

# Task 1

张配天-2018202180

## 3.2

假设分派器进程名为  $P0$ , 且地址从 100 开始占用 4 个位置

Instruction Address	Process Name
4050-4054	P1
100-104	P0
3200-3204	P2
100-104	P0
5000-5004	P3
100-104	P0
6700-6702	P4
P4	finish
100-104	P0
4055-4059	P1
100-104	P0
3205-3206	P2
P2	finish
100-104	P0
5005-5009	P3
100-104	P0
4060	P1
P1	finish
100-104	P0
5010	P3
P3	finish

### 3.3

#### possible cases

记新建态为  $N$ ，就绪态为  $R$ ，就绪挂起态为  $RS$ ，运行态为  $U$ ，阻塞态为  $B$ ，阻塞挂起态为  $BS$ ，退出态为  $E$

表 1: possible cases

pre_State	next_State	Instance
N	R	操作系统准备好再接纳一个进程
	RS	操作系统准备好再接纳一个进程
	E	超时被终结
R	U	操作系统选择一个处于就绪状态的进程运行
	E	目前已就绪的进程被其父进程或操作系统终结
	RS	由于交换、交互请求、定时、父进程请求、其他 os 因素导致操作系统需要释放主存空间
U	R	正在运行的进程的时间片用完或者有更高优先级的进程进入就绪队列抢占了处理器 ( <i>swap</i> )
	RS	由于交换、交互请求、定时、父进程请求、其他 os 因素导致操作系统需要释放主存空间
	B	进程请求了其必须等待的事件
	E	当前运行的进程表示自己已经完成或被取消
B	BS	由于交换、交互请求、定时、父进程请求、其他 os 因素导致操作系统需要释放主存空间
	R	处于阻塞挂起状态进程等待的事件发生了
	E	超时被终结
RS	R	处于就绪挂起状态的进程被操作系统激活
	E	超时等原因被终结
BS	B	处于挂起状态的进程被操作系统激活
	RS	处于阻塞挂起状态进程等待的事件发生了
	E	等待超时、I/O 失败等原因被终结

## impossible cases

表 2: impossible cases

pre_State	next_State	Instance
N	U	没有就绪的进程不会被分派器分派处理器资源
	B	没有运行的进程不会遇到中断进入阻塞
	BS	没有运行的进程不会遇到使其进入阻塞态的事件
R	N	已经处于就绪队列的进程不会再次被重新创建
	B	没有运行的进程不会遇到中断进入阻塞
U	N	正在运行的进程不会再次被重新创建
	BS	正在运行的进程不会在没有遇到阻塞时直接进入阻塞挂起态
E	N	已经退出的进程不会重新被创建
	R	没有就绪的进程不会进入运行态
	RS	没有运行的进程不会被挂起
	B	没有运行的进程不会进入阻塞
	BS	没有运行的进程不会进入阻塞挂起
RS	N	已经被加载进入辅存的进程不会被重新创建
	B	没有运行的进程不会进入阻塞
	BS	没有运行的进程不会进入阻塞挂起
	U	没有被加载进入内存的进程不会被运行
BS	N	已经被加载进入辅存的进程不会被重新创建
	R	等待的事件没有发生的进程不会进入就绪队列
	U	没有被加载进入内存的进程不会被运行

## 3.5

主要考虑空间的优化；记就绪挂起态优先级最高的进程为  $P_1$ ，就绪态将要被交换的进程为  $P_2$ ，某一进程  $P_i$  被加载进入主存所将占用的空间为  $S(P_i)$ ，记主存总空间为  $S_0$ ，现已占用空间为  $S$ ，则若：

$$S(P_0) + S \leq S_0$$

直接激活  $P_1$ ，加载进入主存，成为就绪态；

otherwise, 交换  $P_1$  和  $P_0$ 。



## 3.12

*pid\_child* 0 