

CS302: Lab7 Report

Name: 陆荻芸 SID: 12011537

Answer 1

用户进程 2 号进程被创建了之后，开始执行 `rr.c`。首先 2 号进程进入 `for` 循环后，开始 `fork` 出子进程。由于是单核 `cpu`，所以此时将继续在父进程执行循环，`fork` 出 3、4、5、6、7 五个子进程。

`fork` 完毕后接下来第一次调用 `waitpid` 时将会首先等待 3 号子进程。由于 3 号子进程此时不是僵尸状态，所以 2 号进程会被挂起，`schedule` 函数调度 3 号进程开始执行。3 号从 `fork` 后的 `if` 内的内容执行，因为该进程运行时间大于 `rr` 设置的时间片，所以到时间后，会发生时钟中断。发生时钟中断后，下一次时钟中断将会被设置并且完成进度调度，更新剩余时间片大小。剩余时间片大小为 0 的进程会被标记为需要调度，在中断处理结束之后进行调度。当进程分到时间片，开始运行时会被移除出队列。当进程时间片执行结束且进程状态仍为 `Runnable` 时会被重新加入队尾。3 号执行完一个时间片后将继续调度执行队头的 4 号。同样地继续执行 5、6、7，并循环 3、4、5、6、7 的顺序执行一直到 3 号跑完之后。

由于这 5 个子进程都是相同的，所以 3 会最先结束。3 号结束之后会调用 `do_exit` 函数。因为此时父进程 2 号进程被挂起，所以 3 号结束后会唤醒 2 号，自己进入僵尸状态，把 2 号加入队列等待调度。然后按照队列再结束 4、5、6、7，然后都会检查 2 的状态。到 2 后 2 会逐个检查子进程的状态然后回收资源。回收完之后 `wait` 结束。然后 2 号返回，执行结束。