



东北师范大学

2020 年春季学期操作系统第三次作业：处理机调度

2018 级计算机科学与技术（中美合作）

包亦航
2018011890

2020 年 5 月

问题一：使用 FCFS, SJF, HRRN 对下面的问题进行评估

❖ 假设有五个作业A、B、C、D、E, 它们到达系统的时间分别是0、1、2、3、4, 服务时间分别是4、3、5、2、4, 则采用最高响应比优先算法进行作业调度时, 计算这五个作业的完成时间、周转时间、带权周转时间、平均周转时间以及平均带权周转时间。

解答方案：

各作业情况如下表所示

作业名称	到达时间	服务时间
A	0	4
B	1	3
C	2	5
D	3	2
E	4	4

表 1-1 作业到达时间与服务时间表

我们分别对他们使用不同的调度算法

算法	指标	A	B	C	D	E	平均
FCFS	完成时间	4	7	12	14	18	11
	周转时间	4	6	10	11	14	9
	带权周转时间	1	2	2	5.5	3.5	2.8
SJF	完成时间	4	9	18	6	13	10
	周转时间	4	8	16	3	9	8
	带权周转时间	1	2.67	3.2	1.5	2.25	2.12
HRRN	完成时间	4	7	14	9	18	10.4
	周转时间	1	6	12	6	14	8.4
	带权周转时间	1	2	2.4	3	3.5	2.38

表 1-2 各算法指标对比

为了更好地说明问题, 在这里使用 photoshop 工具对各进程执行效果进行了可视化, 效果如下, 其中, 实线是各进程实际执行的时间, 虚线是各进程等待的时间

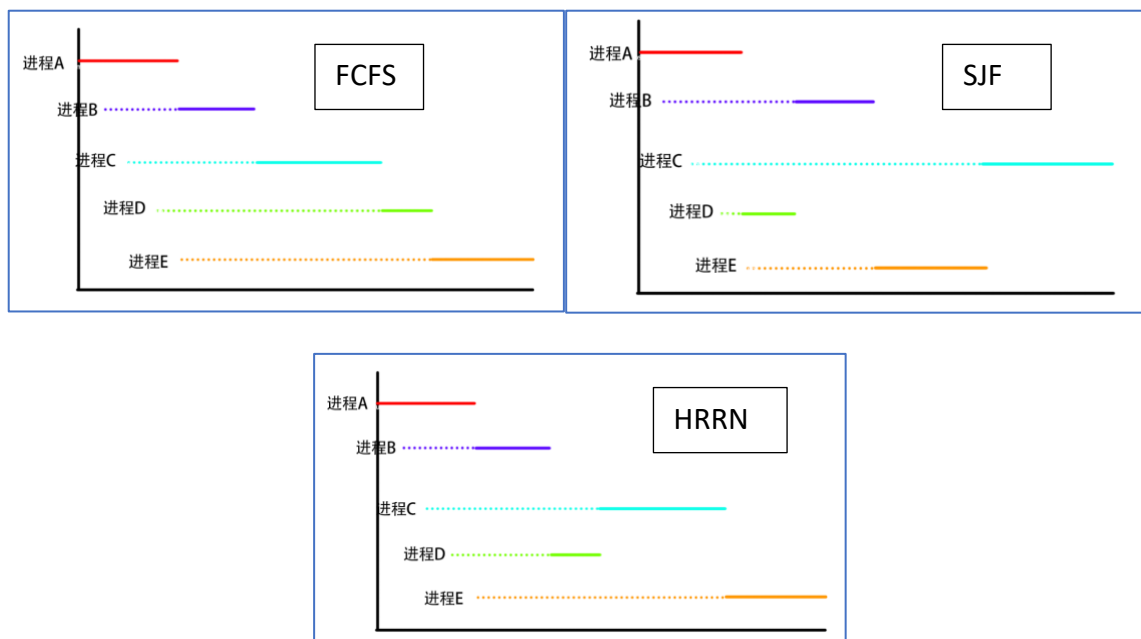


图 1-1 各算法执行过程示意图

首先我们从表格的数据中可以看出，SJF 算法对于这个问题处理序列有较为明显的优势，三个指标的平均值都是最小的。但是我们也应该意识到，全局最优不一定是效果最好的，我们从图中看出，SJF 的进程 C 等待时间过长，如果这些进程由不同的用户所有，那么明显这里是不公平的。反而我们看 HRRN 的全局效果适中且出现等待时间过长的几率相对较小。在实践中应根据实际情况选择适合的算法。

一般化探究：

为了探究本题中涉及到的三种处理机调度算法在一般化情况下的表现，我在这里使用随机生成任务的方式，对三种算法的性能进行简单分析。
(代码见同目录下 processor_dispatching.py)

1 探究方式

为了生成随机作业以及模拟各个算法的运行，这里使用 python 代码进行模拟。代码的设计思路如下：

- 1、通过 python random 模块，生成指定范围内的随机任务，包括他的开始时间和使用时长。同时在这里还可以指定随机生成任务的最早到达时间和最晚到达时间以及每个任务的最长持续时间

- 2、利用 python queue 模块下的 PriorityQueue()方法建立优先队列，作为目前已到达未执行的作业队列。这里的优先级设定根据不同的处理机调度算法不同。FCFS 算法的优先级就是到达时间，SJF 算法的优先级是作业的长短，HRRN 算法的优先级是响应比

3、在 HRRN 算法中，由于优先级是动态变化的，而优先队列又无法对其中的内容进行修改，于是在 HRRN，只要每次有时间的推进，就随时动态将队列中所有元素取出，更新响应比后再压入队列中

4、在工作时，将原先生成的所有进程按到达时间从小到大排序，每次根据目前推进的时间选择作业加入优先队列中参与调度，直到所有作业被调度进队列中。

2 实验分组

为了探究一般化条件下的各算法性能，在下面根据不同的目的设置以下实验分组

分组名称	进程个数	最早进入时间	最晚进入时间	单作业最长持续时间	分组说明
组 1	100	100	100	100	算法正确性验证
组 2	10	0	10	5	小规模密集短作业
组 3	1000	0	1000	5	大规模密集短作业
组 4	1000	0	1000	500	大规模密集长作业
组 5	1000	0	10000	5	大规模稀疏短作业
组 6	1000	0	50000	1000	大规模稀疏长作业

表 2-1 实验分组情况表

3 实验结果

由于生成的进程是随机的，下面各数据都是三次取平均并保留整数部分

组别	指标	FCFS	SJF	HRRN
组 1	平均完成时间	16813	16813	16813
	平均周转时间	16713	16713	16713
	平均带权周转时间	153	153	153
组 2	平均完成时间	17	14	15
	平均周转时间	12	9	10
	平均带权周转时间	6	3	3
组 3	平均完成时间	1462	1125	1203
	平均周转时间	957	619	698
	平均带权周转时间	454	142	203
组 4	平均完成时间	127912	85096	85105
	平均周转时间	127406	84590	84600
	平均带权周转时间	1607	248	249
组 5	平均完成时间	5031	5030	5031

组 6	平均周转时间	3.8	3.7	3.8
	平均带权周转时间	1.3	1.3	1.3
	平均完成时间	248688	164276	167045
	平均周转时间	223873	139462	142230
	平均带权周转时间	2587	191	212

表 2-2 整体结果评估表

在总体结果的基础上，再来看看对个体的影响，我们这里分别计算各组别等待时间最大的三个时间

组别	FCFS			SJF			HRRN		
组 1	51138	51238	51338	51138	51238	51338	51138	51238	51338
组 2	10	12	14	10	14	14	8	14	14
组 3	1967	1971	1972	2062	2064	2067	2020	2022	2023
组 4	238976	239397	239885	238976	239120	239500	238976	239120	239500
组 5	7	8	8	8	9	14	7	7	8
组 6	441579	442540	443123	479609	484304	485050	451645	452754	453075

表 2-3 个体结果评估表

4 实验结论

从以上的数据中我们可以得出以下结论：

- 1、组一情况下三种算法指标相同，验证了算法的正确性
- 2、无论在什么情况下，短作业优先(SJF)总是能获得最优或者持平的整体表现，
- 3、当作业队列中存在长作业或作业相对较为密集时，FCFS 的整体性能明显低于另外两种算法
- 4、大多数情况下，虽然 HRRN 效果均不及 SJF，但是差距都不大，尤其在大规模密集长作业时，二者几乎没有整体性能上的区别。
- 5、从个体来看，我们看到 FCFS 和 HRRN 对于单个作业等待时间的处理明显好于 SJF，除大规模密集长作业外，SJF 均有作业等待时间较长，如果这是个多用户系统的话可能影响用户体验。

综合上述实验，我们发现，HRRN 的表现较为稳定，在各种情况下对整体和个体均有较好的性能，归根结底在于其动态更新优先级的算法特性。但是我们也要意识到，动态更新优先级的会带来更大的系统开销（在实验程序出结果速度上可见一斑），我们在实际应用中还是需要综合考量来选取最优的算法。