

HW3

PB21111686_赵卓

1.

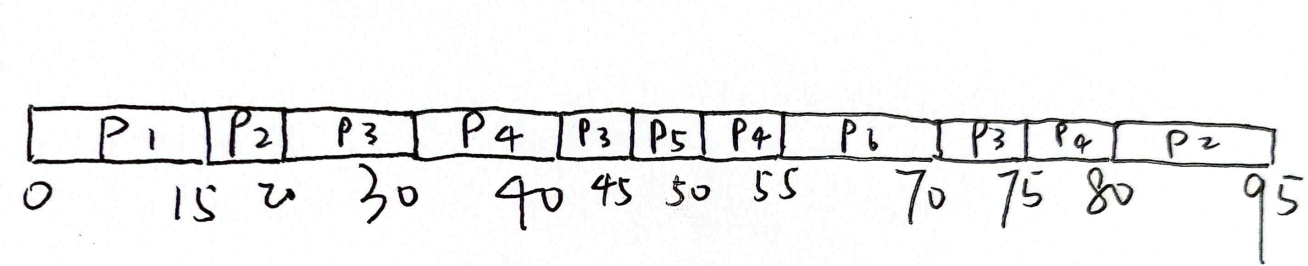
- 在子进程创建成功时会执行到"LINE J"行。对于父进程，fork()会返回子进程的PID，则pid>0，会执行else后面的语句。而对于子进程，fork()会返回0，从而确定这是一个子进程，因此会执行"LINE J"行。因此当子进程创建成功时，"LINE J"行会在子进程中执行，否则会在父进程中执行pid<0后的语句。

2.

- 调度的饥饿现象是指由于调度算法分配不公平或者资源不够时，部分进程长时间甚至永远无法获得资源执行的情况。
- 在介绍的调度算法中，SJF（短作业优先算法）和PSA（优先级调度算法）可能出现饥饿现象。
 - 例：以PSA为例，优先级数从0-127，且优先级数越小，优先级越高。假设有n个进程P1...Pn，Pn优先级为127，而其他进程优先级都比127高，那么就会导致Pn长时间得不到执行，出现饥饿现象。为解决饥饿现象，我们可以采取aging策略，随时间增长提高优先级。

3.

- 甘特图如下：



- 轮转时间如下：

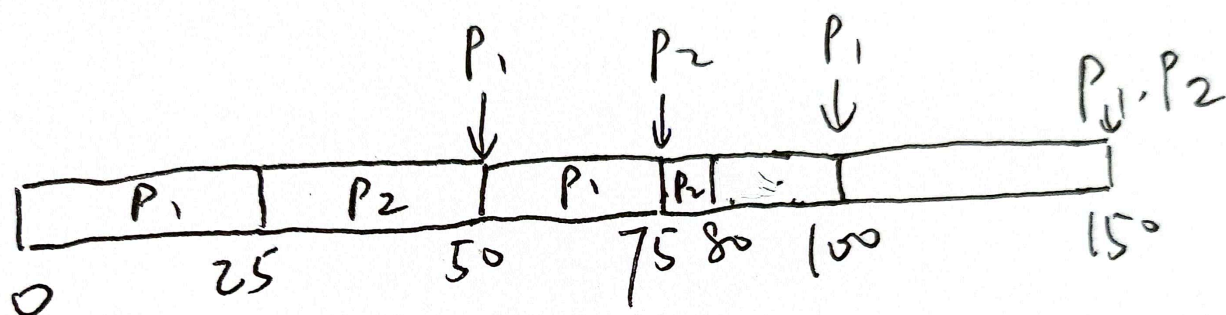
| P1 | P2 | P3 | P4 | P5 | P6 |
|----|----|----|----|----|----|
| 15 | 80 | 55 | 55 | 5 | 15 |

- 等待时间如下：

| P1 | P2 | P3 | P4 | P5 | P6 |
|----|----|----|----|----|----|
| 0 | 60 | 40 | 40 | 0 | 0 |

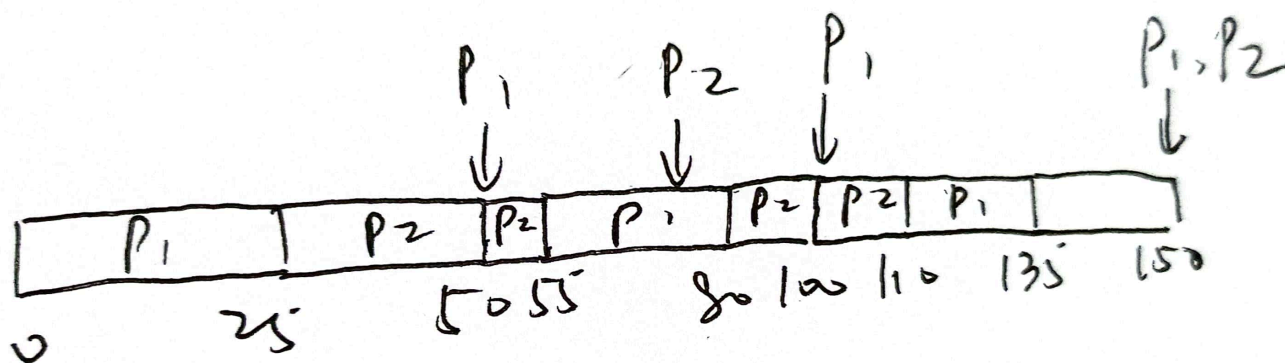
4.

- 从CPU利用率和甘特图两方面来看：
 - 从CPU利用率来看，两个进程总CPU利用率为 $25/50 + 30/75 = 90\%$ ，可以用RM调度。
 - 但是作图可知：



在P2截止时间75之前，P2还没有被执行完，因此不可用RM调度。

- 用EDF调度如下：



5.

- 只用一种调度算法不合适，这样无法充分利用系统资源。
- 正确做法是动态选择调度算法和组合利用调度算法：

- 对于CPU密集程序，选择充分利用计算资源的调度算法；对于I/O密集程序，选择减少等待时间的调度算法。
- 同时将FCFS（先到先服务算法）、PSA（优先级调度算法）、SJF（短作业优先算法）等算法综合使用。