## 《Linux 操作系统设计实践》实验四：文件操作

实验环境：Ubuntu 14.04.5 LTS

实验内容：

运行代码：

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define FLAGS O\_WRONLY | O\_CREAT | O\_TRUNC

int main() {

char buf1[] = {"Hello "};

char buf2[] = {"World!"};

int fd = 0, count = 0;

const char \*path = "test";

if ((fd = open(path, FLAGS, 0600)) == -1) {

printf("Open file failed!\n");

exit(1);

}

count = strlen(buf1);

if (write(fd, buf1, count) != count) {

printf("Write file failed!\n");

exit(1);

}

system("cat test");

printf("\n");

if (lseek(fd, 10, SEEK\_SET)==-1) {

printf("lseek failed!\n");

exit(1);

}

count = strlen(buf2);

if (write(fd,buf2,count)!=count) {

printf("write file failed!\n");

exit(1);

}

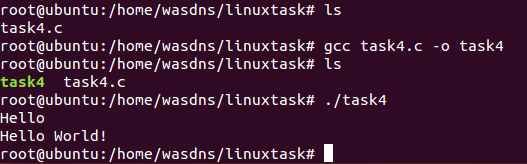
system("cat test");

printf("\n");

return 0;

}

运行结果：



实验总结：

本次实验实现是简单的，参考课件，通过对文件读写等操作很容易实现实验目的，如果要结合第二次和第三次实验的话，只用把他们之间联系起来，比如当server读入数据时，同时将数据存入文件中去，client就可以从文件中直接读取数据。这里就没写了 如果需要代码的话，参考下实验二和实验三的代码，再加上实验四（也就是上面的代码），往输入输出的代码中增加文件的操作，实现保存信息的发送接收，道理是一样的。