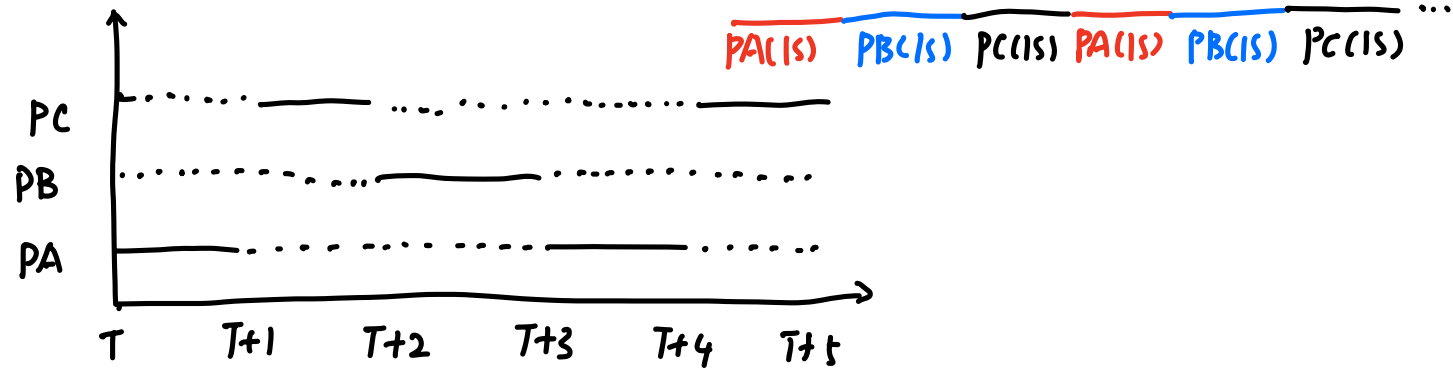
操作系统作业 11-20：

Part 1：

1.

		T	T+1	T+2	T+3	T+4	T+5
P_cpu	PA	0	40	20	0	40	20
	PB	0	0	0	40	20	0
	PC	0	0	40	20	0	40
P_pri	PA	100	102	101	100	102	101
	PB	100	100	100	102	101	100
	PC	100	100	102	101	100	102

3S一个周期，PA，PB，PC轮流执行



2.

T+3时PA得到再次运行的机会 ✓

中断前为用户态，∴进程 u_utime++，p_cpu++，Time::lbolt++，若时间没到1S，中断处理结束，发EOF返回，若到1S，则：

①维护系统时间：Time::lbolt-60；系统全局时间 Time::Time+1

②向中断控制器发EOI指令，STI开中断，唤醒延时睡眠进程

③所有进程 p_time++，p_cpu=max(0, p_cpu-SCHMAG)

④重算优先数大于PUSER的进程优先级

⑤若RunIn被设置，则清除标志，唤醒0进程

⑥重算当前进程优先数，中断返回

下次再运行时恢复现场

1. T+1：第1次保护现场：中断响应过程把PA的用户态寄存器压入自己的核心栈
2. 第2次保护现场：中断返回用户态前执行调度。
3. 恢复现场：先恢复核心态现场

Part 2：

(1) 系统调度操作

用户

4. 恢复现场：恢复用户态现场

响应中断 → 唤醒PA PB → 全局执行下半段 → 执行应用程序

扫描Process表，对每个Process对象，检查其P_wchan成员，检查其睡眠原因是否为chan，若是，调用Process::SetRun函数将PB进程转入就绪态

核心态：响应中断，不唤醒PA → 返回到先前是用户态的第一个时钟中断

(2) 对tout变量的维护

PB waketime到 → 不维护

tout记录所有进程waketime的最小值。每个进程记录自己的waketime，tout = min(waketime, tout)。1025时，PB的waketime = tout，sleep系统调用返回，返回用户态，tout更新为1365

作业1：

时刻1000S：PA放弃CPU，成为低优先级睡眠进程

时刻1020S：PB放弃CPU，成为低优先级睡眠进程

时刻1025S：PA、PB被唤醒

PA上台执行，waketime到期，将tout修正为1365，放弃CPU，成为低优先权睡眠进程

PB上台执行，waketime到期，sleep系统调用完成。PB返回用户态前执行setpri()函数，修正自己的P_pri。

时刻1365S：PA被唤醒，waketime到期，sleep系统调用完成，PA返回用户态

作业3：存在1025S系统无法唤醒PB的情况

①1025S时CPU关中断

②1025S时CPU在核心态运行

每无法唤醒一次就延迟一个时钟滴答

Part3:

(1)先响应磁盘中断

系统先响应磁盘中断，但时钟中断优先级更高，响应时钟中断，时钟中断响应完毕后返回磁盘中断处理 全部完成后能流使用CPU

(2)先响应时钟中断

系统首先响应时钟中断，在时钟中断处理完成后，再去执行磁盘中断。 全部完成后能流使用CPU