

CS334 Lab7 Report

产生的进程是通过一个调度队列进行实现的。

通过为每一个进程维护一个最大运行时间片，当一个进程运行了其最大运行时间片的时间之后，调度器将其标记为需要调度，并且会将它的进程控制块放在进程队列的队尾，并且会重置其时间片。

对于需要入队的进程，调度器会放在进程调度队列的队尾，并且特判这个进程的剩余时间片，如果为0，也就是表明该进程刚刚被收回时间片，那么就需要将剩余的时间片设置为最大的时间片。

在时钟中断的时候会调用 `proc_tick` 函数，对当前正在运行的进程的剩余时间片减一。如果在减一之后，该进程的剩余时间片为 0，就直接将该进程标记为需要调度，这样在有一个进程剩余时间片为 0 的时候，内核会在判断需要调度的进程的时候，进行对该进程的调度。

代码的主进程在调用 `waitpid()` 之后，找到 `pid[0] = 3` 的进程，如果该进程已经处于僵尸状态则会收回他的资源，并且结束等待。如果该进程不是僵尸状态，父进程则会进入睡眠状态，并且调用调度程序，将子进程调度运行。

此时会运行 3 号子进程，会在 `fork()` 语句的位置开始执行，之后进入循环，在 `trap.c` 进行调度，之后会调度到 4 号进程。并且会重复这样的循环调度过程。也就是调度顺序为 34567,34567...。直到某一个进程运行结束。由于每个子进程都是相同的，所以 3 号进程最先执行结束。

结束之后就会进入 `do_exit` 函数，将进程自己的状态设置为僵尸状态，唤醒父进程，将父进程插入到运行队列，之后进行调度，也就是调度到父进程。

回到初始的判断状态，这时候子进程为僵尸状态，那么收回其资源，结束 `wait()` 函数