

第一题：

题目：

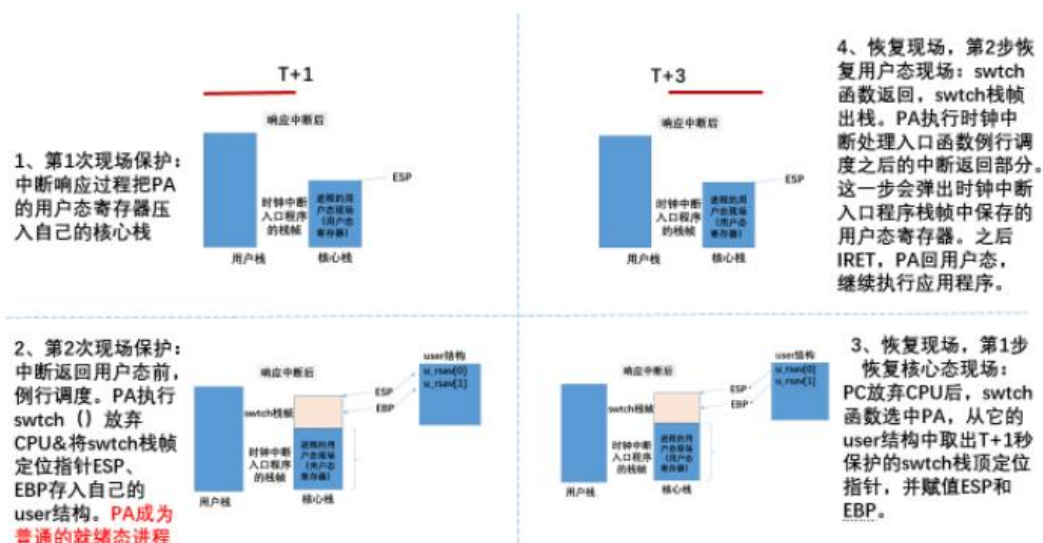
2、T+1 时刻，PA 进程用完时间片放弃 CPU。何时，PA 进程会再次得到运行机会？简述 T+1 时刻系统怎样保护 PA 进程的用户态 CPU 执行现场，下次再运行时系统如何恢复 PA 进程的用户态 CPU 执行现场。

自己的答案：

(2) 在进程的时间片用完之后，会调用 swtch()函数，会将 PA 进程的 ESP 和 EBP 指针保存在 PA 进程的 User 结构中。然后遍历 Process 表中的进程，会选择优先级高的进程上台执行。在 T+3 时刻，通过 Switch()函数选择进程优先级最高的 PA 进程，然后使用 PA 进程 User 结构中的 u\_rav 数组还原 EBP 和 ESP，尽可以继续执行下去了，上述进程应该属于中断过程。

正确答案：

2、T+3 时刻 PA 进程再次得到运行机会。T+1 时刻系统保护 PA 进程的用户态 CPU 执行现场，T+3 时刻恢复 PA 进程的用户态 CPU 执行现场的具体过程图示如下：



第二题：

题目：

分析 1025 时刻 (1) 系统调度操作 (2) Sleep 系统调用下半部对 tout 变量的维护。

(1) 我们知道在系统运行的时候会使用 tout 全局变量，记录所有进程 waketime 的最小值。当达到 1025 时刻的时候，此时的 Time 等于 tout 说明有进程达到定时时刻，就会执行 swtch() 函数依次遍历整个 Process 数组，找到 waketime == tout 的进程，并将 tout 设置为其他进程 waketime 的最小值，即下一个定时进程的时间。

在这一题中，我们会执行 swtch()函数，然后 PB 进程会上台执行，并将 tout 设置为进程 PA 的 waketime。之后会恢复 PB 进程的 ESP 和 EBP 指针，然后从 Sleep(5)之后的代码开始继续执行。

正确答案:

(1) 1025s 的系统调度操作

- 假设 1025s 现运行进程是 PX，运行在用户态。  
响应时钟中断，PX 执行时钟中断处理程序：先前用户态，调整系统时钟 time，唤醒 waketime 到期的 PA、PB 进程。中断返回前，例行调度，PX 优先级不及被唤醒的进程，被剥夺，放弃 CPU。随后，PA、PB 分别执行自己的 sleep 系统调用下半段。
  - PA 执行系统调用下半段：waketime 未到期，设置 tout (1365) 之后，再次入睡。
  - PB 执行系统调用下半段：waketime 到期，sleep 系统调用返回，恢复应用程序优先数 ( $p\_pri \geq 100$ ) 之后，让出 CPU。

这 2 个系统调用下半段全部执行完毕后，CPU 开始执行应用程序。PB、PX 还有其它进程时间片轮转，直至有进程执行系统调用或 1365 秒系统响应时钟中断唤醒 PA。

系统调用下半段执行期间，有可能 PA 先运行，也有可能 PB 先运行 (1) 如果 PA 先运行，入睡后 PB 执行 sleep 系统调用下半段，完成后，系统开始执行应用程序 (2) 如果 PB 先运行，返回用户态前让出 CPU 给 PA 执行系统调用下半段，PA 入睡后系统开始执行应用程序，PB、PX 和其它进程时间片轮转。

- 假设 1025s 现运行进程是 PY，运行在核心态。  
PY 响应时钟中断，先前是核心态，直接返回；不调整系统时钟 time、不会唤醒 PA。上述红色操作延迟，直至先前是用户态的第一个时钟中断。
- 假设 1025s，系统 idle。  
执行上述红色操作。时钟中断处理程序执行完毕后，0#进程选中 PA 或 PB，让出 CPU，恢复睡眠等待 RunIn 或 RunOut 的状态。

第三题:

题目:

全嵌套中断处理模式。低优先级中断处理程序运行时，系统响应高优先级中断处理请求。已知，时钟中断优先级高于磁盘中断优先级。假设：900s，PA 进程执行 sleep (100) 入睡。998s，PB 进程执行 read 系统调用，读磁盘文件。1000s，现运行进程 PX 正在执行应用程序。PA 设置的闹钟到期、PB 读取的磁盘文件数据 IO 结束。分析 1000s，系统详细的调度过程。分两种情况考虑：

- 1、先响应磁盘中断
- 2、先响应时钟中断

自己的答案:



### (1) 先响应磁盘中断

在 1000s，PX 进程在执行，在用户态运行。

之后会先响应磁盘中断请求，并执行一部分代码，直到响应时钟中断。

此时响应时钟中断，由于时钟中断的优先级更高，所以会开始执行时钟中断的代码。

在时钟中断处理完成后，由于进入时钟中断前是核心态，正在执行磁盘中断代码，所以并不会执行进程调度，会返回磁盘中断程序。

磁盘中断程序的剩余代码会继续执行，执行完成后，由于之前是用户态进程 PX 的执行，所以会开始进程调度。此时 PB 进程的优先级会高于 PX 进程，PB 会上台执行。执行 PB 的 read 后面的代码。

在时间片使用完成的时候，会进行进程调度，在进程表中选择优先级高的进程来执行。

之后可能会在 PA，PB，PX 三个进程中轮流执行。

### (2) 先响应时钟中断

在 1000s，PX 进程在执行，在用户态运行。

之后会先响应时钟中断请求，执行时钟中断程序代码。

在时钟中断处理完成后，由于进入时钟中断前是用户态，所以会执行进程调度。此时会响应磁盘中断，并执行磁盘中断程序。

执行完成后，由于之前是时钟中断程序 swtch() 的执行，属于核心态，所以不会开始进程调度。

返回时钟中断程序继续 swtch() 调度。此时 PB 进程的优先级会高于 PX 进程，PB 会上台执行。执行 PB 的 read 后面的代码。

在时间片使用完成的时候，会进行进程调度，在进程表中选择优先级高的进程来执行。

之后可能会在 PA，PB，PX 三个进程中轮流执行。

## 正确答案：

答：

### 1、先响应磁盘中断

1000s。内核任务执行次序：磁盘中断处理程序。。。被打断，时钟中断处理程序（没有调整 time，不会唤醒 PA），磁盘中断处理程序恢复执行，read 系统调用后半段。全部完成后，PX、PB 轮流使用 CPU 执行应用程序。

1000s 之后，先前是用户态的第一次时钟中断。内核任务执行次序：时钟中断处理程序（调整 time，唤醒 PA），sleep 系统调用后半段。全部完成后，PX、PA、PB 轮流使用 CPU 执行应用程序。

### 具体调度细节：

1000s。

现运行进程 PX 执行磁盘中断处理程序，唤醒 PB、RunRun++。其间，嵌套执行中断优先权更高的时钟中断处理程序。后者先前核心态，不调整 time，不会唤醒 waketime 到期的 PA。磁盘中断处理程序运行结束后，现运行进程 PX 被剥夺、放弃 CPU。SRUN 进程集合 { PX, 优先数  $\geq 100$ ; PB, 优先数  $= -50$  }。

PB 优先级高，被选中，执行 read 系统调用后半段。完成后恢复其应用程序优先数 (p\_pri  $\geq 100$ )，让出 CPU。SRUN 进程集合 { PX, 优先数  $\geq 100$ ; PB, 优先数  $\geq 100$  }。PB 和 PX 轮流使用 CPU 执行应用程序。

1000s 之后，先前是用户态的第一次时钟中断。

现运行进程执行时钟中断处理程序，调整 time，唤醒 PA、RunRun++。时钟中断处理程序运行结束后，现运行进程被剥夺、放弃 CPU。SRUN 进程集合 { PX, 优先数 $\geq 100$ ; PA, 优先数 $=90$ ; PB, 优先数 $\geq 100$  }。

PA 优先级最高，被选中，执行 sleep 系统调用后半段。waketime 到期，PA sleep 系统调用返回，恢复应用程序优先数 ( $p\_pri \geq 100$ ) 之后，让出 CPU。SRUN 进程集合 { PX, 优先数 $\geq 100$ ; PA, 优先数 $\geq 100$ ; PB, 优先数 $\geq 100$  }。PA、PB、PX 轮流使用 CPU 执行应用程序。

## 2、先响应时钟中断

1000s，内核任务执行次序：时钟中断处理程序。。被打断，硬盘中断处理程序，时钟中断处理程序恢复执行，read 系统调用后半段，sleep 系统调用后半段。全部完成后，PX、PA 和 PB 轮流使用 CPU 执行应用程序。

### 具体调度细节：

现运行进程 PX 执行时钟中断处理程序，更新 Time::time，STI、EOI 后，中断嵌套，响应磁盘中断。

磁盘中断处理程序唤醒 PB、RunRun++。先前核心态，不调度，磁盘中断处理程序运行结束后，PX 返回、继续执行被打断的时钟中断处理程序。

时钟中断处理程序唤醒 PA、RunRun++(值是 2)。先前用户态，例行调度。RunRun 非

0，现运行进程 PX 被剥夺、放弃 CPU。SRUN 进程集合 { PX, 优先数 $\geq 100$ ; PA, 优先数 $=90$ ; PB, 优先数 $=50$  }。

PB 优先级最高，被选中，执行 read 系统调用后半段。完成后恢复其应用程序优先数 ( $p\_pri \geq 100$ )，让出 CPU。SRUN 进程集合 { PX, 优先数 $\geq 100$ ; PA, 优先数 $=90$ ; PB, 优先数 $\geq 100$  }。

PA 优先级最高，被选中，执行 sleep 系统调用后半段。waketime 到期，PA sleep 系统调用返回，恢复应用程序优先数之后，让出 CPU。SRUN 进程集合 { PX, 优先数 $\geq 100$ ; PA, 优先数 $\geq 100$ ; PB, 优先数 $\geq 100$  }。PA、PB、PX 轮流使用 CPU 执行应用程序。