6 调试步骤

6.1结果异常调试

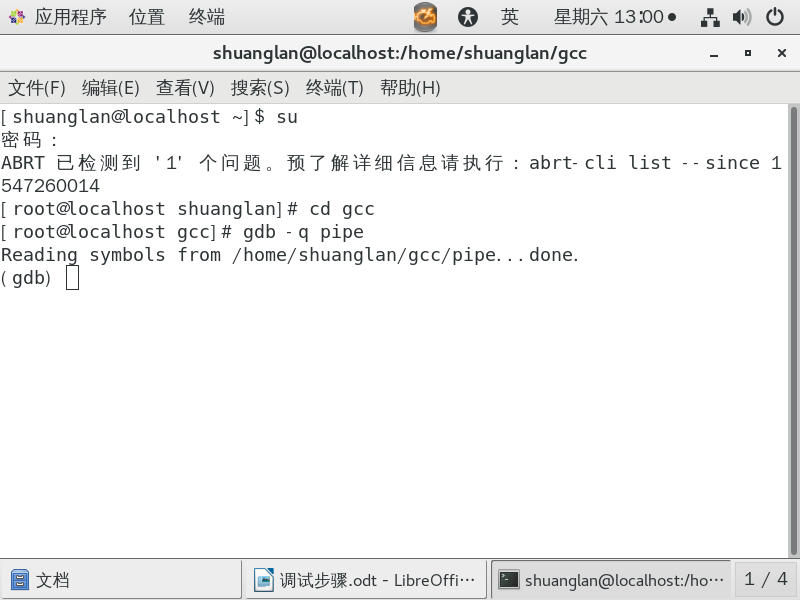
实验过程中，字符串“is sending a message to parent!”以字符串形式存储在数组中，由于最初仅为buf数组申请了30个字符空间，在程序执行时上述字符串被截断，末尾的“！”没有写入数组。程序运行后的输出结果为“is sending a message to parent”。分析了运行结果之后，我们为buf数组分配了50个字符空间，并把write、read函数中限制向管道中写入、读出的字符数量也改为50。再次运行程序，程序正确输出“is sending a message to parent!”。

6.2 gdb调试

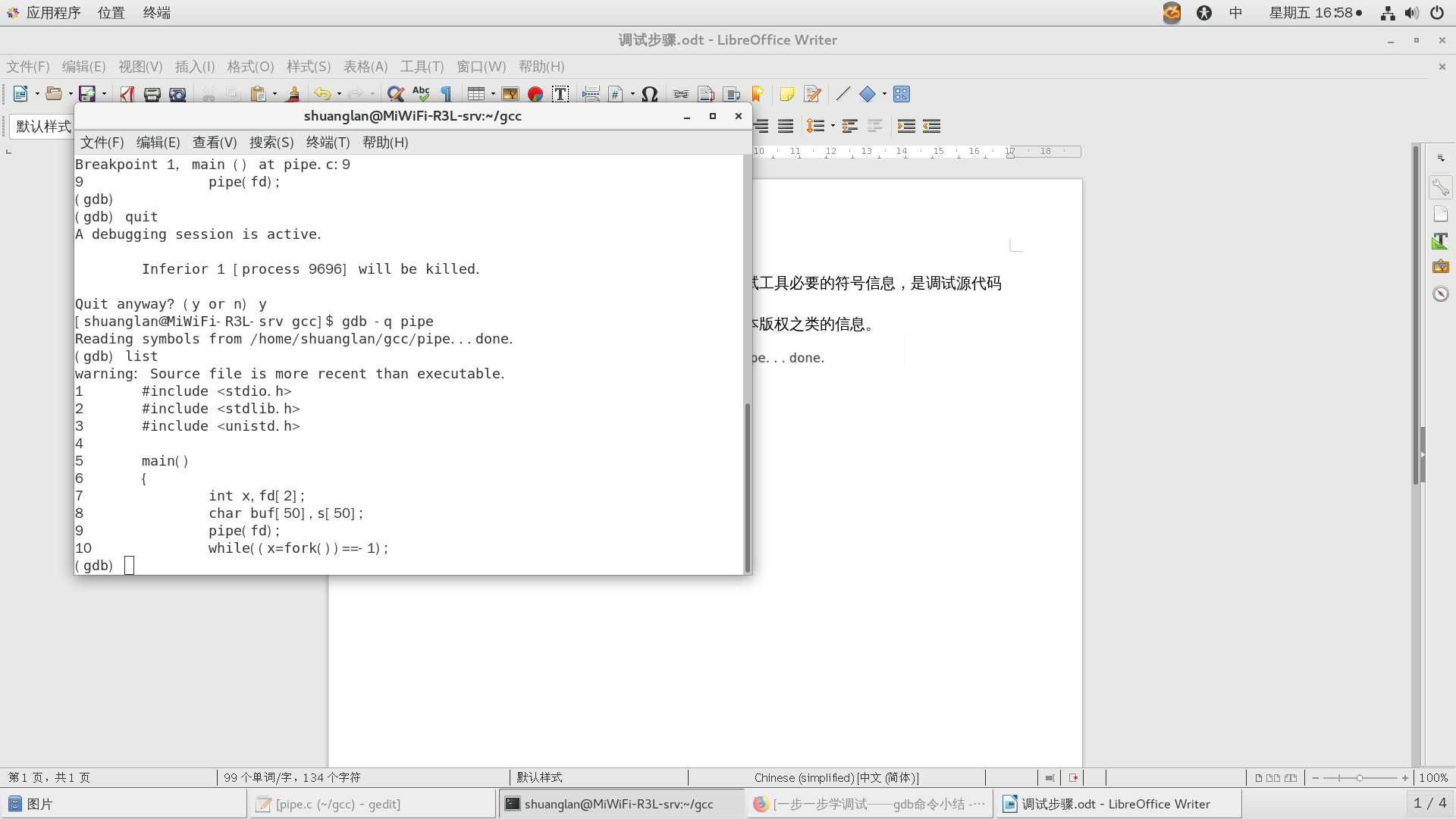
gdb调试的主要目的是通过程序的单步执行，观察父进程和子进程在Linux系统下的运行次序，分析其运行结果。

(gdb) gcc -g pipe.c -o pipe 参数-g用于产生符号调试工具必要的符号信息，是调试源代码必须的选项。

(gdb) gdb pipe 进入调试状态，参数-q表示不打印版本版权之类的信息[1]。

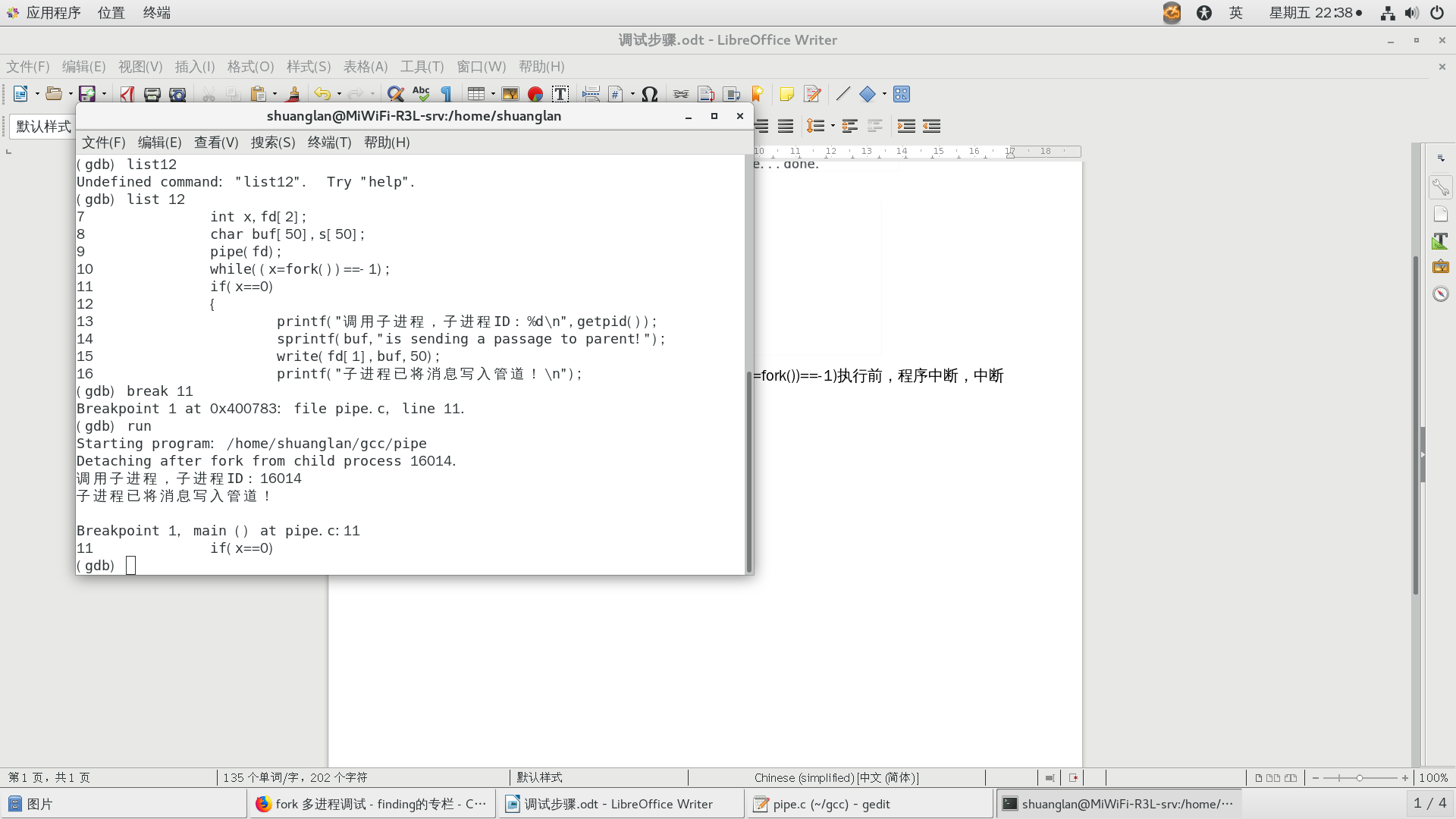


(gdb) list 显示源代码中10行内容。

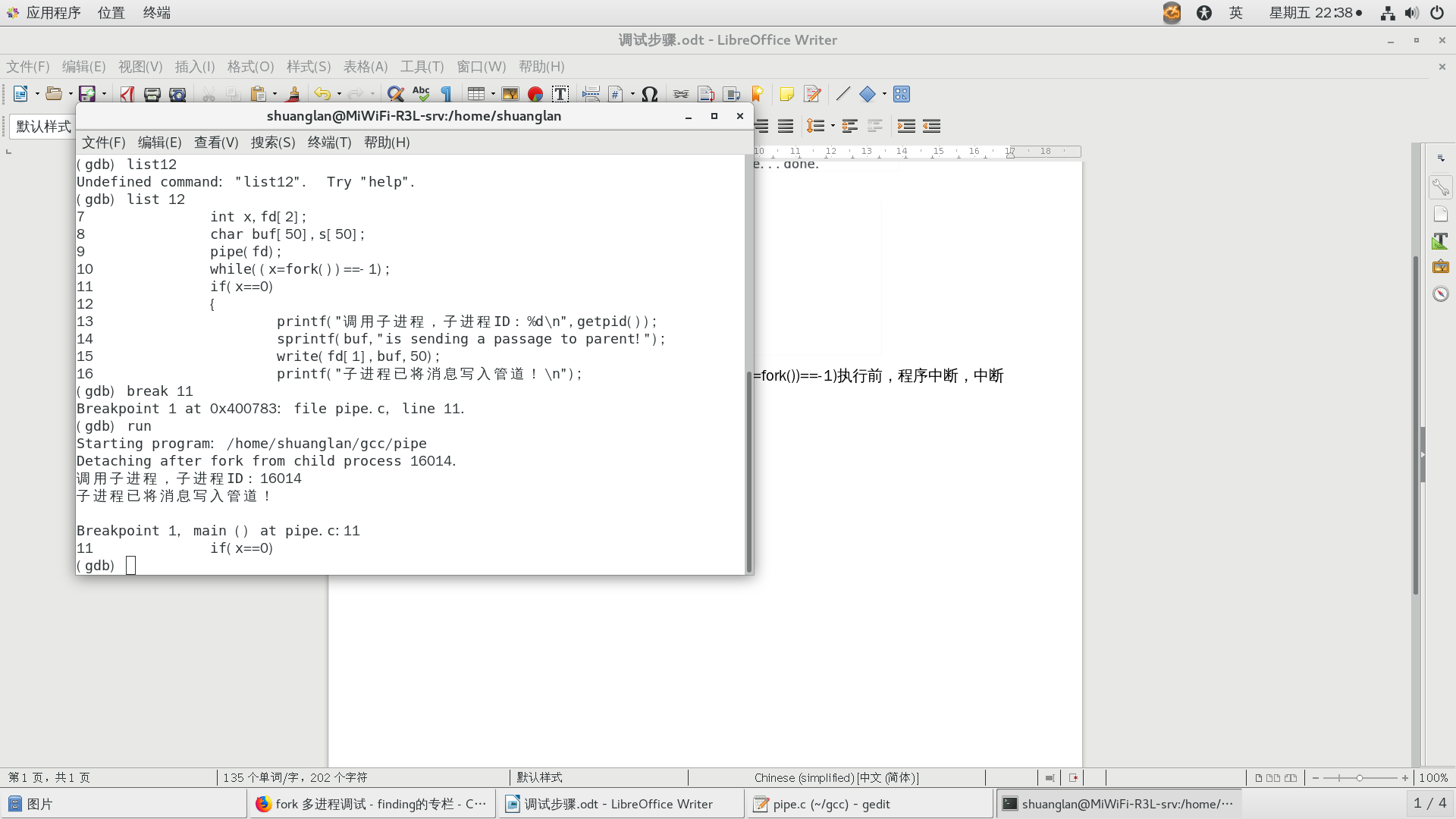


（1）子进程正常运行，调试父进程

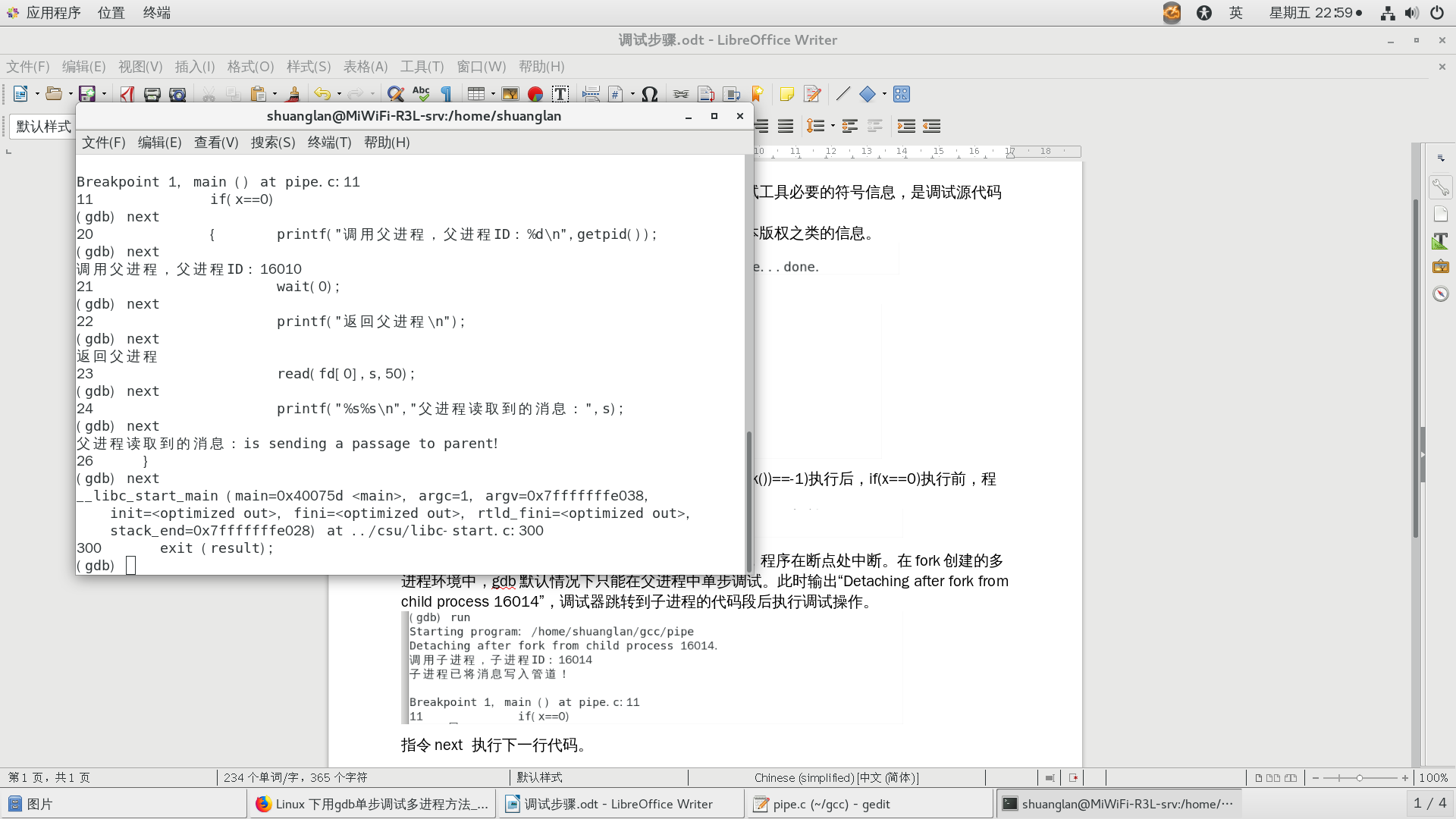
(gdb) break 11 在行号11处设置断点。while((x=fork())==-1)执行后，if(x==0)执行前，程序中断。



(gdb) run 运行程序，利用x=fork()成功创建子进程后，程序在断点处中断。在fork创建的多进程环境中，gdb默认情况下只能在父进程中单步调试。此时输出“Detaching after fork from child process 16014”，调试器跳转到子进程的代码段后执行调试操作[2]。



(gdb) next 执行下一行源代码中的一条指令。在父进程中按行执行，便于我们观察每一行代码的执行效果。父进程部分代码都能正确地执行，父进程运行无误。



(2)子进程处于暂停状态，调试父进程

set follow-fork-mode [parent | child]

用于设置被调试的进程，默认情况下是parent。若想调试子进程，将参数设置为child。

set detach-on-fork [on | off]

on：只调试父进程或子进程中的一个(根据follow-fork-mode来决定)，这是默认的模式。

off: 父子进程都在gdb的控制之下，其中一个进程正常调试(根据follow-fork-mode来决定),另一个进程会被设置为暂停状态[3]。

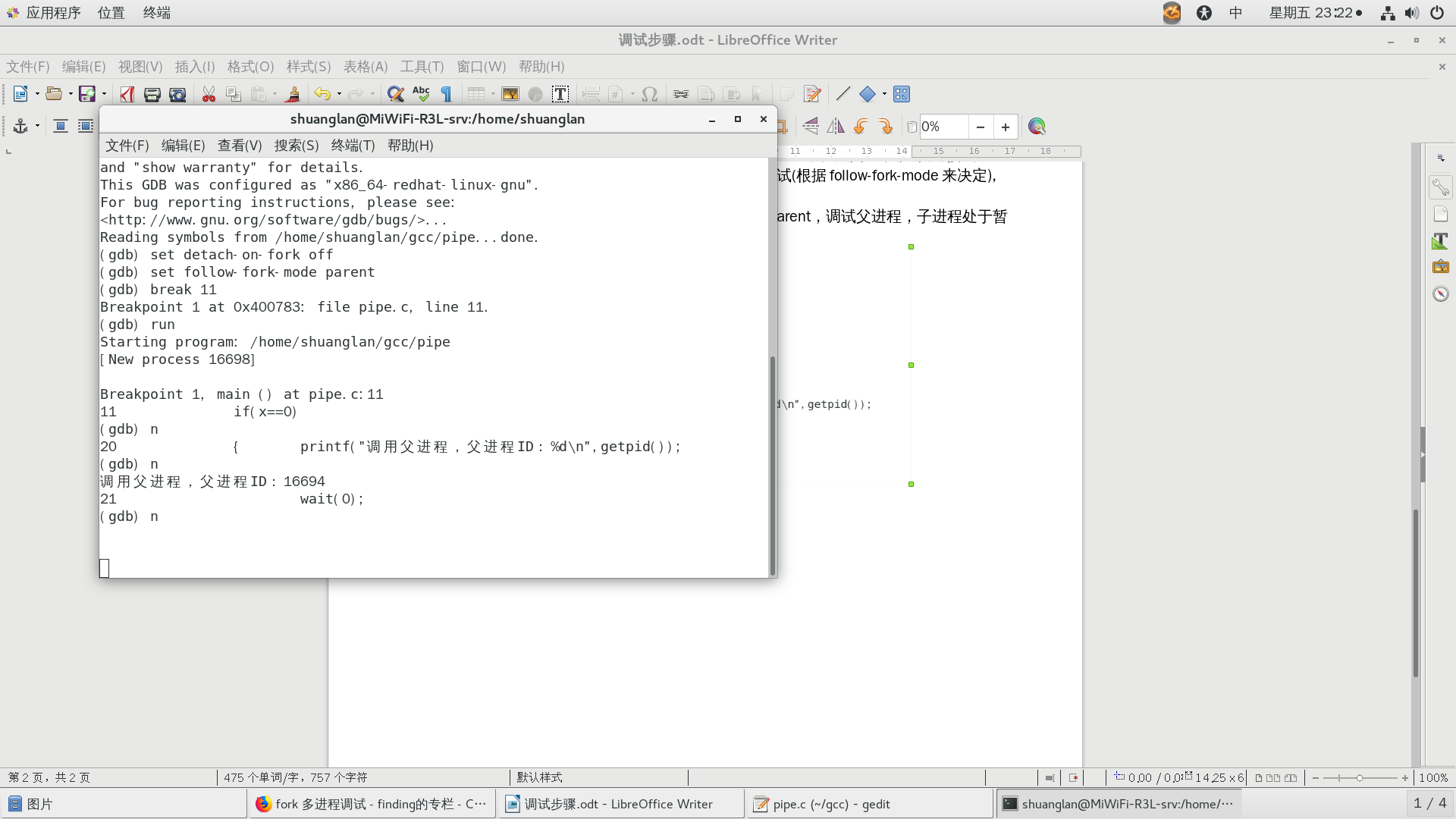
(gdb) set detach-on-fork off

(gdb) set follow-fork-mode parent

(gdb) break 11 设置断点，调试父进程，子进程处于暂停运行状态。

(gdb) run

(gdb) n 在断点后，单步执行程序。



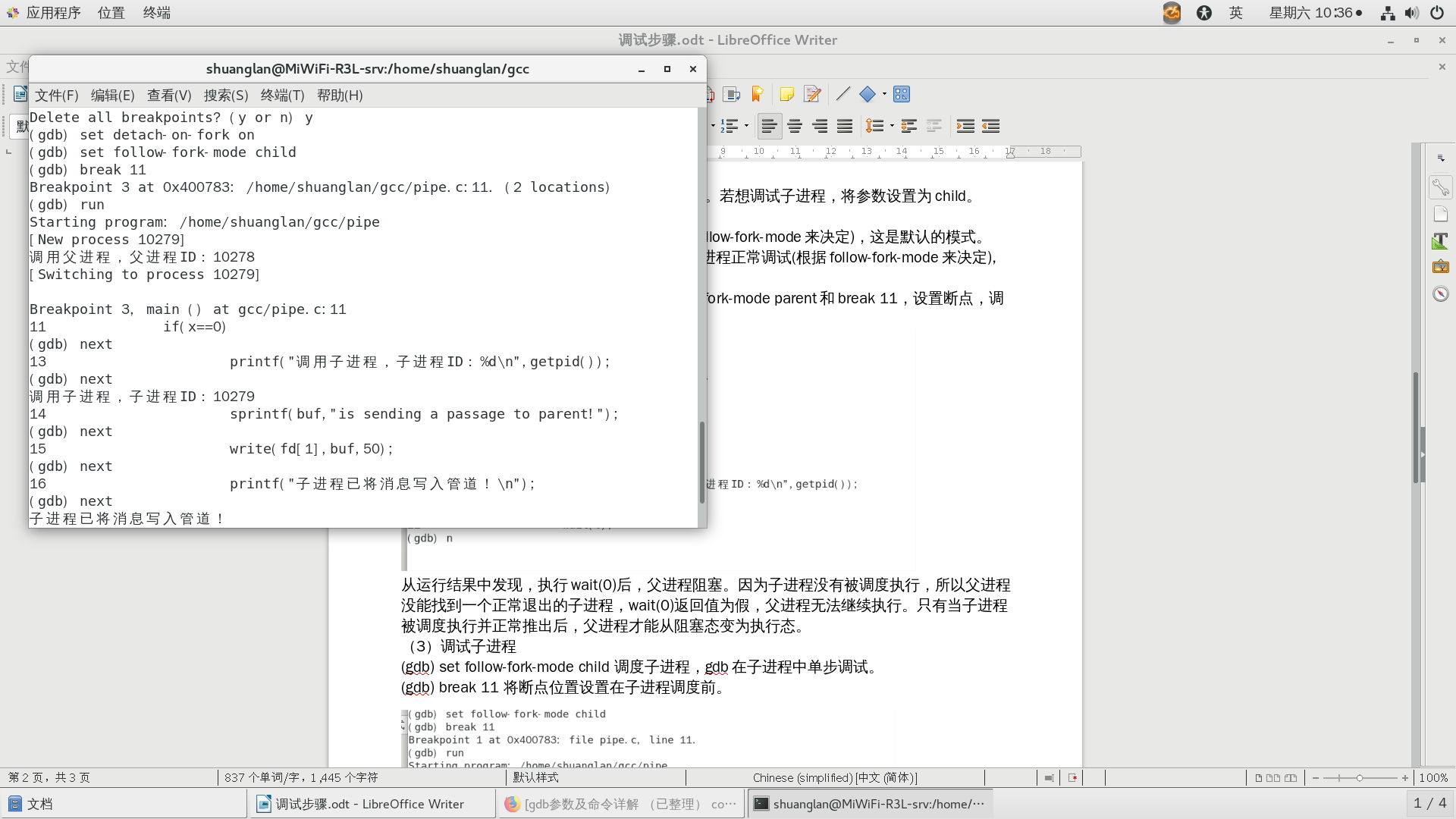
从运行结果中发现，执行wait(0)时，父进程阻塞。因为子进程没有被调度执行，所以父进程没能找到一个正常退出的子进程，wait(0)返回值为假，父进程无法继续执行。只有当子进程被调度执行并正常退出后，父进程才能从等待态变为运行态。

（3）父进程正常运行，调试子进程

(gdb) set detach-on-fork on 只调试子进程或父进程中的一个，不被调试的进程正常执行。

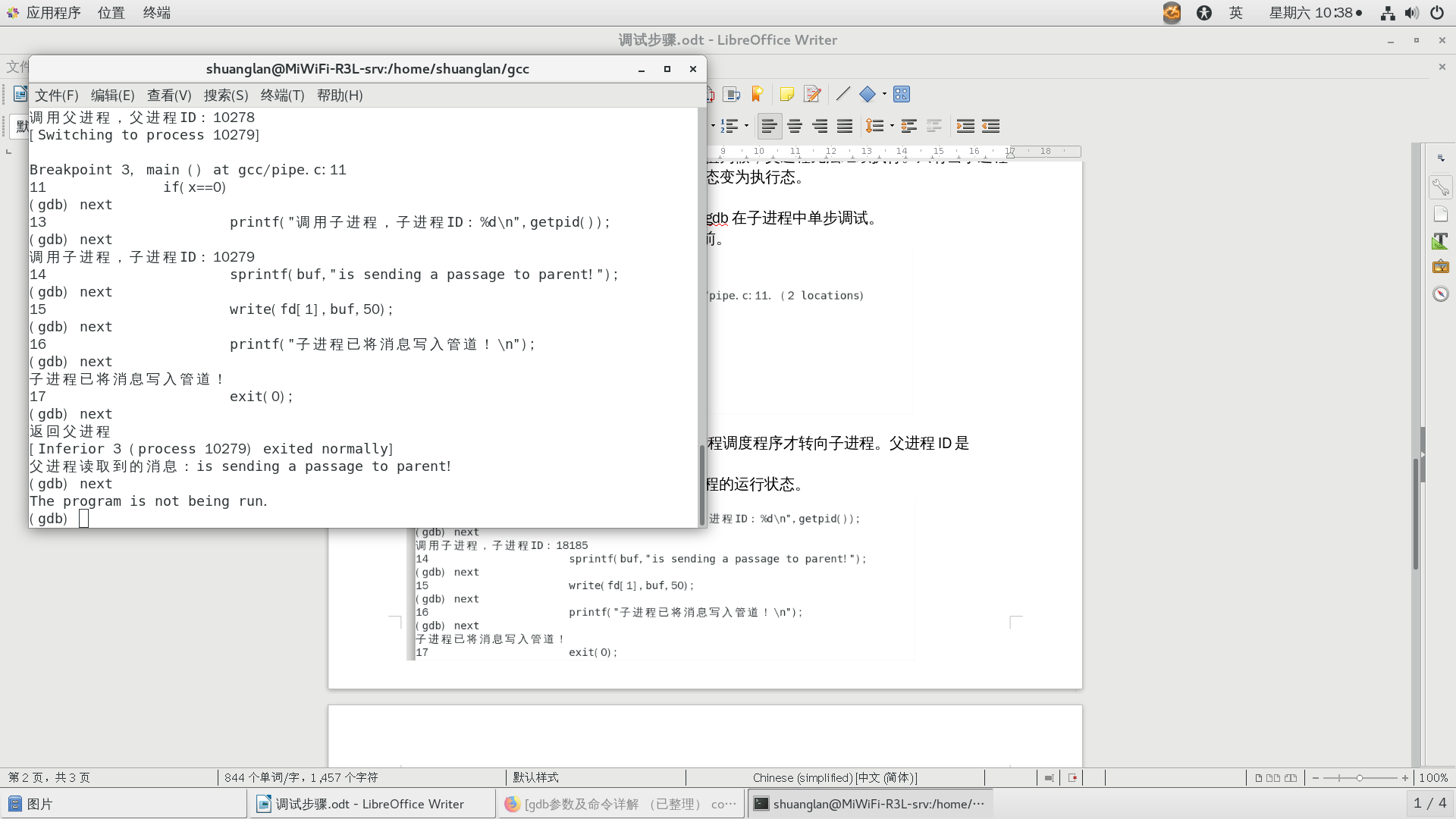
(gdb) set follow-fork-mode child 调试子进程，gdb在子进程中单步调试。

(gdb) break 11 将断点位置设置在子进程调度前。



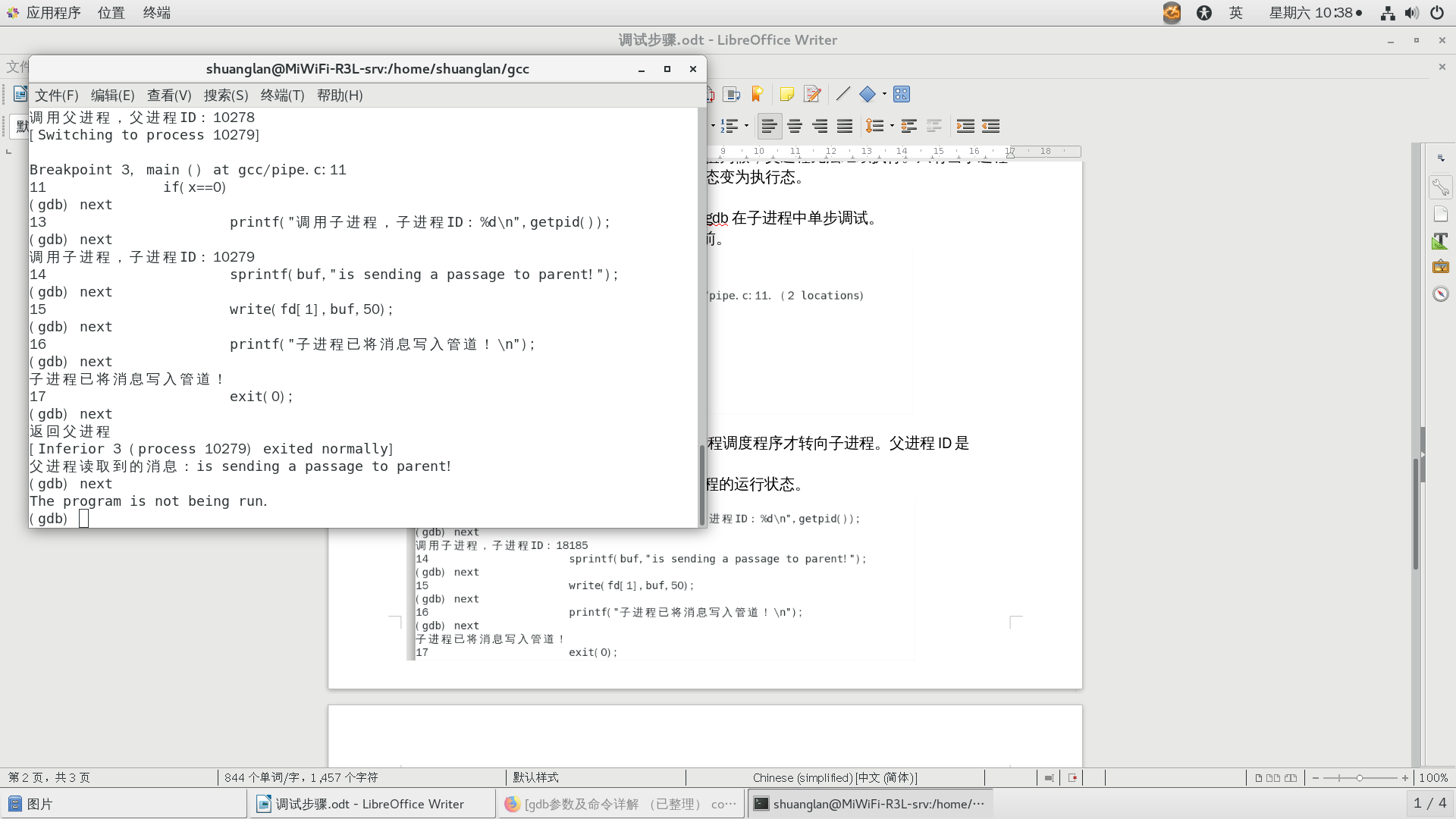
从运行结果看出，父进程在wait(0)阻塞后，进程调度程序才转向子进程。父进程ID是10278，子进程ID是10279。

(gdb) next 在子进程中单步执行，便于观察进程的运行状态。



从运行结果看出，子进程部分的代码均被正确执行，子进程运行无误。

(gdb) next 子进程执行exit(0)并正常退出，进程调度程序转向父进程。父进程在wait(0)处能顺利执行，父进程从等待态变为运行态，父进程正常运行直至结束。



（4）总结

在fork创建的多进程环境中，gdb调试器默认调试父进程，且一次只调试一个进程。当我们想调试子进程时，要用到follow-fork-mode和detach-on-fork方法。通过对三次调试结果的对比和分析，我们发现在Linux系统中，进程调度程序优先调度父进程，只有在父进程阻塞时，才调度子进程；当子进程运行结束并删除后，进程调度程序再转向父进程。

参考文献

[1] gdb参数及命令详解 （已整理） core dump调试

https://blog.csdn.net/ghostyu/article/details/8083228

[2]【Linux】关于理解fork()函数的简单例子https://blog.csdn.net/ww1473345713/article/details/51708003

[3] fork 多进程调试 https://blog.csdn.net/fingding/article/details/46459095