

时钟中断和时间片轮转调度

同济大学计算机系 操作系统作业参考答案

邓蓉

2023-11-28

Part 1、Unix V6 时间片轮转调度的实现

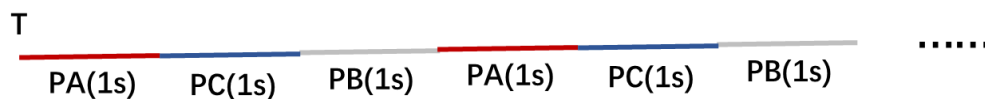
习题： Unix V6++系统中存在 3 个 CPU bound 用户态进程 PA、PB 和 PC。3 个进程静态优先数相等：100，p_cpu 是 0。Process[8]、[5]、[9]分别是 PA 、PB、PC 进程的 PCB。T 时刻是整数秒，PA 先运行。

1、画进程运行时序图。

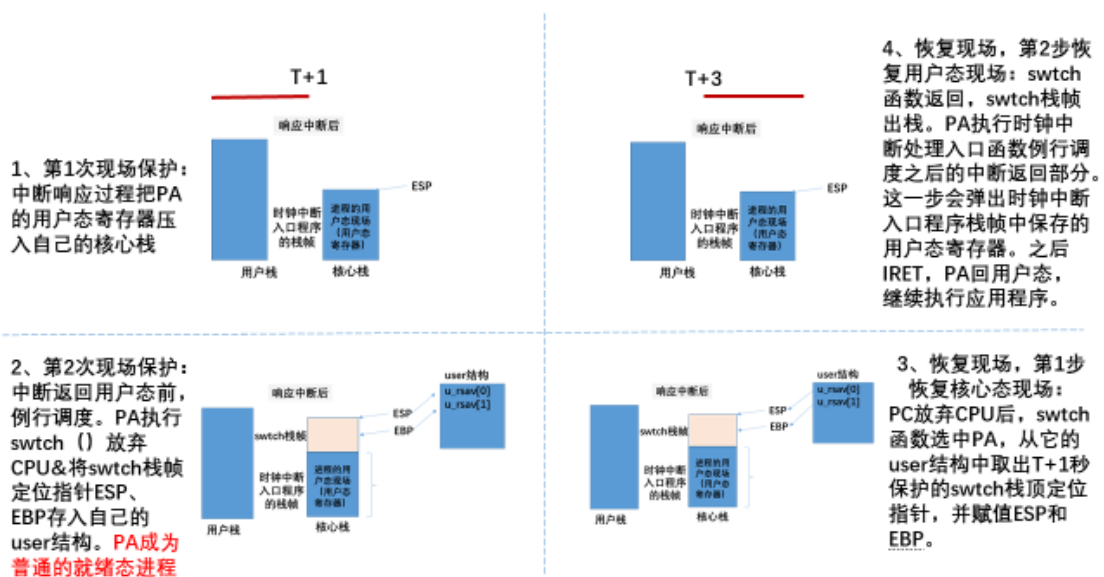
2、T+1 时刻，PA 进程用完时间片放弃 CPU。何时，PA 进程会再次得到运行机会？简述 T+1 时刻系统怎样保护 PA 进程的用户态 CPU 执行现场，下次再运行时系统如何恢复 PA 进程的用户态 CPU 执行现场。

答：

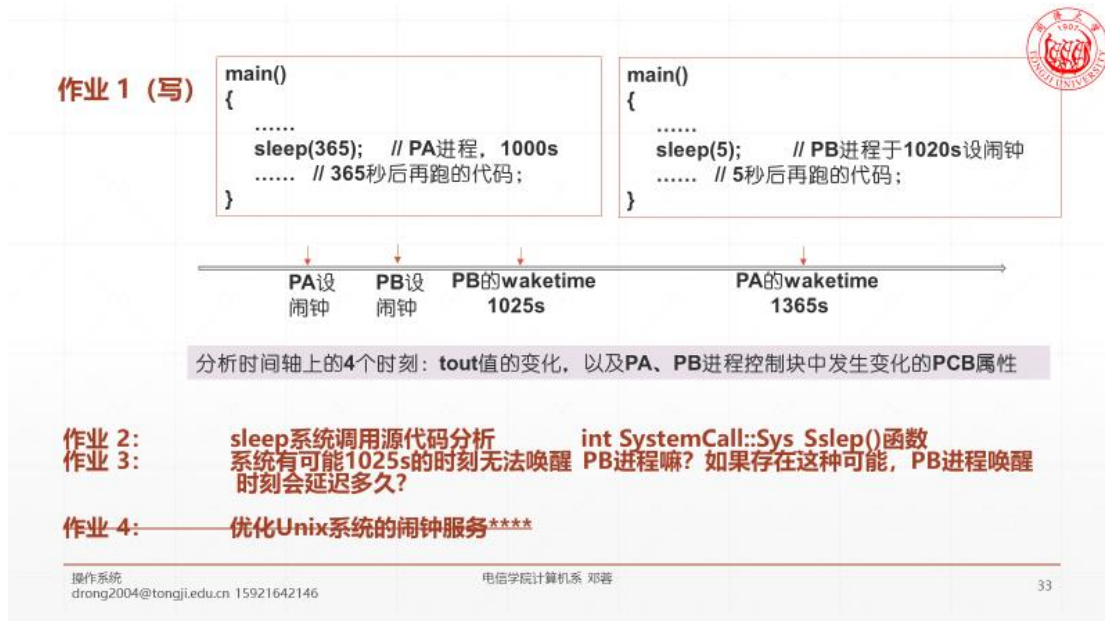
1、



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：



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。

答：

作业 1:

时刻 1000s: PA 设闹钟, waketime=1365s

tout = 1365s

PA 的 PCB: p_stat = SWAIT, p_wchan = &tout (是内核变量 tout 的地址), p_pri = 90。PA 放弃 CPU, 成为低优先级睡眠进程。

时刻 1000s: PB 设闹钟, waketime=1025s

tout = 1025s

PB 的 PCB: p_stat = SWAIT, p_wchan = &tout, p_pri = 90。PB 放弃 CPU, 成为低优先级睡眠进程。

时刻 1025s: PA、PB 被唤醒,

PA、PB 的 PCB: p_stat = SRUN, p_wchan = 0, p_pri 不变还是 90。

PA 调度上台执行 sleep 系统调用的后半部分: waketime 没到, 将 tout 修正为 1365, 放弃 CPU, 成为低优先级睡眠进程。p_pri = 90, p_stat = SWAIT, p_wchan = &tout。

PB 调度上台执行 sleep 系统调用的后半部分: 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 (100+), 与其正在执行应用程序的身份相匹配。

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 2：(略)

作业 3：存在 1025s，系统无法唤醒 PB 进程的情况。

1、1025s，CPU 关中断运行。

2、1025s，CPU 核心态运行。比如，它正在执行低优先级的中断处理程序或正在执行系统调用。

下次时钟滴答（若 CPU 正在执行应用程序）唤醒 PB 进程，PB 唤醒时刻延迟了一个时钟滴答。若下次时钟滴答 CPU 仍然核心态运行，PB 唤醒时刻就再延后一个时钟滴答。

(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 >= 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。

(2) 1025s，Sleep 系统调用下半部对 tout 变量的维护。

PB waketime 到，sleep 系统调用返回，不参与 tout 维护。

1025s，tout==time，不再是所有需要再次入睡的定时器进程 waketime 的最小值。PA，waketime 未到，需要再次入睡。所以用自己的 waketime 更新 tout 值。

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，

入睡优先数是 90。998s, PB 进程执行 read 系统调用, 读磁盘文件, 入睡优先数是-50。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。只考虑 PX、PA 和 PB。

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 ≥ 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 ≥ 100 ; PB, 优先数 ≥ 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 ≥ 100 ; PA, 优先数 $= 90$; PB, 优先数 ≥ 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 ≥ 100 ; PA, 优先数 ≥ 100 ; PB, 优先数 ≥ 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 ≥ 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 ≥ 100 ; PA, 优先数 $=90$; PB, 优先数 ≥ 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 ≥ 100 ; PA, 优先数 ≥ 100 ; PB, 优先数 ≥ 100 }。PA、PB、PX 轮流使用 CPU 执行应用程序。