[MIT 6.S081 Lecture Notes | Xiao Fan (樊潇) (fanxiao.tech)](https://fanxiao.tech/posts/MIT-6S081-notes/)

ZachVec老哥的代码实现:MIT-6.S081上

**xv6笔记**

## 添加xv6代码：

在XV6的user中编写c语言代码（如copy.c），之后在user上一级目录中找到MakeFile文件并编辑，将编写的c语言文件名添加到该文件中，之后重新启动qemu，即可运行开始的程序。

## 退出xv6：

ctrl+a，回到qemu的monitor界面，之后q+回车退出qemu；

ctrl+a+x，直接退出。

## xv6代码头文件

#include “kernel/types.h”

#include “user/user.h”

#include “kernel/fcntl.h”

## xv6常用函数

**main函数**：**int main(int argc, char\* argv[ ])**，argc为总的参数个数，argv[i]为字符串（用户输入），其中argv[0]为程序名，argv[i]为用户输入的参数。参考网站：[C语言中 int main(int argc,char \*argv[])的两个参数详解\_Jeff\_的博客-CSDN博客](https://blog.csdn.net/weixin_40539125/article/details/82585792)

**exec(filename,\*argv):** 加载并执行一个文件，常用于在子程序中执行另一个程序；

fork(): 可以用来创建子进程，对于父进程返回子进程的pid(process identifier)，对于子进程返回0；

exit(): 退出当前进程；

wait(): 等待子进程结束；

kill(pid): 结束pid所指进程；

read(fd, buf, n); 从fd中读取最多n个字节，并将其拷贝到buf中，返回读取的字节数。

write(fd, buf, n)：写buf中的n个字节到fd中，返回实际写入到buf中的字节数

**pipe():** pipe(int p[2])创建一个管道，并将管道的读写端赋予到p中去，其中表示p[0]表示读端（即进程写入管道），p[1]表示写端（管道的内容写入进程）；pipe实际为一个内核缓冲区，可以实现进程之间的通信，如父进程与子进程之间的通信。

get (pid) 获得当前进程pid；

sleep(n) 睡眠n秒；

sbrk(n) 为进程空间增加n个字节；

open(filename, flags) 打开文件，flag指定读写模式；

close(fd) 关闭打开的fd（file descriptor 文件描述符）；

**dup(fd)** 复制fd；

chdir(dirname) 改变当前目录；

mkdir(dirname) 创建新的目录；

mknod(name, major,minor) 创建设备文件；

fstat(fd) 返回文件信息；

link(f1, f2) 给f1创建一个新的名字f2；

unlink(filename) 删除文件；

exit(): exit(1) 异常退出，exit(0) 正常退出

I/O文件描述符：

**0：标准输入，1：标准输出；2：标准错误输出**