2152118 史君宝 UnixV6 交换子系统 订正

第一题：

自己的答案：

（1）T0 时刻，现运行进程 p2 执行 read 系统调用读磁盘文件（磁盘高速缓存中没有 p2 需要的文件数据）

T0 时刻，现运行进程 p2 执行 read（读文件）系统调用，高优先权入睡（p_stat = SSLEEP, p_pri = -50），放弃 CPU。但是内存中并没有就绪的进程供以进程调度，

表格修改如下：

正确答案：

（1）T0 时刻（非整数秒），现运行进程 p2 执行 read 系统调用读磁盘文件（磁盘高速缓存中没有 p2 需要的文件数据）

答：read 系统调用发出 IO 命令后，p2 进程入睡放弃 CPU，高优先权睡眠状态。系统没有就绪进程，idle。0#进程成为现运行进程，idle 期间负责中断处理任务，主要是 2 件事：调整时钟，唤醒 IO 操作结束或闹钟到期的进程。

第二题：

自己的答案：

（2）T1 时刻，已完成 read 系统调用的 p2 进程运行在用户态。p3 等待的 I/O 操作完成。

T1 时刻，正在执行的进程是 p2，此时 p3 等待的 I/O 操作完成，会执行中断处理程序，唤醒 p3 进程，并唤醒 0#进程，在中断处理程序结束之后，由于之前是用户态，会发生例行调度，并将 CPU 让给优先级更高的 0#进程。

0#进程会执行 sched()函数，为盘交换区上就绪的 p3 进程分配内存空间，会在内存中查找低睡（SWAIT）或 SSTOP（暂停）的进程。此时会发现进程 p1 的状态为低优先级睡眠，因此 p1 入选。0#进程会换出 p1 进程，并换入盘交换区的 p3 进程(放在 p1 进程原先占据的内存区域)。

完成上述的换入换出的操作之后，0#进程会 sleep(&Runout,-100)入睡，并执行 swtch()函数将 CPU 让给进程 ，然后进程会执行下去。

之后遇到时间片到的时钟中断，p2 进程和 p3 进程会时间片轮转，轮流执行下去。

正确答案：

（2）T1（T1=T0+1 秒）时刻，已完成 read 系统调用的 p2 进程运行在用户态。p3 等待的 I/O 操作完成。

答：T1 时刻现运行进程是 p2。p2 进程响应中断唤醒 p3 进程。p3 进程在盘交换区上，还要唤醒 0#进程。中断处理结束后 p2 进程将 CPU 让给 0#进程执行对换操作。本题，p1 进程图像的尺寸大于 p3 进程，对换操作可以成功。0#进程先换出内存中的 p1 进程，后换入盘交换区上的 p3 进程。对换操作结束后，0#进程入睡，让出 CPU 给 p3 进程执行系统调用。