更多资源，无法正常执行下去，此时即无法给任何一个进程分配资源以使其完成，所以系统会进入死锁状态，此时系统不安全。

5. 某时刻，系统的资源分配状态如下。

1) 系统是否安全？如果安全，请给出安全序列。

2) 如果进程 P_0 和 P_1 均发出请求 $\text{Request}(1, 0, 1)$ ，系统应该如何进行处理？

进程	已分配资源			最大资源需求			可用资源		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
P_0	1	0	0	3	2	2	2	1	2
P_1	4	1	1	6	1	3			
P_2	2	1	1	3	1	4			
P_3	0	0	2	4	2	2			

答：

- 1) 系统安全。给出其中一个安全序列： $\langle P_1, P_0, P_2, P_3 \rangle$ 。事实上安全序列不止这一个，只要第一个是 P_1 ，后面三个可以交换顺序，也是安全序列。
- 2) 系统通过试分配会发现，若将资源分配给 P_0 ，则系统将处于不安全状态；而若将资源分配给 P_1 ，系统仍能够找到一个安全序列，也即这样分配下系统仍然是安全的。因而系统会响应 P_1 的请求将资源分配给 P_1 而阻塞 P_0 的请求。