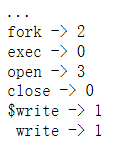
**第一部分：系统调用跟踪**

修改核，给每一个系统调用打印一行内容。显示调用的名字与变量个数



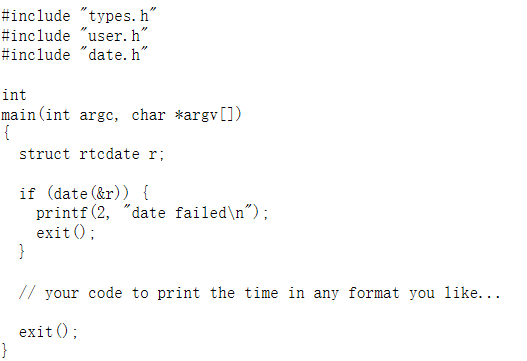
Shell的输出与系统调用的跟踪是相互混合在一起的，这是由于shell使用write系统调用来打印这个输出。

提示：修改syscall.c里面的syscall（）

**第二部分：时间系统调用**

第二个任务是给xv6增加一个新的系统调用，练习的目的是为了让你看到系统调用机制的一些关键点。你的新系统调用会得到当前UTC的时间，并返回给用户程序。（你可能需要lapic.c中的cmostime函数），来读取实时时钟。Date.h里面定义了rtcdate结构，你需要把它作为一个指针变量放入cmostime

你需要创建一个用户程序，调用你的写的时间系统调用：



为了让你的date程序可以在shell上运行，在makefile中的UPROGS加上\_date.

你写日期系统调用的方法可以用在所有的系统调用上，比如uptime，你应该在所有的源代码中grep所有的uptime。使用

结束之后，可以在shell上面输入date，然后便会打印UTC时间。

**挑战：**写一个dup2系统调用