

Lab1

姓名: 陈锐林, 学号:21307130148

2023 年 9 月 16 日

一、代码:

```
1 // pingpong.c
2 #include "kernel/types.h"
3 #include "kernel/stat.h"
4 #include "user/user.h"
5 int main() {
6     int pp[2]; //管道
7     int pid;
8
9     if (pipe(pp) == -1) {
10         printf("pipe error"); //if pipe失败
11         exit(1);
12     }
13     pid = fork(); // 创建子进程
14     if (pid == -1) {
15         printf("fork error"); //if fork失败
16         exit(1);
17     }
18     if (pid == 0) { //子进程
19         char buf;
20         // 子进程等待接收数据
21         read(pp[0], &buf, 1);
22         printf("%d: received ping\n", getpid());
23         // 子进程写回给父进程
24         write(pp[1], &buf, 1);
25         // 子进程退出
26         exit(0);
27     } else { // 父进程
28         char buf = '!'; // 父进程要发送的数据
```

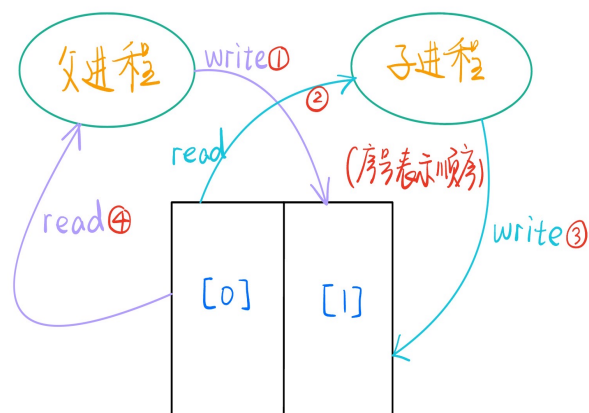
```

29 // 父进程向管道中写入数据，传递给子进程
30 write(pp[1], &buf, 1);
31 // 父进程等待从子进程接收数据
32 read(pp[0], &buf, 1);
33 printf("%d: received pong\n", getpid());
34 // 父进程退出
35 exit(0);
36 }
37 return 0;
38 }

```

二、实验思路:

这个实验比较简单，思路也很清晰。根据题目要求，我们要利用一个管道在父进程和子进程之间传递数据 (示意图如下)。大致步骤如下：pipe 函数进行管道创建，根据 fork 返回的 pid 执行不同操作；父进程向管道中写入数据传递给子进程；子进程接收数据，并且发送回父进程。



三、测试结果:

```

== Test pingpong ==
$ make qemu-gdb
pingpong: OK (0.9s)

```

经测试，0.9s 可完成一次，即一秒钟内能完成大概 1.11 次该操作。