实 验 报 告

课程名称： 操作系统实验

学 院： 计算机科学与工程学院

专 业： 软件工程 班 级： 软件18-1班

姓 名：KAFLE SAMRAT学 号： 201801060933

山 东 科 技 大 学

实 验 报 告

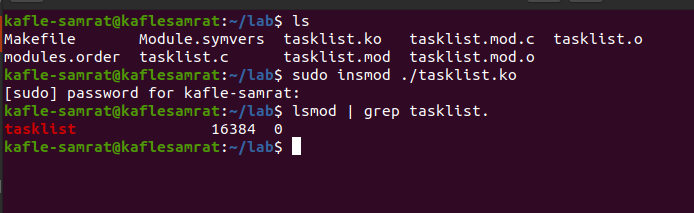
页

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组 别 |  | 姓 名 | KAFLE SAMRAT | 同组实验者 |  |
| 实验项目  名称 | Proc 文件系统及查看进程信息 | | 实验日期 |  | |
| 教师评语 |  | | | | |
| 实验成绩： | | | 指导教师（签名）：  年 月 日 | | |

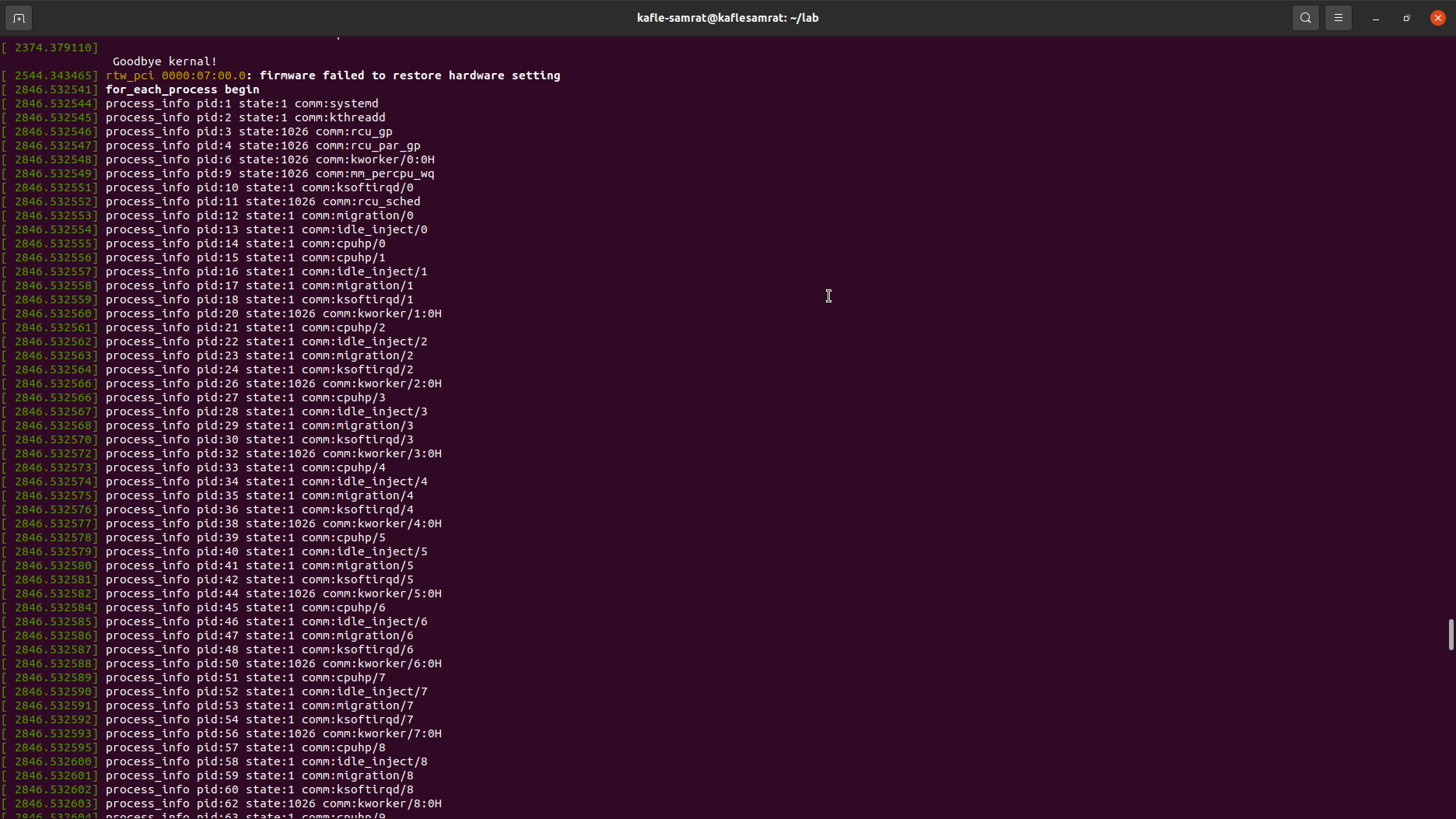
实验的目的：

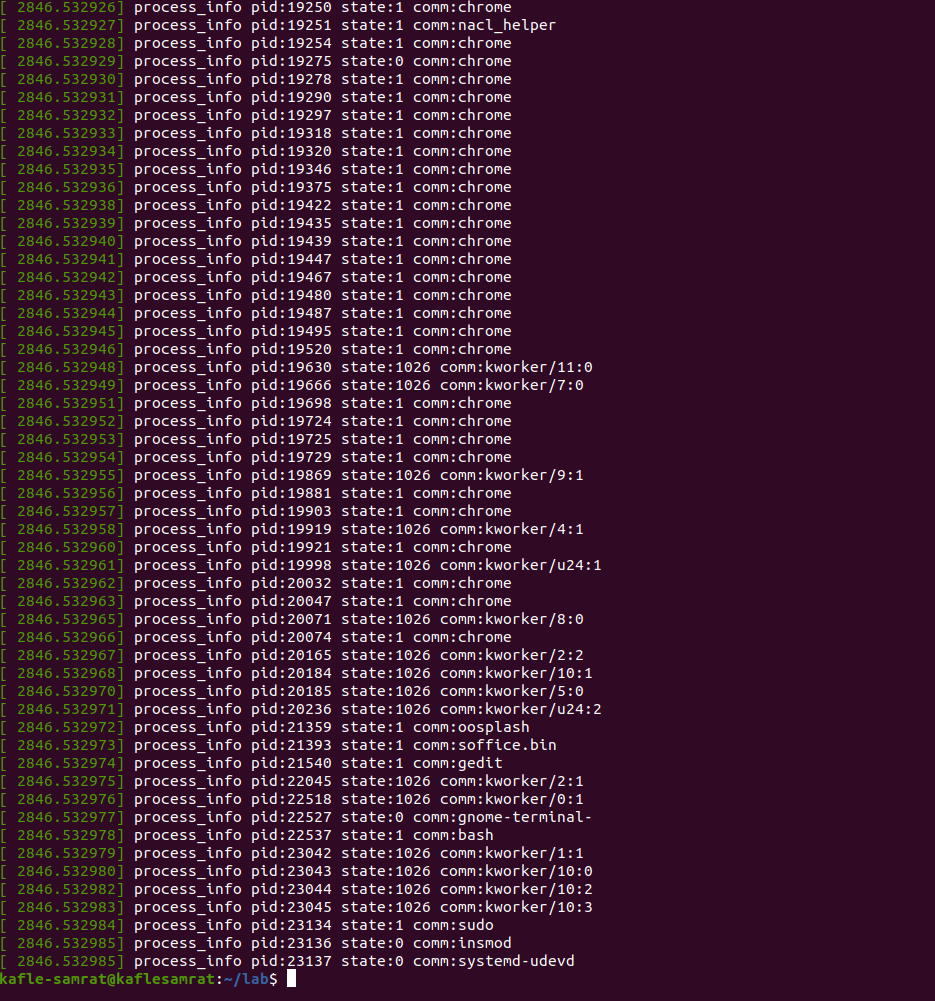
简单了解如何在Linux下使用PROC文件系统来获取进程信息。通过PROC文件系统获取的信息主要是进程使用的虚拟内存，以及Linux下的实际内存、信号机制信息等监控工具，可以全面掌握系统的运行情况。

实验的内容：

.

sudo dmesg

.

.

code : tasklist.c

#include <linux/kernel.h>

#include <linux/module.h>

#include <linux/proc\_fs.h>

#include <linux/sched/signal.h>

#include <linux/init.h>

static int \_\_init hello\_init(void)

{

struct task\_struct \*pp;

printk("for\_each\_process begin\n");

for\_each\_process(pp)

{

printk(KERN\_INFO "process\_info pid:%i state:%lu comm:%s \n",pp->pid,pp->state,pp->comm);

}

return 0;

}

static void \_\_exit hello\_exit(void)

{

printk("for\_each\_process end!\n");

}

module\_init(hello\_init);

module\_exit(hello\_exit);

MODULE\_LICENSE("GPL");

mod.c

#include <linux/build-salt.h>

#include <linux/module.h>

#include <linux/vermagic.h>

#include <linux/compiler.h>

BUILD\_SALT;

MODULE\_INFO(vermagic, VERMAGIC\_STRING);

MODULE\_INFO(name, KBUILD\_MODNAME);

\_\_visible struct module \_\_this\_module

\_\_section(.gnu.linkonce.this\_module) = {

.name = KBUILD\_MODNAME,

.init = init\_module,

#ifdef CONFIG\_MODULE\_UNLOAD

.exit = cleanup\_module,

#endif

.arch = MODULE\_ARCH\_INIT,

};

#ifdef CONFIG\_RETPOLINE

MODULE\_INFO(retpoline, "Y");

#endif

MODULE\_INFO(depends, "");

MODULE\_INFO(srcversion, "F79FE6886324425D89377F9");

make file :

obj-m := tasklist.o

KVERSION = $(shell uname -r)

KERNELDR :=/lib/modules/$(KVERSION)/build

PWD := $(shell pwd)

modules:

$(MAKE) -C $(KERNELDR) M=$(PWD) modules

moduels\_install:

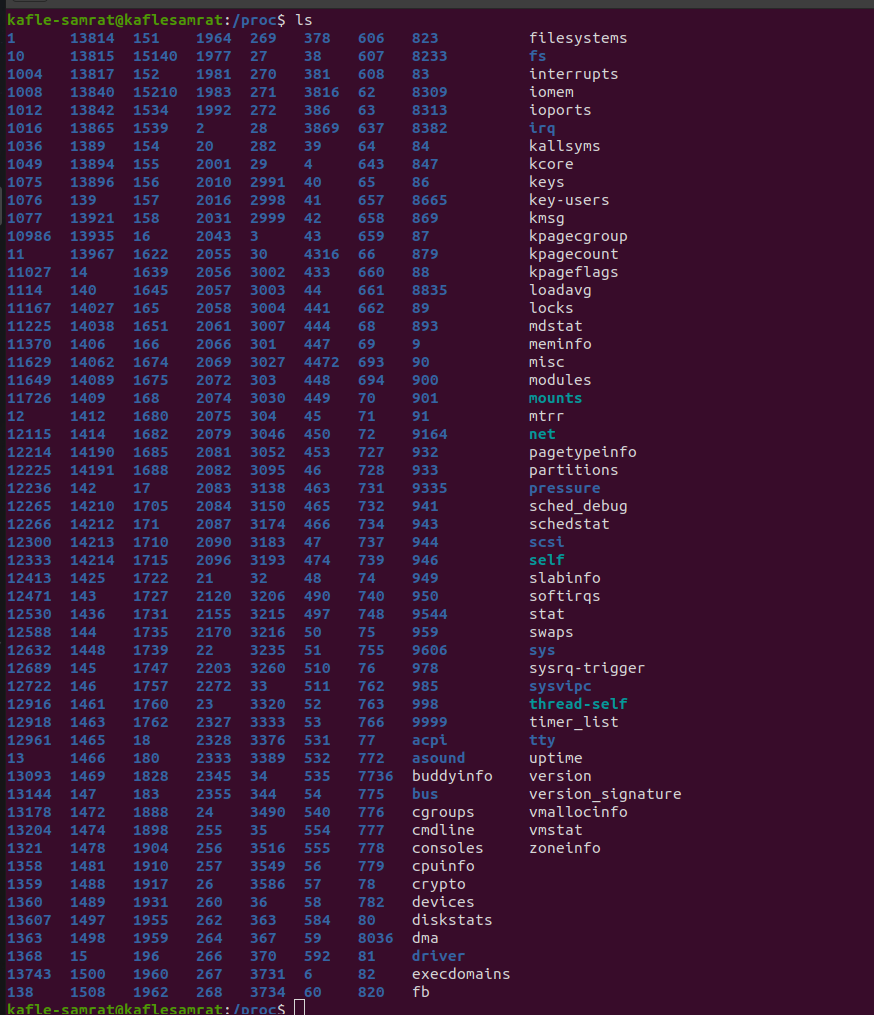
$(MAKE) -C $(KERNELDR) M=$(PWD) modules\_install

clean:

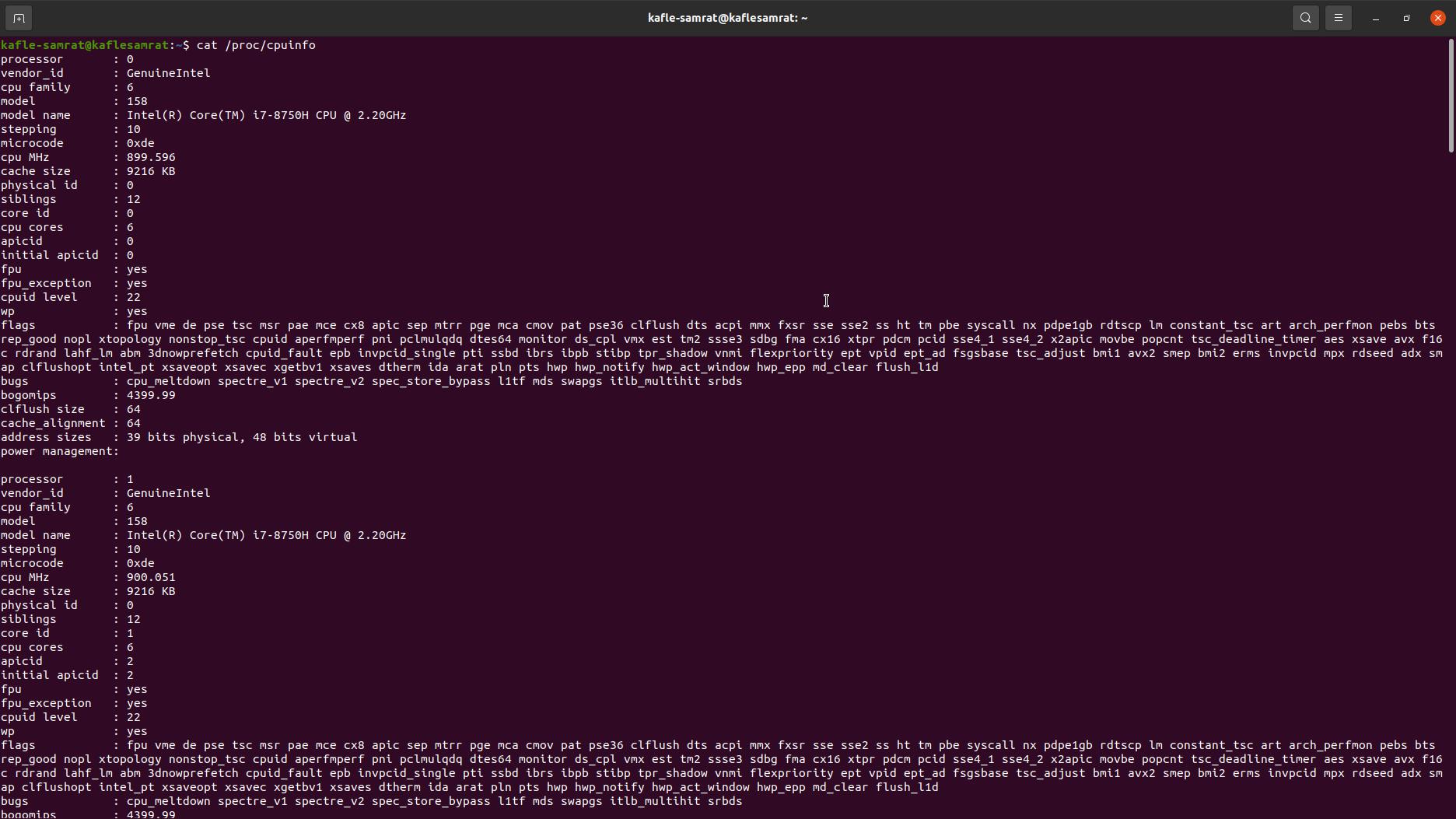
rm -rf \*.o \*~ core .depend .\*.cmd \*.ko \*.mod.c .tmp\_versions

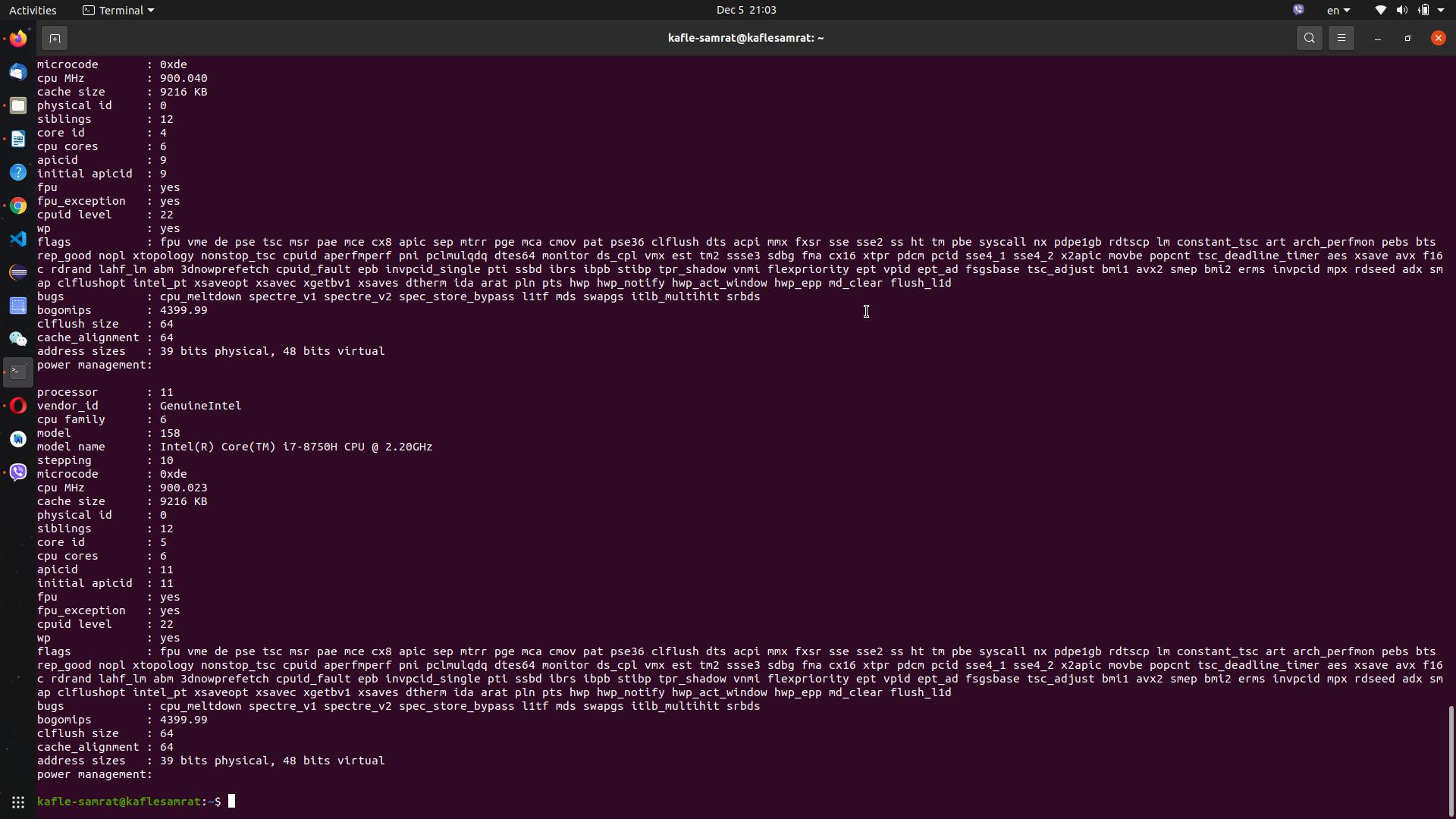
ls /proc

View the contents of the /proc directory

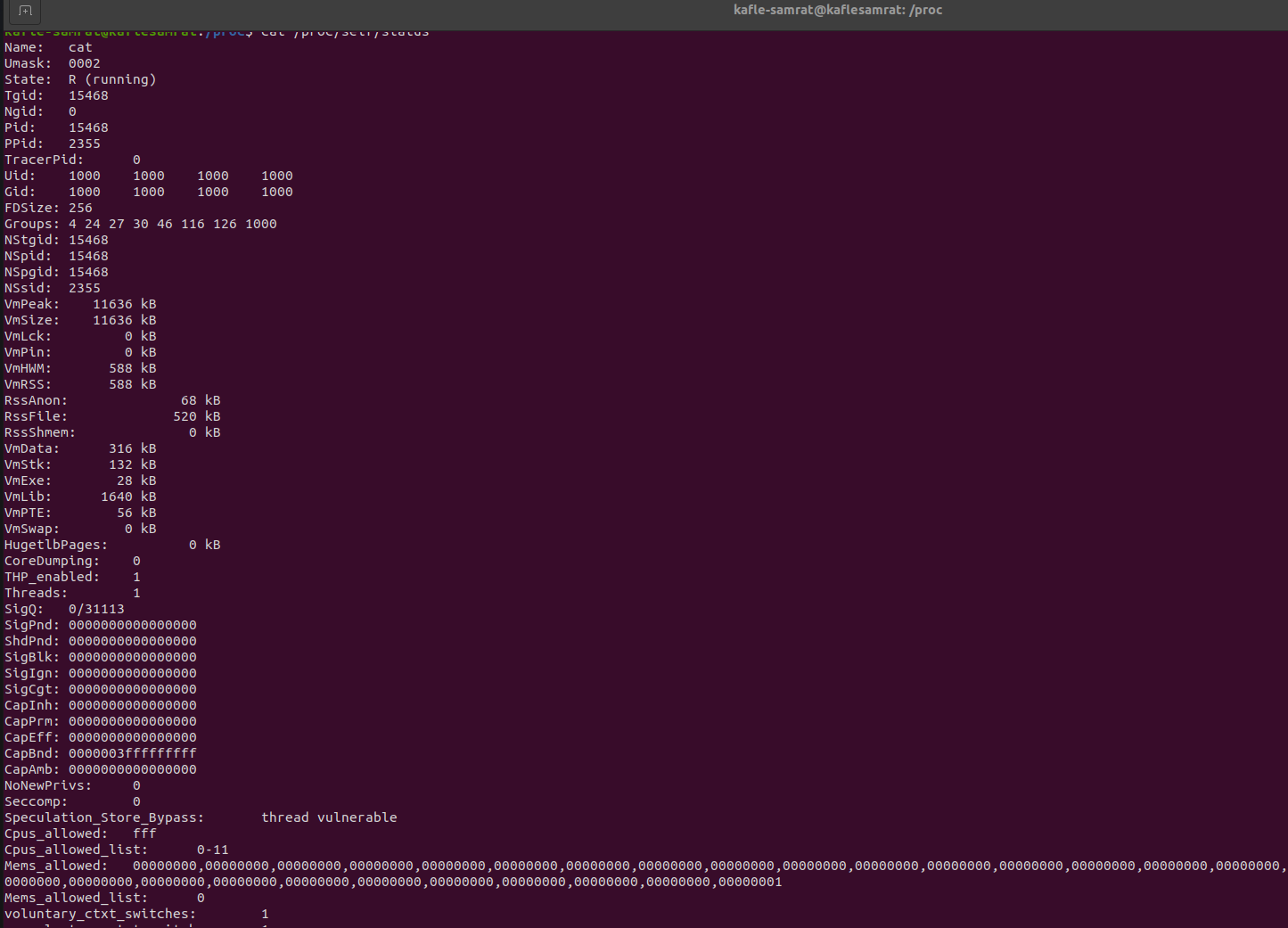
.

cat /proc/cpuinfo check cpu info

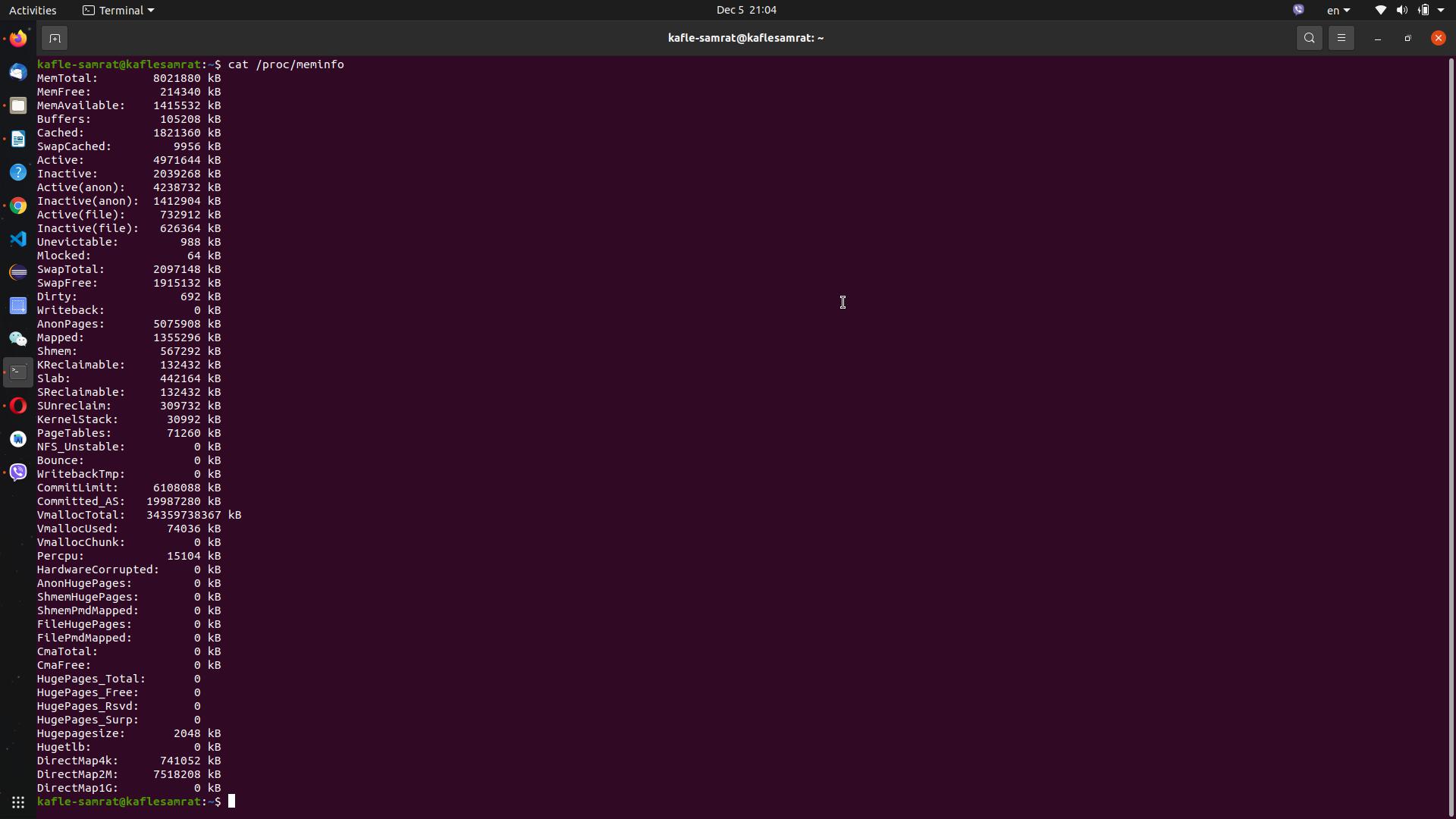


.

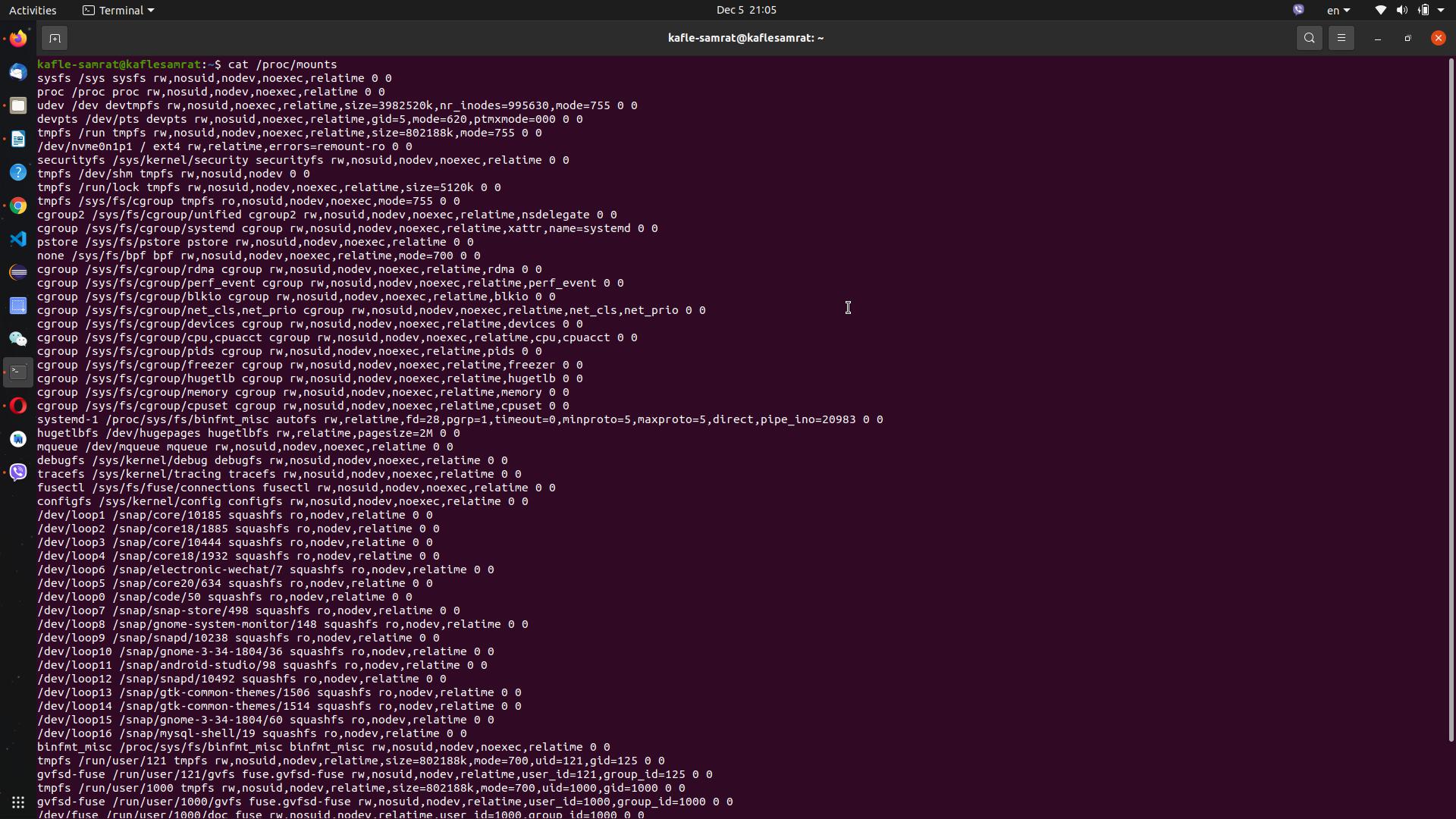
Cat /proc/self/status view the current status of the system

.

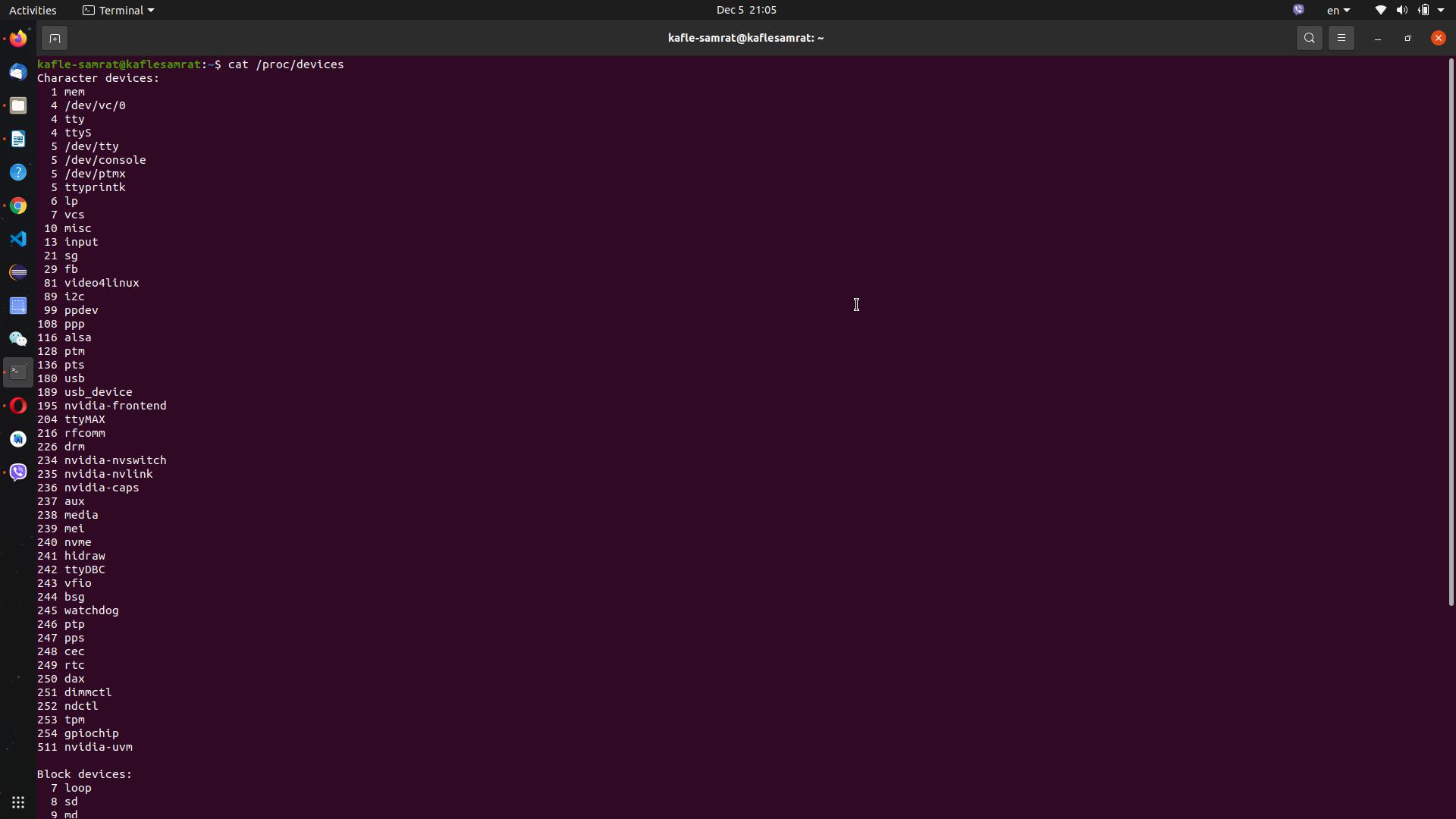
/proc/meminfo and converts the number of bytes given to kilobytes



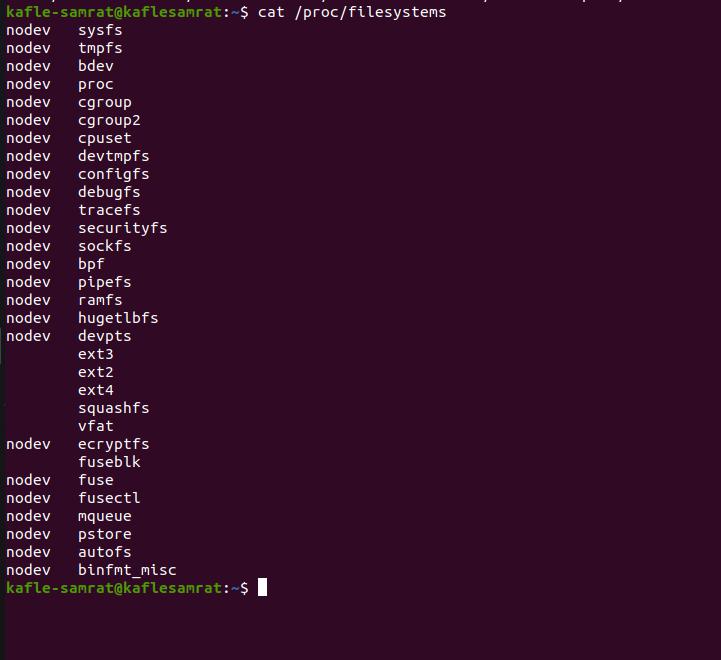
cat *proc/*mounts Table of the file system that has been loaded



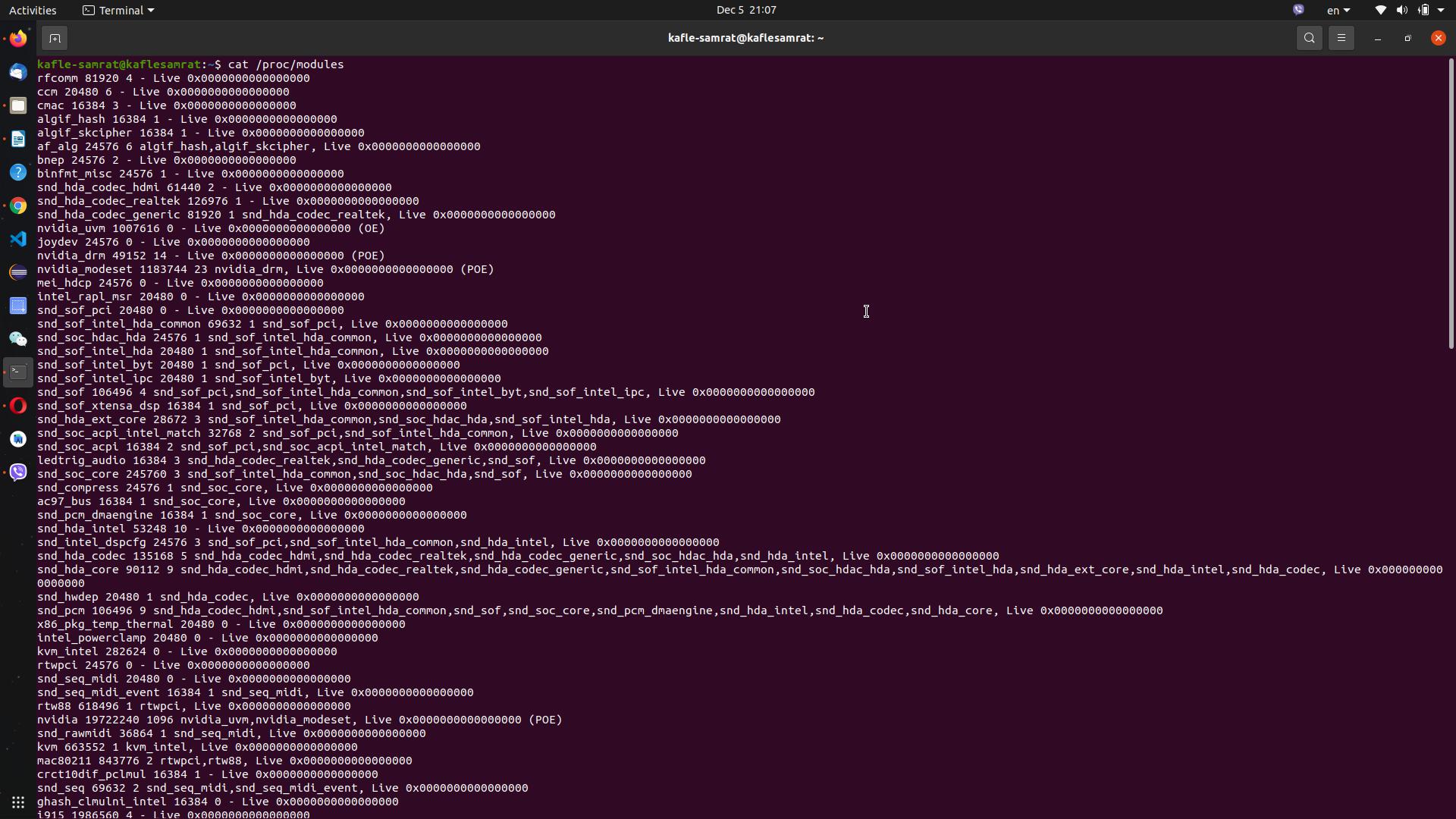
cat *proc*devices uses devices info

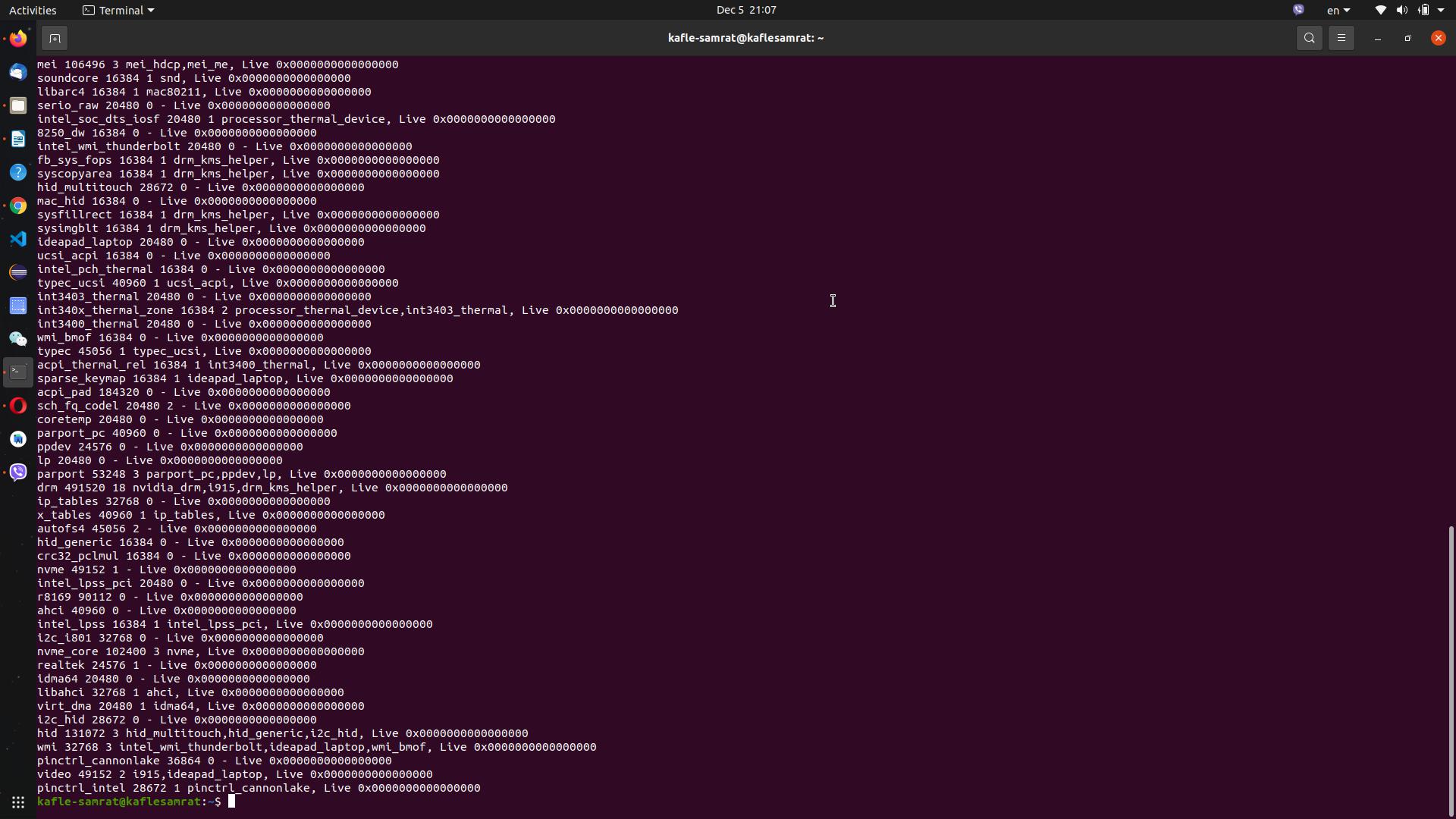


cat *proc/*filesystems check file system

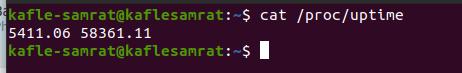


cat *proc*modules check modules

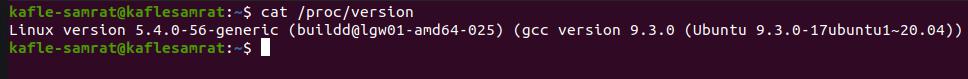




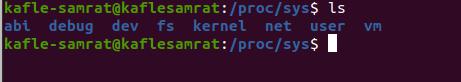
cat proc/uptime shows time



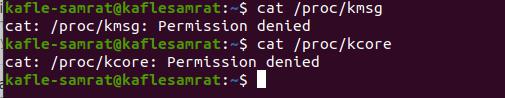
cat *proc/*version check version



la *proc*/sys list system files

.

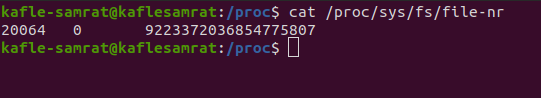
cat proc/kmsg



cat *proc/*swaps



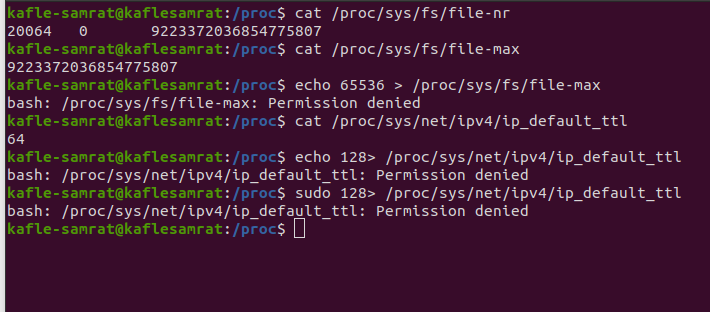
Cat /proc/sys/fs/file-nr looks at the current usage of the file handle

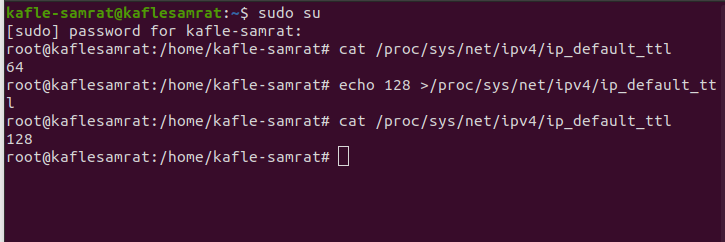
.

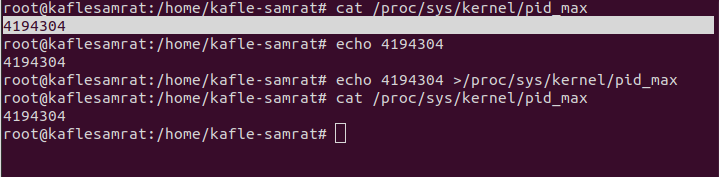
The total number of file handles allocated, the number of file handles currently used, and the maximum number of file handles that can be allocated.

.

.

.

.

.

实验心得机会：

通过这次实验我了解到PROC文件系统是一个虚拟的文件系统，通过文件系统的接口实现，用于输出系统的运行状态。形式的文件系统,它提供了一个接口,用于操作系统本身和应用程序之间的通信过程,这样应用程序就可以安全地和容易获得的当前健康系统和内核的内部数据信息,并且可以修改某些系统的配置信息。此外，由于PROC是作为文件系统接口实现的，所以用户可以像访问常规文件一样访问它，但是它只存在于内存中，而不存在于实际的物理磁盘上。因此，当系统重新启动并关闭电源时，系统中的所有数据和信息都会消失。