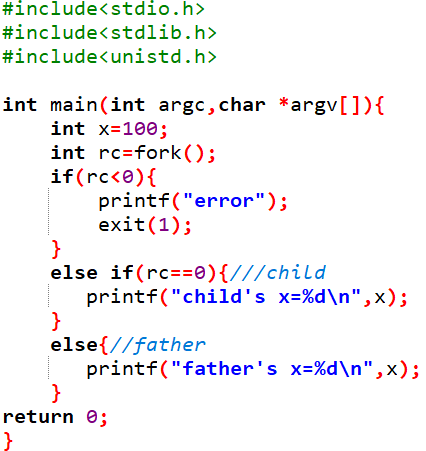
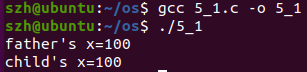
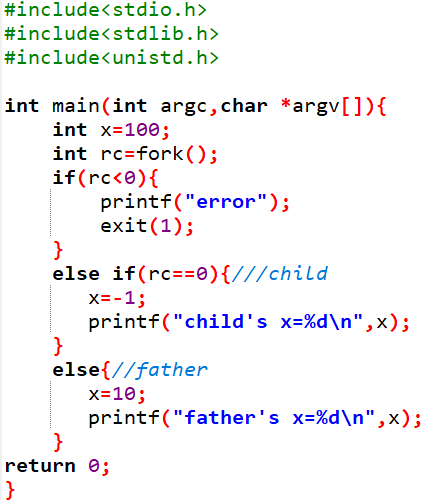
# 5.1

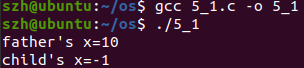
编写一个调用fork()的程序。在调用fork()之前，让主进程访问一个变量（例如x）并将其值设置为某个值（例如100）。子进程中的变量有什么值？当子进程和父进程都改变x的值时，变量会发生什么？





两者返回结果都是100

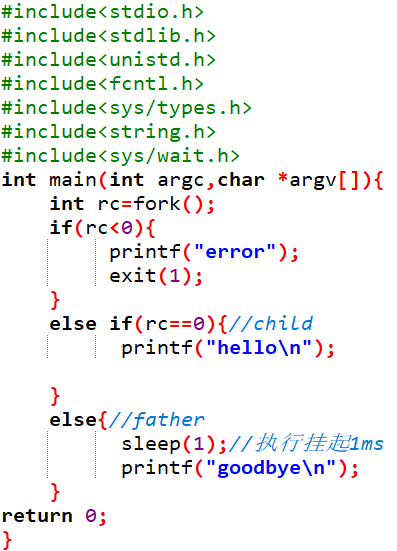


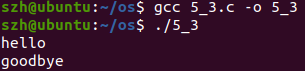


修改之后则各自变成各自修改后的的x

# 5.3

使用fork()编写操一个程序。子进程应打印“hello”，父进程应打印“goodbye”。你应该尝试确保子进程始终先打印。你能否不在父进程调用wait()而做到这一点呢？

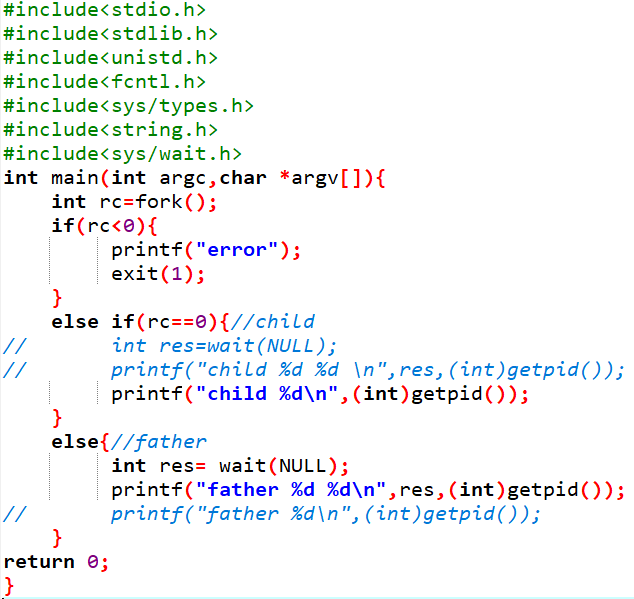


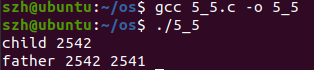


程序执行时会先打印hello，一段时间后goodbye被打印出

# 5.5

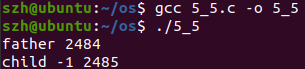
现在编写一个程序，在父进程中使用wait()，等待子进程完成。wait()返回什么？如果你在子进程中使用wait()会发生什么？





父进程等待子进程完成，返回子进程的pid

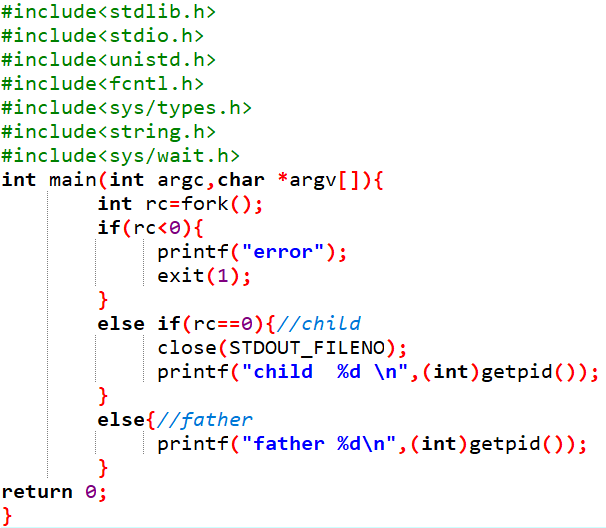




子进程等待父进程完成，返回的是-1

# 5.7

编写一个创建子进程的程序，然后在子进程中关闭标准输出（STDOUT\_FILENO）。如果子进程在关闭描述符后调用printf()打印输出，会发生什么？





只会输出父程序的printf