

操作系统 第四章 进程管理

第一阶段 小结

同济大学计算机系

例题

系统并发 3 个进程，PA CPU bound，始终运算；PB 下载网络数据；PC 使用打印机打印文档。已知：网卡中断优先级>键盘中断优先级>打印机中断优先级。T1 时刻，PB 执行 read 系统调用读网络数据包；T2 时刻，PC 执行 write 系统调用使用打印机。T3 时刻，PA 进程用户态执行，系统相继收到打印机中断请求和网卡中断请求（一）详细分析这 3 个时刻系统的调度状态（二）简述 T1、T2、T3 系统发生的调度事件。←





系统并发 3 个进程，PA CPU bound，始终运算；PB 下载网络数据；PC 使用打印机打印文档。已知：网卡中断优先级>键盘中断优先级>打印机中断优先级。T1 时刻，PB 执行 read 系统调用读网络数据包；T2 时刻，PC 执行 write 系统调用使用打印机。T3 时刻，PA 进程用户态执行，系统相继收到打印机中断请求和网卡中断请求（一）详细分析这 3 个时刻系统的调度状态（二）简述 T1、T2、T3 系统发生的调度事件。←

T1

(1) T1 时刻，现运行进程 PB，执行 read 系统调用上半段，入睡等待网络数据：p_stat = SSLEEP，p_pri = -60，p_wchan = &a，a 指向用来装网络数据的内核缓存块。←
read 系统调用 Sys_Read→Sleep→Swch，PB 放弃 CPU。←

Sys_Read()

{

上半段

sleep (&接受网络数据的缓存 a, -60)

下半段，将缓存 a 中的数据送用户空间，数据段/堆/栈，供应用程序处理。

}

系统并发 3 个进程，PA CPU bound，始终运算；PB 下载网络数据；PC 使用打印机打印文档。已知：网卡中断优先级>键盘中断优先级>打印机中断优先级。T1 时刻，PB 执行 read 系统调用读网络数据包；T2 时刻，PC 执行 write 系统调用使用打印机。T3 时刻，PA 进程用户态执行，系统相继收到打印机中断请求和网卡中断请求（一）详细分析这 3 个时刻系统的调度状态（二）简述 T1、T2、T3 系统发生的调度事件。←



（2）T2 时刻，现运行进程 PC，执行 write 系统调用上半段，向打印机发送命令后，入睡等待打印操作完成：p_stat = SWAIT，p_pri = 20，p_wchan = &b，b 指向存放打印数据的内核缓存块。←

write 系统调用 Sys_Write→Sleep→Swch，PC 放弃 CPU。←

Sys_Write()

{

上半段，从用户空间读出应用程序需要打印的数据，送内核缓存 b；

向打印机发写命令；

sleep (& b, 20) // 睡眠等待打印操作结束

下半段

}

随后系统只有一个进程 PA，PA 是现运行进程，Curpri >= 100。一直是它在运行。←

系统并发 3 个进程，PA CPU bound，始终运算；PB 下载网络数据；PC 使用打印机打印文档。已知：网卡中断优先级>键盘中断优先级>打印机中断优先级。T1 时刻，PB 执行 read 系统调用读网络数据包；T2 时刻，PC 执行 write 系统调用使用打印机。T3 时刻，PA 进程用户态执行，系统相继收到打印机中断请求和网卡中断请求（一）详细分析这 3 个时刻系统的调度状态（二）简述 T1、T2、T3 系统发生的调度事件。←

T3



(3) T3 时刻，CPU 响应中断，现运行进程 PA 用户态运行。中断优先级是 0，响应打印机中断请求；随后，网络中断请求中断优先级高于打印机中断请求，中断嵌套，打印机中断处理程序被打断，系统执行网络中断处理程序。后者执行完毕后，先前核心态不调度，不考虑剥夺 PA 进程，它继续运行打印机中断处理程序。执行完毕后，先前用户态，例行调度。←

系统并发 3 个进程，PA CPU bound，始终运算；PB 下载网络数据；PC 使用打印机打印文档。已知：网卡中断优先级>键盘中断优先级>打印机中断优先级。T1 时刻，PB 执行 read 系统调用读网络数据包；T2 时刻，PC 执行 write 系统调用使用打印机。T3 时刻，PA 进程用户态执行，系统相继收到打印机中断请求和网卡中断请求（一）详细分析这 3 个时刻系统的调度状态（二）简述 T1、T2、T3 系统发生的调度事件。←

现在考虑调度系统的状态变化。←

- 中断处理期间，调度系统的状态←

- 现运行进程 PA 不会被剥夺。这是因为 Unix V6++核心态不调度。←
- 整个中断处理过程，Curpri 没变、是 PA 的 p_pri，一个大于 100 的值。←
- 中断嵌套执行不影响 CPU 执行的中断处理逻辑，←
 - ◆ 网卡中断处理程序送数据入缓存 a，WakeUpAll 唤醒 PB 进程，RunRun++；←
 - ◆ 打印机完成打印任务（缓存 b 中的数据已经打印完成），中断处理程序 WakeUpAll 唤醒 PC 进程，RunRun++。←

- 所有中断处理完成时刻←

- 先前用户态，例行调度，Swch()→Select()，后者选优先级最高的就绪进程。←
- 候选进程集合 {PA, p_pri ≥ 100; PB, p_pri == -60; PC, p_pri == 20}，选中 PB。←

Sys_Read()

{

上半段

sleep (&接受网络数据的缓存 a, -60)

下半段，将缓存 a 中的数据送用户空间，数据段/堆/栈，供应用程序处理。

}

PB

Sys_Write()

{

上半段，从用户空间读出应用程序需要打印的数据，送内核缓存 b；

向打印机发写命令；

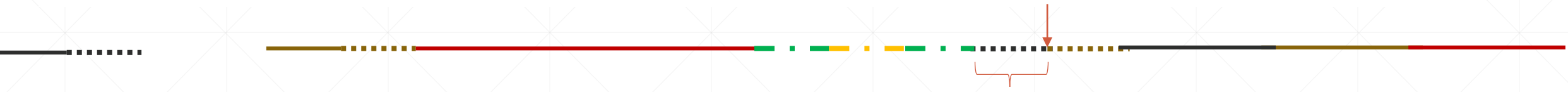
sleep (&b, 20) // 睡眠等待打印操作结束

下半段

}

PC

系统并发 3 个进程，PA CPU bound，始终运算；PB 下载网络数据；PC 使用打印机打印文档。已知：网卡中断优先级>键盘中断优先级>打印机中断优先级。T1 时刻，PB 执行 read 系统调用读网络数据包；T2 时刻，PC 执行 write 系统调用使用打印机。T3 时刻，PA 进程用户态执行，系统相继收到打印机中断请求和网卡中断请求（一）详细分析这 3 个时刻系统的调度状态（二）简述 T1、T2、T3 系统发生的调度事件。←



PB 执行 read 系统调用下半段，完成后放弃 CPU。具体而言：←

- Swch→Sleep→read 系统调用 Sys_Read()，执行下半段送数据入用户空间，供应用程序处理。←
- Sys_Read()→Trap()，Setpri()，PB 恢复用户空间优先数。p_pri 从 -60 升至大于等于 100，现运行进程优先权下降，RunRun++。←
- Trap()→SystemCallEntrance()，系统调用返回时的例行调度：RunRun 非 0，PB 被剥夺，让出 CPU。←
- 候选进程集合 { PA, p_pri ≥ 100; PB, p_pri ≥ 100; PC, p_pri == 20}, 选中 PC。←

```

Sys_Read( )
{
    上半段 .....
    sleep ( &接受网络数据的缓存 a, -60)
    下半段，将缓存 a 中的数据送用户空间，
    数据段/堆/栈，供应用程序处理。
}
    
```

系统并发 3 个进程，PA CPU bound，始终运算；PB 下载网络数据；PC 使用打印机打印文档。已知：网卡中断优先级>键盘中断优先级>打印机中断优先级。T1 时刻，PB 执行 read 系统调用读网络数据包；T2 时刻，PC 执行 write 系统调用使用打印机。T3 时刻，PA 进程用户态执行，系统相继收到打印机中断请求和网卡中断请求（一）详细分析这 3 个时刻系统的调度状态（二）简述 T1、T2、T3 系统发生的调度事件。←

PC 执行 write 系统调用下半段，完成后放弃 CPU。具体而言：←

- Swch→Sleep→Write 系统调用 Sys_Write()，执行下半段。目前来看，没啥实质性的工作要做。。。中断发生时 IO 任务已经完成，输出的数据在打印纸上了。。。←
- 系统调用返回时的例行调度，RunRun 非 0，PC 被剥夺，让出 CPU。←
- 候选进程集合 {PA, p_pri ≥ 100; PB, p_pri ≥ 100; PC, p_pri ≥ 100}，都有被选中的可能。←

Sys_Write()

{

上半段，从用户空间读出应用程序需要打印的数据，送内核缓存 b;
向打印机发写命令;
sleep (& b, 20) // 睡眠等待打印操作结

下半段

}

系统并发 3 个进程，PA CPU bound，始终运算；PB 下载网络数据；PC 使用打印机打印文档。已知：网卡中断优先级>键盘中断优先级>打印机中断优先级。T1 时刻，PB 执行 read 系统调用读网络数据包；T2 时刻，PC 执行 write 系统调用使用打印机。T3 时刻，PA 进程用户态执行，系统相继收到打印机中断请求和网卡中断请求（一）详细分析这 3 个时刻系统的调度状态（二）简述 T1、T2、T3 系统发生的调度事件。←



最后，3个用户态进程，PA、PB、PC时间片轮转，轮流执行应用程序。