

使用 Ubuntu（虚拟机）

实验进度：除了分屏没有做，其他都已完成

下面是各个必做题或思考题的答案

❓ Why executing the "poweroff" command requires superuser privilege in some Linux distributions?

Can you provide a scene where bad thing will happen if the `poweroff` command does not require superuser privilege?

比如几个用户同时使用一台设备，若其中一个用户成功执行 `poweroff` 指令，其他人会失去未保存的内容

发现：Root 权限也有做不了的更改

`ls -li` 查找文件的 inode 号。`ls -l` 显示文件的硬链接数量。

`find / -inum xxxxxx` (inode) 查找硬链接，这一步即使有 root 权限也做不到

```
root@scs-virtual-machine:~# sudo find / -inum 931421
find: '/run/user/1000/gvfs': Permission denied
find: '/run/user/1000/doc': Permission denied
/home/scs/ics2024/Makefile
```

经搜索，可能与 FUSE（用户空间文件系统）技术有关，但我还没找到具体原因

Write a "Hello World" program under GNU/Linux

```
scs@scs-virtual-machine:~$ cat hello.c
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

Write a Makefile to compile the "Hello World" program

```
scs@scs-virtual-machine:~/Cproject$ cat Makefile
# Makefile for Hello World program

# Compiler
CC = gcc

# Compiler flags
CFLAGS = -g -Wall

# Target executable name
TARGET = hello

all: $(TARGET)

$(TARGET): hello.o
    $(CC) $(CFLAGS) -o $(TARGET) hello.o

hello.o: hello.c
    $(CC) $(CFLAGS) -c hello.c

clean:
    rm -f *.o $(TARGET)
```

当执行 make 时,

- 首先, 它会检查 all 目标, 这是默认目标, 它依赖于\$(TARGET), 即 hello。
- 接着, make 会查找如何构建 hello 的规则, 它会发现依赖于 hello.o。
- 然后, make 会查找如何构建 hello.o 的规则, 它会发现依赖于 hello.c。
- make 会依次执行生成 hello.o 和链接成 hello 可执行文件的命令。

分别是 gcc -g -Wall -c hello.c 和 gcc -g -Wall -o hello hello.o

Learn to use GDB

gdb hello 针对可执行程序 hello 进行分析

(gdb) next 逐条测试

进行到结束, 若仍使用 next 指令, 会报错

```
(gdb) next
__libc_start_call_main (main=main@entry=0x555555555149 <main>, argc=argc@entry=1
, argv=argv@entry=0x7fffffffd88) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
```

在 github 上添加 ssh key

在 ~/.ssh/id_rsa.pub 存放具体密钥，再把密钥复制到仓库

ssh -T git@github.com 测试是否连接成功

