Lab6 Report

12011702 张镇涛

1

sstatus中的SPP域用于保存trap发生时的权限模式（即处于什么态），trap结束后sret指令会根据 SPP 的值恢复权限模式。user_main()调用kernel_execve()实现exec的功能。而内核中我们是实现的是 do_execve(), 在do_execve() 函数中，以下代码

```
tf->status = sstatus & ~(SSTATUS_SPP | SSTATUS_SPIE);
```

将sstatus的SPP域置0，使得中断处理结束sret返回时返回至Umode。

2

在用户态进行系统调用的核心操作是，通过内联汇编进行 ecall 环境调用。这将产生一个trap, 进入S mode进行异常处理。

用户系统调用会调用到`syscall`函数，首先初始化参数列表，然后将参数列表依次取出，然后将参数值放入寄存器中，并执行`ecall`指令以及一些其他的汇编指令，并将返回值存到ret。

这里的ecall会触发trap进入S态处理，由`exception_handler`函数，在CAUSE_USER_ECALL中进行处理，再其中将sepc的值加4，并调用内核中的`syscall`方法。然后`syscall.c`中根据参数调用对应的`syscall`方法，从而完成系统调用。

3

1. 在进程执行完工作后，需要退出，释放资源，退出进程是 调用 `do_exit` 函数来实现的。首先执行 `lcr3(boot_cr3)` 回收内存资源，调用 `exit_mmap` 函数释放 `mm` 中的 `vma` 描述的进程合法空间中实际分配的内存。
2. 设置进程的状态为 `PROC_ZOMBIE`, 等待父进程回收
3. 如果当前进程的父进程处于等待子进程的状态，则唤醒父进程让父进程回收资源。
4. 如果该进程还有子进程，那么就指向第一个孩子，把后面的孩子全部置为空，然后把孩子过继给内核线程 `initproc` ，把子进程插入到`initproc` 的孩子链表中，如果某个子进程的状态是僵尸的状态，并 且 `initproc` 的状态时等待孩子的状态，则唤醒 `initproc` 来回收子进程的资源。
5. 然后开启中断，执行 `schedule` 函数，选择新的进程执行

需要从U态转换成S态

4

子进程已经exit()，但是父进程还存在并且没有进入wait()收集子进程。此时子进程成为僵尸进程。