时钟中断和时间片轮转调度

同济大学计算机系　操作系统作业　　2023-11-20

学号　　2152118　　姓名　　史君宝

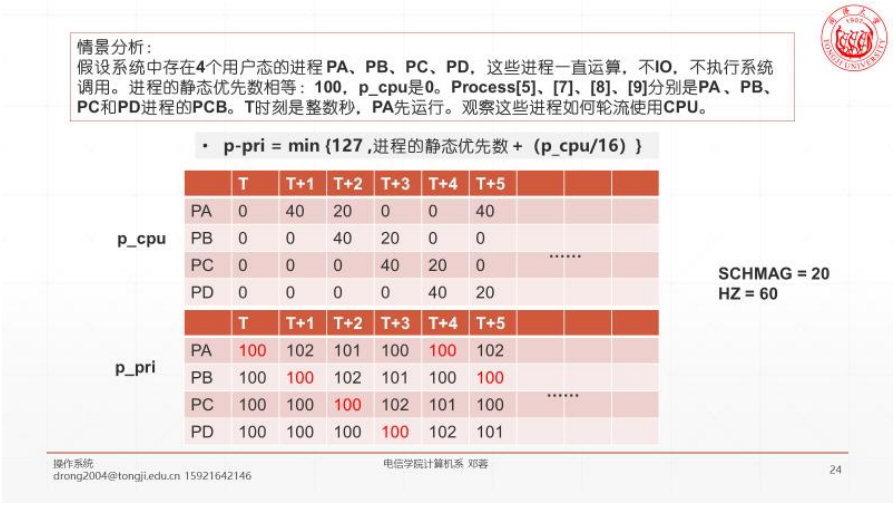
Part 1、Unix V6 时间片轮转调度的实现

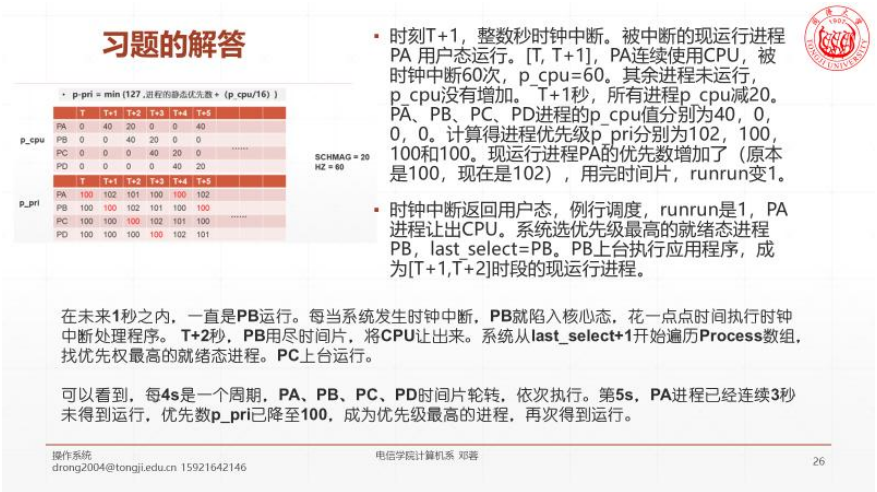
习题： Unix V6++系统中存在 3 个 CPU bound 用户态进程 PA、PB 和 PC。3 个进程静态优先数相等：100，p_cpu 是 0。Process[8]、[5]、[9]分别是 PA 、PB、PC 进程的 PCB。T 时刻是整数秒，PA 先运行。

1、 画进程运行时序图。

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。

参考：PPT24 的表格和对这张 PPT 的讲解。





问题一：

我们按照上述各进程在 process 表中的顺序进行排序

p_cpu

	T	T+1	T+2	T+3	T+4	T+5
PB	0	0	40	20	0	40
PA	0	40	20	0	40	20
PC	0	0	0	40	20	0

p_pri

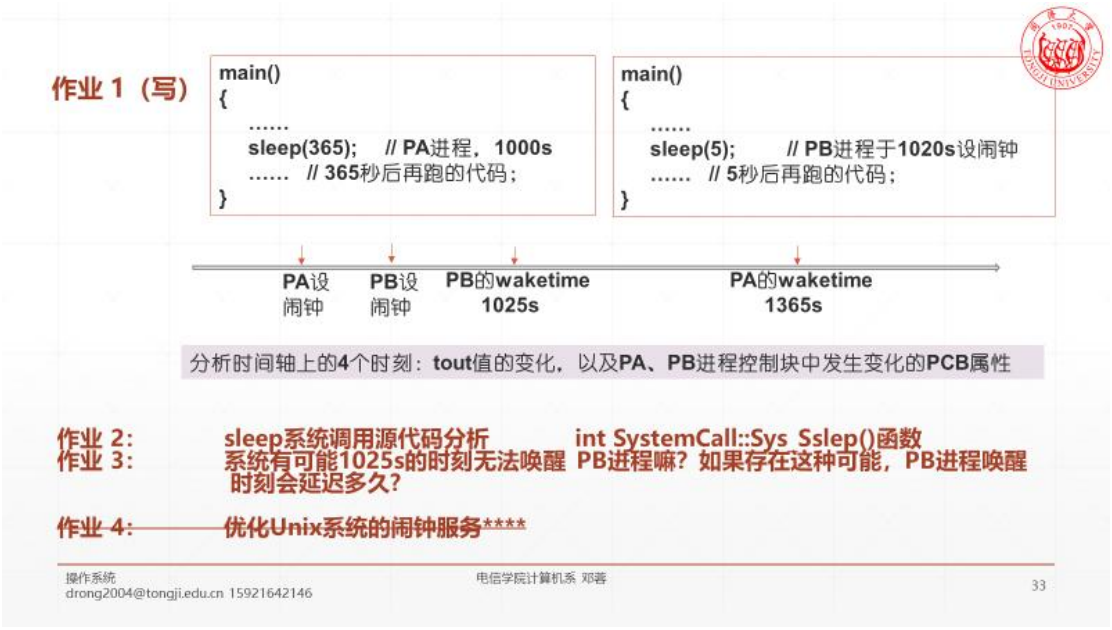
	T	T+1	T+2	T+3	T+4	T+5
PB	100	100	102	101	100	102
PA	100	102	101	100	102	101
PC	100	100	100	102	101	100

问题二：

（1）我们看到上面会在 T 时刻用完时钟片，之后会在 T+1 时刻执行 PB 进程，会在 T+2 时刻执行 PC 进程，在 T+3 时刻进程 PA 的优先级再次变为最高，此时进程 PA 会再次得到执行。因此上面会按照 PA PB PC 的顺序依次执行，每 3 秒一个循环。

（2）在进程的时间片用完之后，会调用 swith()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。

Part 2、定时器服务



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。

（2）在 1025 时刻会执行中断，然后调用 Sleep 函数，并在 Sleep 函数中调用 swtch()函数。则 Sleep 函数的下半部分就是执行 swtch()函数后的代码，此时进程 PB 已经上台，我们需要更新 tout 变量，会遍历所有的进程，找到对应 waketime 的最小值，并将其赋值给 tout 变量，即下一个定时进程。

Part 3、综合题

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

解答：
我们会明确上述过程，在 900s 的时候对于进程 PA 执行 Sleep(100)设置定时时间 1000s。
而在 998s 时进程 PB 会执行 read 操作，等待磁盘读写的 IO 操作。之后两个进程都会下台，转而执行其他进程。在 1000s 时此时在执行 PX 进程，此时来两个中断。

（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。