**0#进程和盘交换区**

同济大学计算机系 操作系统作业 2023-11-22

学号 2151769 姓名 吕博文

例题：假设某UNIX 系统中只存在4 个进程，分别是：睡眠中的0#进程，正在CPU 上执行的pa 进程，内存中就绪状态的pb 进程（CPU bound，不会IO），盘交换区上低优先级睡眠的pc 进程。已知，当前时刻T=1000.5 s，内存空间已满。请尽量详细地分析以下时刻系统中与进程调度和中断相关的行为，并修改下表中的相关字段。

1. 1000.5 s，现运行进程pa 执行read（读文件）系统调用
2. 1001.5 s，pc等待的IO操作完成

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 占用空间 | 优先数 | 状态 | 位置 | p\_time |
| 0# | - | -100 | 高睡（RunOut） | SLOAD | - |
| pa | 100K | 103 | 执行 | SLOAD | 2 |
| pb | 10K | 101 | 就绪 | SLOAD | 2 |
| pc | 60K | 90 | 低睡 | ~SLOAD | 3 |

解：

* 1. 1000.5 s，现运行进程pa 执行read（读文件）系统调用

1000.5s，现运行进程pa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。随后，系统选中pb进程，因为它是内存中就绪的唯一进程。表格修改如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 占用空间 | 优先数 | 状态 | 位置 | p\_time |
| 0# | - | -100 | 高睡（RunOut） | SLOAD | - |
| pa | 100K | -50 | 高睡SSLEEP | SLOAD | 2 |
| pb | 10K | 101 | 执行 | SLOAD | 2 |
| pc | 60K | 90 | 低睡 | ~SLOAD | 3 |

1001.0s, 现运行进程 pb 执行整数秒时钟中断处理程序，会++所有进程的 p\_time 。修正用户态进程的优先数，pb此轮执行了 0.5s，p\_cpu增加30，整数秒时减20。总的效果：p\_cpu加了10。整除16后，p\_pri可能是102，也可能是101。表格修改如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 占用空间 | 优先数 | 状态 | 位置 | p\_time |
| 0# | - | -100 | 高睡（RunOut） | SLOAD | - |
| pa | 100K | -50 | 高睡SSLEEP | SLOAD | 3 |
| pb | 10K | 101/102 | 执行 | SLOAD | 3 |
| pc | 60K | 90 | 低睡 | ~SLOAD | 4 |

* 1. 1001.5 s，pc等待的IO操作完成

1001.5 s。

* 现运行进程pb执行中断处理程序，唤醒pc，同时唤醒 0#进程。
* 中断处理程序执行完毕后，pb将CPU让给0#进程执行sched（）。0#进程为盘交换区上的pc进程分配内存空间。首先找内存中低睡（SWAIT）或SSTOP（暂停）的进程，未果。pc的p\_time达到4s，值得为它换出内存中高优先级或就绪的进程。pa的p\_time达到3s，入选。0#进程首先换出pa，之后换入盘交换区上的pc（放在pa原先占据的内存区域）\*。
* 完成对换操作后，0#进程 sleep(&Runout,-100)入睡，执行swtch（）将CPU让给内存中就绪的pc进程。pc执行系统调用后半部。完成后，
* pc和pb时间片轮转，轮流执行应用程序。

表格修改如下：

\*处的细节，不要求掌握。 就是换入、换出过程中，0#进程会睡眠等待IO结束的，此时pb使用CPU，是现运行进程。一旦IO完成，pb立即将CPU让给0#进程，因为它优先级最高。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 占用空间 | 优先数 | 状态 | 位置 | p\_time |
| 0# | - | -100 | 高睡（RunOut） | SLOAD | - |
| pa | 100K | -50 | 高睡SSLEEP | ~SLOAD | 0 |
| pb | 10K | 101/102 | 执行SRUN | SLOAD | 3 |
| pc | 60K | 100+ | 就绪SRUN | SLOAD | 0 |

习题：T0时刻，某UNIX V6++系统进程状态如下表所示。内存空间已满，除图示空间外，其余空间不可用。请尽量详细地分析以下时刻系统中与进程调度和对换操作（swap in，swap out）相关的行为，并修改下表中的相关字段。本题不考虑时钟中断。1小题，2小题相关。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 占用空间 | 状态 | 位置 | 内存起始地址 |
| 0# | - | 高睡（RunOut） | SLOAD | \*\*\*\* |
| p1 | 40K | 低睡 | SLOAD | 0x00408000 |
| p2 | 30K | 执行 | SLOAD | 0x00430000 |
| p3 | 30K | 低睡 | ~SLOAD | 0x00450000 |

（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，p2执行read系统调用读取磁盘文件，进入高优先级睡眠状态，放弃CPU，由于此时内存中没有就绪态的进程，0#进程执行Sched（）函数尝试在外存中寻找就绪进程，由于外存中也没有就绪进程，0#进程入睡，此时没有进程利用CPU。表格如下：

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。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 占用空间 | 状态 | 位置 | 内存起始地址 |
| 0# | - | 高睡（RunOut） | SLOAD | \*\*\*\* |
| p1 | 40K | 低睡 | SLOAD | 0x00408000 |
| p2 | 30K | 高睡 | SLOAD | 0x00430000 |
| p3 | 30K | 低睡 | ~SLOAD | 0x00450000 |

（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。

P3进程等待的I/O操作完成后，p2进程进入中断处理程序，唤醒p3,同时唤醒0#进程，执行完中断处理程序后，p2进程让CPU给0#进程，0#进程执行Sched()函数尝试进行swap in 操作将p3进程换入内存，首先寻找内存中低优先级睡眠状态的进程，找到了p1进程，0#进程首先换出p1进程至盘交换区，随后将p3进程换入内存（占据p1进程原来的内存地址），完成对换操作后，0#进程入睡，进行Switch()函数调用根据p2和p3此时的进程优先数来决定谁上台运行。此时的表格如下：（以最后p3执行为例）

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。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 占用空间 | 状态 | 位置 | 内存起始地址 |
| 0# | - | 高睡（RunOut） | SLOAD | \*\*\*\* |
| p1 | 40K | 低睡 | ~SLOAD | \*\*\*\* |
| p2 | 30K | 就绪 | SLOAD | 0x00430000 |
| p3 | 30K | 执行 | SLOAD | 0x00408000 |