Q:

1. 解释 `wc` 和 `grep` 指令的含义。
2. 解释 `ps aux | grep firefox | wc -l` 的含义。
3. `echo aaa | echo bbb | echo ccc` 是否适合做shell实验中管道符的检查用例？说明原因。
4. 对于匿名管道，如果写端不关闭，并且不写，读端会怎样？
5. 对于匿名管道，如果读端关闭，但写端仍尝试写入，写端会怎样？
6. 假如使用匿名管道从父进程向子进程传输数据，这时子进程不写数据，为什么子进程要关闭管道的写端？
7. fork之后，是管道从一分为二，变成两根管道了吗？如果不是，复制的是什么？
8. 解释系统调用 `dup2` 的作用。
9. 什么是shell内置指令，为什么不能fork一个子进程然后 `exec cd` ？
10. 为什么 `ps aux | wc -l` 得出的结果比 `get_ps_num` 多2？
11. 进程名的最大长度是多少？这个长度在哪定义？
12. `task_struct` 在Linux源码的哪个文件中定义？
13. 为什么无法通过 `SYSCALL_DEFINEx` 定义二维数组（如 `char (*p)[50]` ）为参数？
14. 在修改内核代码的时候，能用 printf 调试吗？如果不能，应该用什么调试？
15. `read()` 、 `write()` 、 `dup2()` 都能直接调用。现在我们已经写好了一个名为 `ps_counter` 的系统调用。为什么我们不能在测试代码中直接调 `ps_counter()` 来调用系统调用？

A:

1. wc： 统计字数；grep：筛选并高亮指定字符串。
2. 显示进程状态、筛选含有firefox的进程、计算列数：显示与firefox相关的进程数。
3. 不适合 echo本身不接收输入 前两个echo的结果不会显示。
4. 阻塞，直到有数据写入时才继续。
5. 写进程收到信号 `SIGPIPE` 导致进程异常终止

6. 假如一条管道有多个写端，那么只有在所有写端都关闭之后（管道的引用数降为0），读端才会解除阻塞状态。
7. 不是；复制了pipefd数组。
8. `int dup2(int oldfd, int newfd);` 该函数相当于将 `newfd` 标识符变成 `oldfd` 的一个拷贝，与 `newfd` 相关的输入/输出都会重定向到 `oldfd` 中。如果 `newfd` 之前已被打开，则先将其关闭。
9. 就是由Bash自身提供的命令，而不是文件系统中的某个可执行文件。管道中的内置命令也是在新的子进程中运行的，不会改变当前进程（shell）的状态。
10. 多了一个ps aux进程和表头
11. 16 在sched.h中定义
12. sched.h
13. 宏定义要把类型和名字分开，p[50]能当作类型名
14. 不能；用 `printk()`
15. glibc库没有封装这个系统调用，就没办法通过使用封装好的API来调用该系统调用。应该使用glibc提供的 syscall 库函数