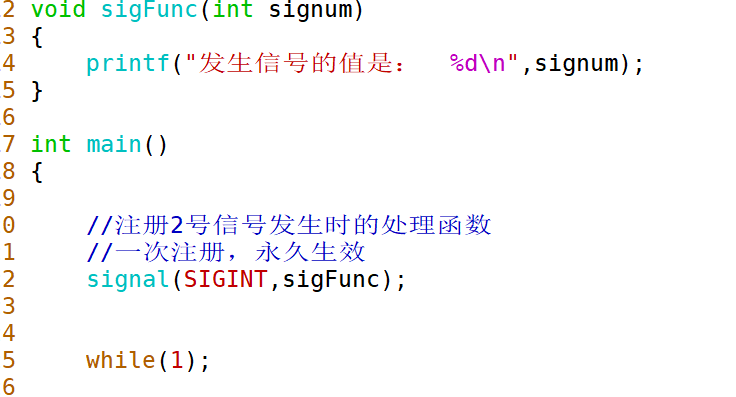
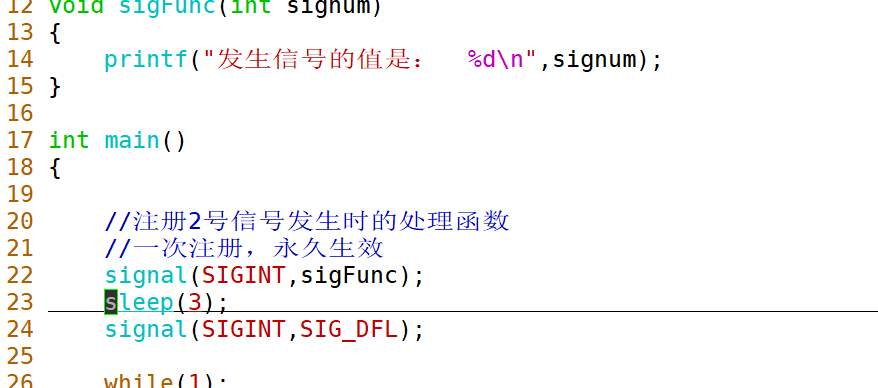
基础题

信号的处理

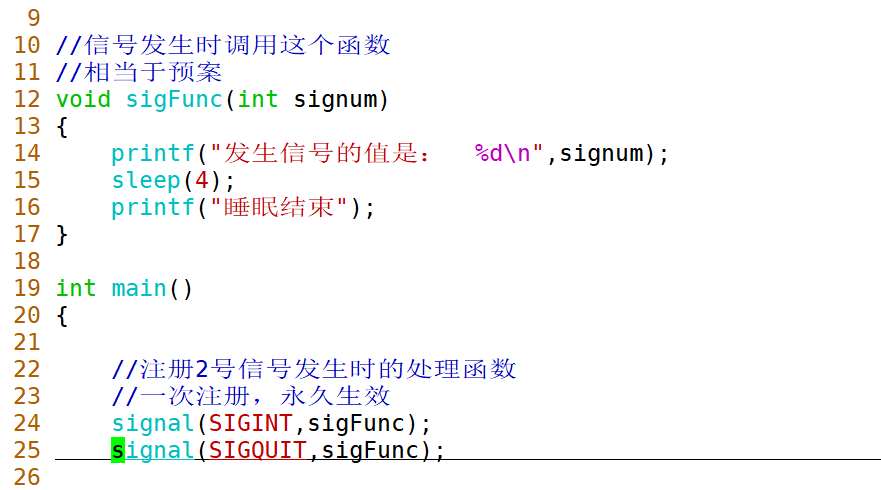
1、捕捉SIGIINT信号并设置信号处理函数，把默认的终止进程的处理行为改为打印，打印发生的信号的值。

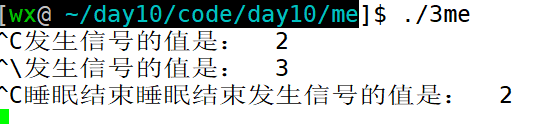
2、 在2题的基础上，再通过signal函数改回2号信号默认的处理行为（终止）。

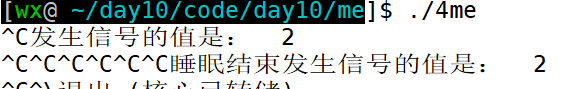
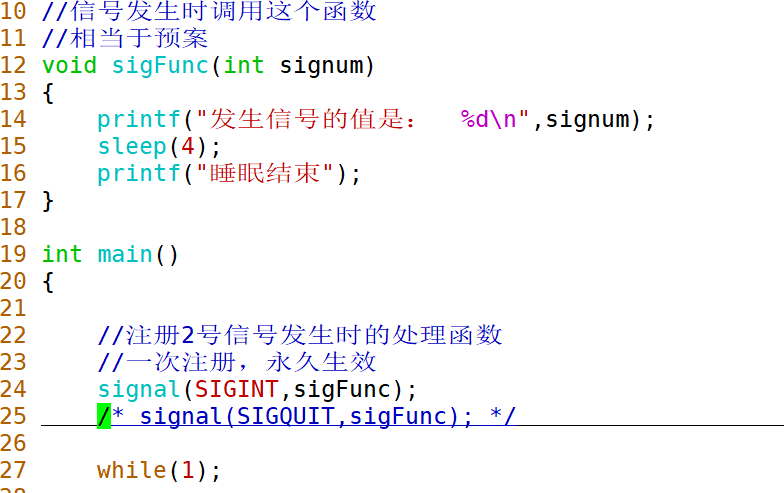
信号的打断



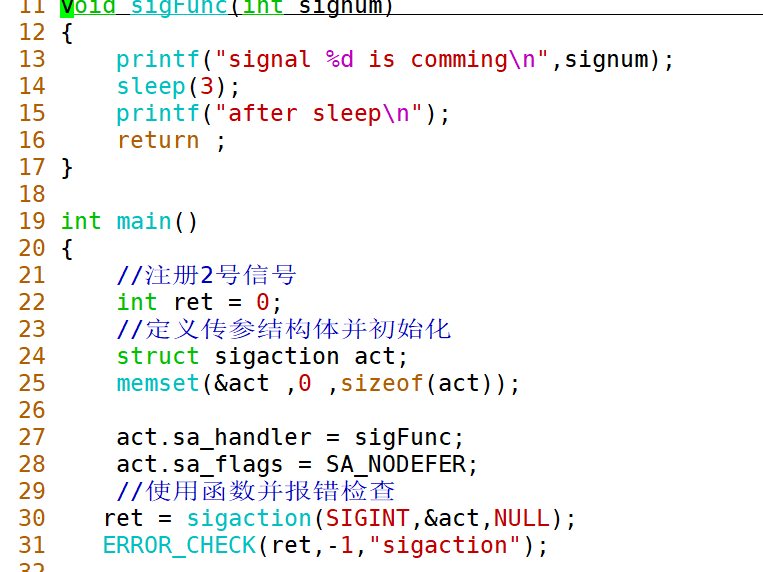
3、signal设置信号处理函数，演示不同信号到来时，后来的信号会打断原有信号的信号处理函数效果

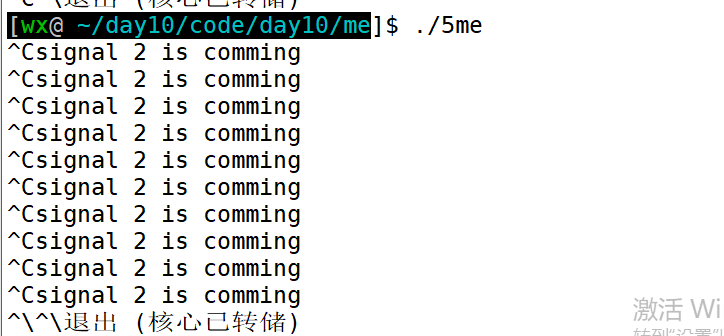




4、signal设置信号处理函数，演示相同信号到来时，无法打断本信号的效果

5、sigaction实现信号和信号之间不断重入的执行效果(信号之间随时可以互相打断的效果)。



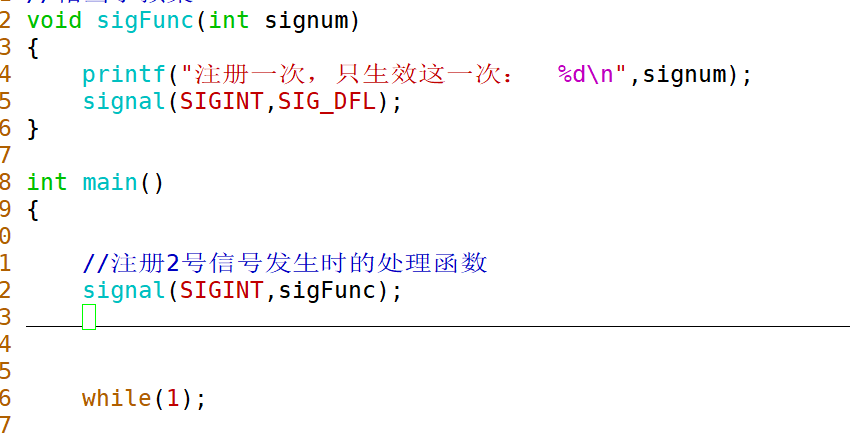


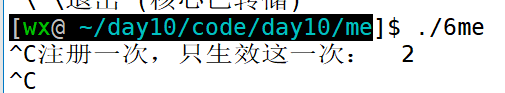
--------------------------------------------

难度题

1、思考一下，不用sigaction函数，就用signal，如何实现注册一次，就生效一次的效果，下一次需要重新注册（类似于sigaction里的SA\_RESETHAND）。

提示：根据前面的一题和二题综合思考一下。

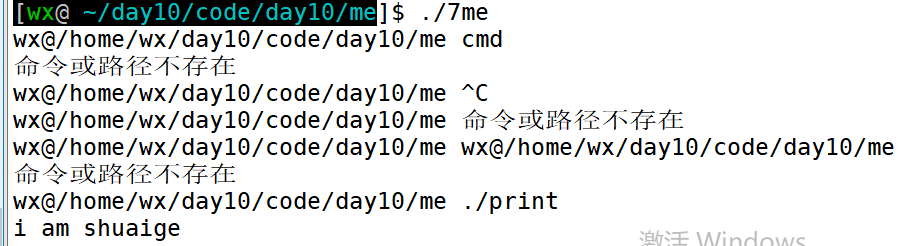
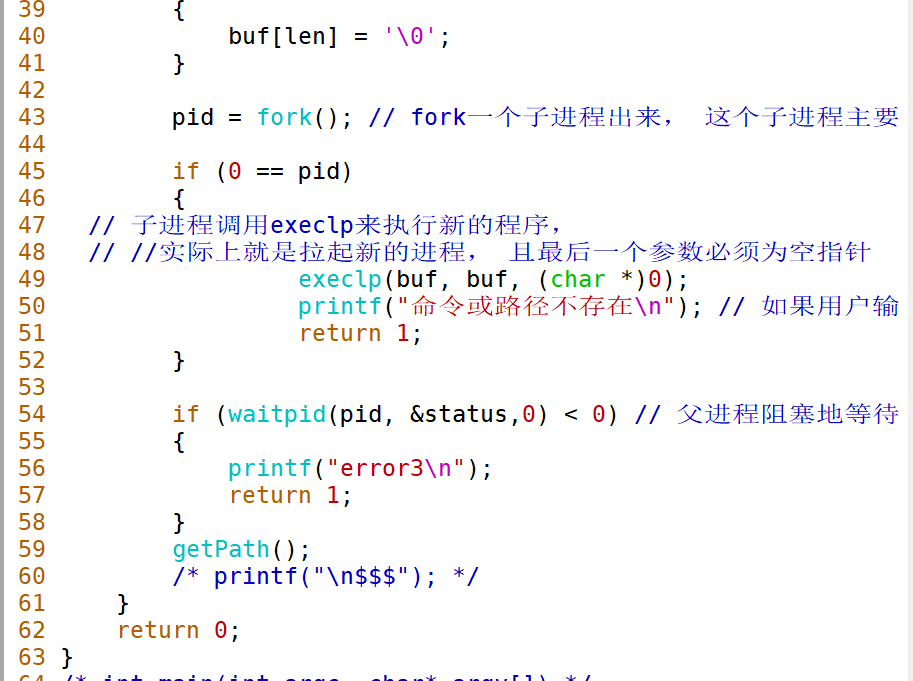




2、实现一个简单的bash程序，实现如下效果：

1： 程序启动时，打印一个用户名和路径（用户名和路径就是一个字符串，自己定义即可，类似于[ken@ubuntusigaction]$），2号信号到来时，打印一下当前的路径，每次输入回车时，也打印一下路径（跟bash一样）。

2：可以创建子进程执行其他程序，比如输入./print，会执行print程序，父进程等待（fork+execl），效果跟bash程序一样。



3: 其他内容自己添加，比如输入exit就退出(不要求必须完成，可以自行扩展)。



