**《Linux实验报告》**

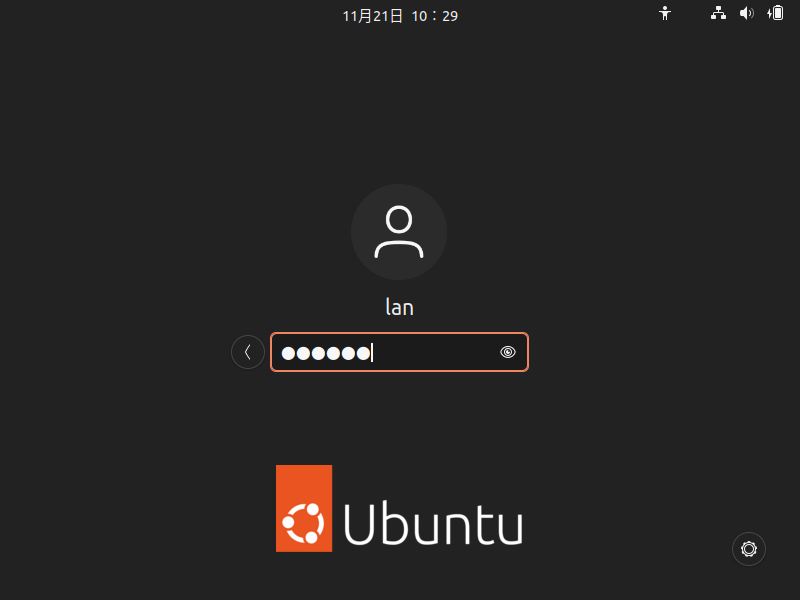
## **实验一**

## **实验内容：**

### 熟悉Linux的登录，Linux的窗口，掌握man、ls、passwd、cp、who、uname、whereis等命令的用法

方式一（虚拟机正常启动）

运行截图：

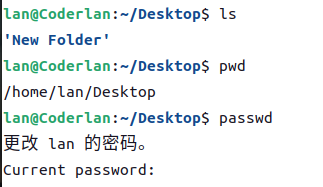


方式二（Xshell等连接工具远程连接）

运行截图：

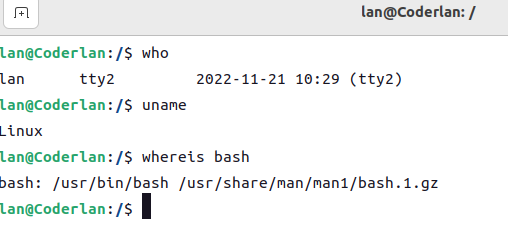
熟练掌握常用命令：

运行截图：



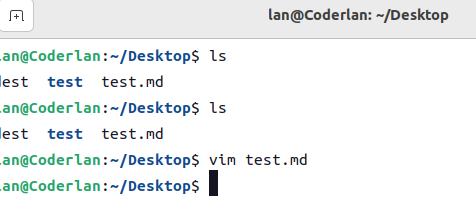
### 学习常用的26个命令

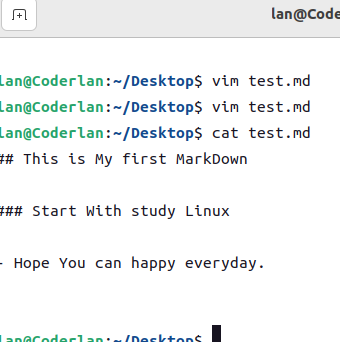
运行截图：



### 运行命令vimtutor，完成Vimtutor教程一遍

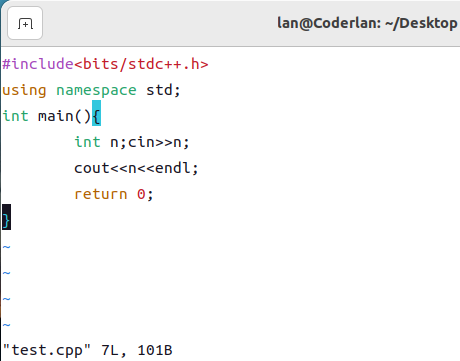
运行截图：





### 用 vi 完成一个简单的 c 语言程序，尽可能多的用 vi 中的各种命令。 用 gcc 编译($ gcc test.c -o test )，并执行该程序。

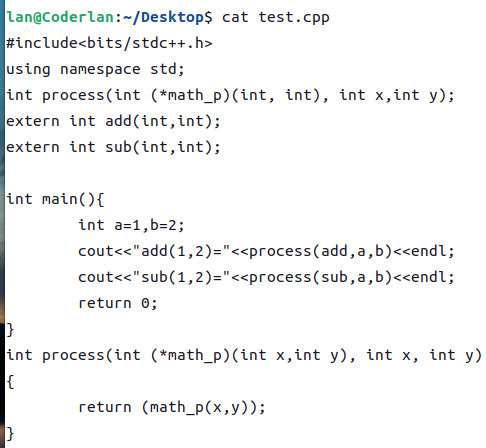
运行截图：

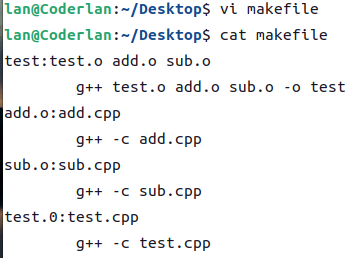


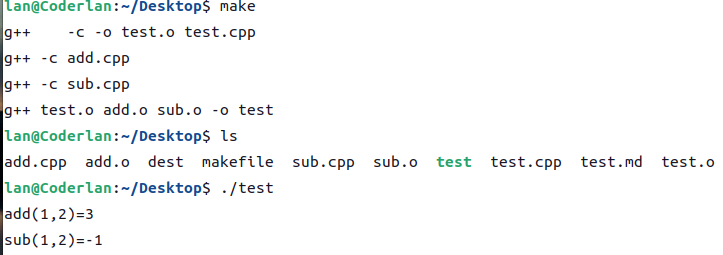


### 学会编写基本的 makefile 文件，使用 makefile 和 make 工具做一个复杂一点的 c 语言程序

运行截图：







## **实验总结：**

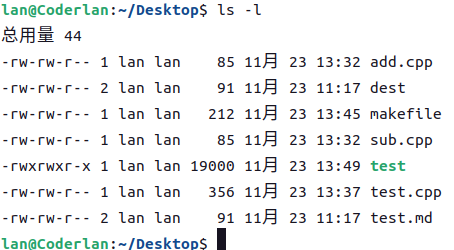
通过这次的实验，我了解了Linux系统的一些基本命令和操作。以及如何在Linux上编写简单的程序，同时学习了如何编写makefile文件和make的使用。

## **实验二**

## **实验A内容：**

### 2.1 运行 ls -l 命令 ,掌握输出各列的意义。

运行截图：



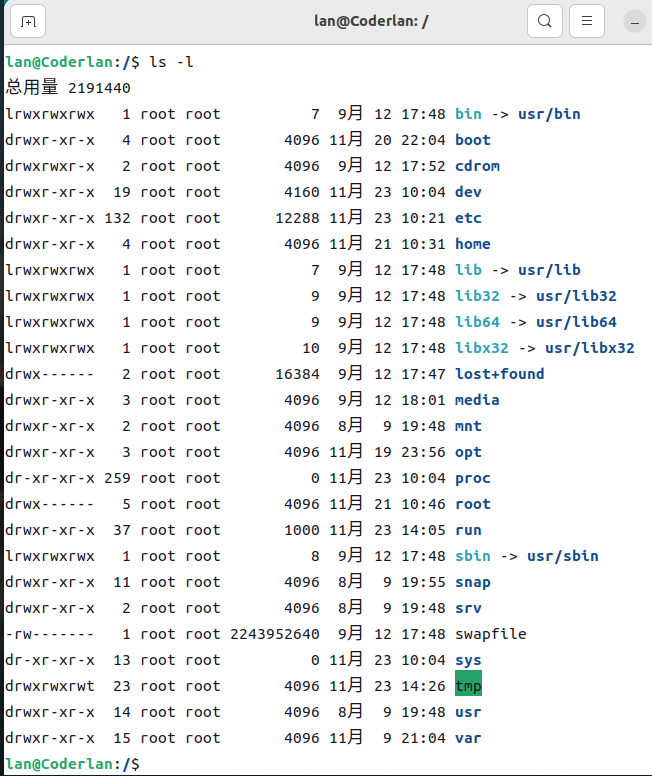
### 2.2 运行 umask，了解掩码的设置。

运行截图：



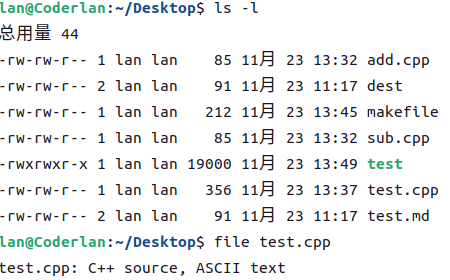
### 2.3 参考课本“3.4 按图索骥”，探索 Linux 系统。对下面表格里的目录逐一探索

运行截图：



### 2.4 参考课本“4.7 实战演练”练习 Linux 系统中对文件的操作

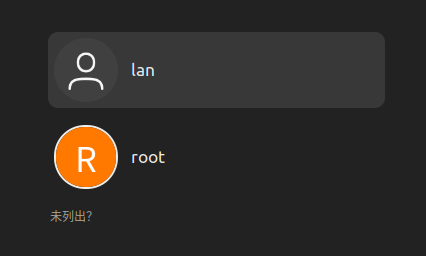
运行截图：



## **实验B内容：**

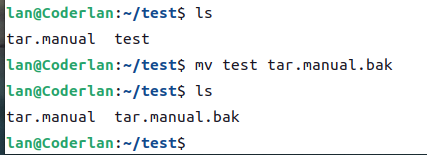
### 建立另外一个普通帐号，并且设置密码。ctrl+d 退出当前登录。

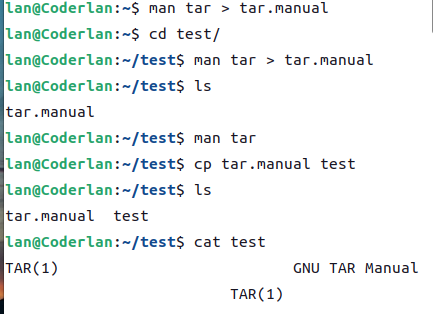
运行截图：



### 创建test文件之后，运行命令 man tar > tar.manual，运行命令复制 tar.manual 为当前目录的 tar.manual.bak 。分别用 cat tac more less head tail 查看 tar.manual ，比较这几个命令的区别。

运行截图：





### 2.3、cat是一次显示整个文件

more命令类似于cat，不过会一页一页的形式显示。

less与more类似，但使用less可以随意浏览文件，而more仅能向前移动，却不能向后移动，而且less在查看之前不会加载整个文件。

head 命令可用于查看文件的开头部分的内容。

tail 命令可用于查看文件尾部的内容。

### 2.4、通过实验过程中的尝试，当使用“>”来重定向标准输出时，会从文件的首位置开始覆盖文件，而“>>”是从文件的尾部开始添加输出内容。



### 运行命令获得当前权限掩码，然后根据权限掩码算一下 tar.manual 的权限值（8 进制）应该是多少，具体权限是什么？ls -l 验证一下。



“002”换算为000 000 010 ，tar.manual 文件默认最大权限为“666”，即 110 110 110，

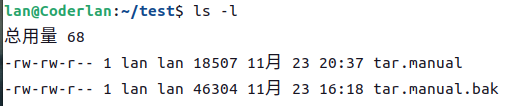
（1）对umask取反 111 111 101

（2）将默认权限和umask取反后的值做与运算

算出 110 110 100

（3）将得到的权限值 110 110 100 转化为8进制即为“664

如下图所示，tar.manual 的权限值为rw-rw-r--，转化之后为权限值为“664”，计算正确。



而该权限值意味着：

创建者权限：可读可写不可执行

创建者所属组权限：可读可写不可执行

访客权限：可读不可写不可执行

## **实验总结：**

通过这次实验，我学习了系统文件的权限设置分配还有，同时也了解了ls-l的各行意义。

## **实验三**

## **实验内容：**

1、运行 du 命令了解/usr/share/man/ 目录所占空间。把/usr/share/man/ 用 tar 打包，分别用

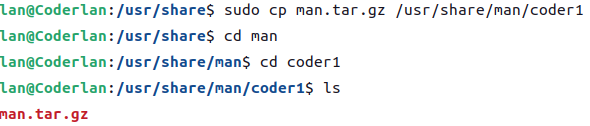
gzip、bzip2、xz 压缩，生成三个压缩包 。比较三者的压缩比。把三个压缩包分别解压到当

前目录下面的 man1 、man2、man3 下面（提示，要先建目录）。

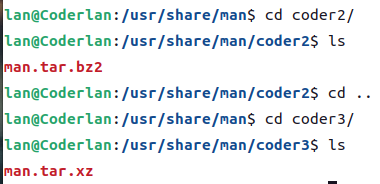
先对/usr/share/man进行打包







同样的方式进行bzip2和xz进行打包压缩

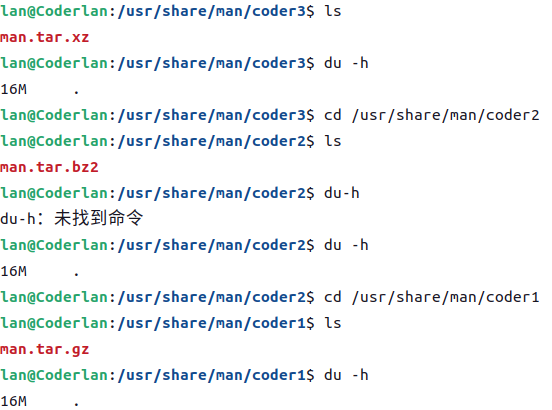


在Coder1、Coder2、Coder3的目录下分别存放了由gzip、bzip2、xz压缩的包

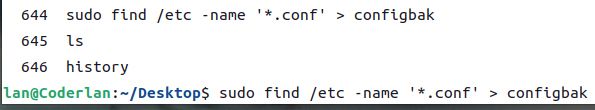
gzip方式的压缩比为 16/136≈0.12

bzip2方式的压缩比为16/136≈0.12

xz方式的压缩比为16/136≈0.12



1. 用 find 命令找出 /etc 下所有 以 .conf 结尾的文件 ，并且把结果 保存到文件 configbak中。

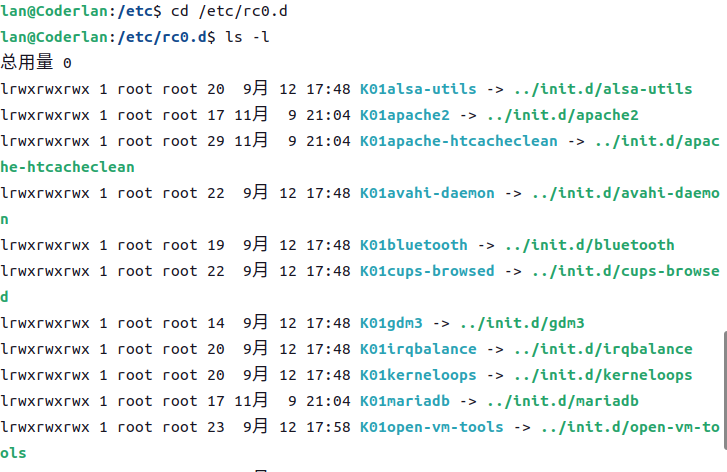


1. 搜索 configbak 文件中和 pass 有关的行。



4、到/etc 打开 rcN.d(N=0,1,2,...)中的任意一个目录，用 ls 查看目录中的文件的类型，请问，

目录中的文件是哪种文件类型？



使用ls -l查看目录中的文件，因为前缀是 l ，所以可以确定这些文件都是符号链接（软链接）

5、运行 whereis 命令找到 passwd 的路径，运行 ls -i 命令了解它的 inode 值，用 ls 命令获 得 passwd 的权限。（思考题：passwd 的权限值是多少？）

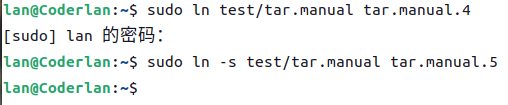


ls -i 得到/usr/bin/passwd 的inode值为2229065

用 ls 命令获得passwd的权限：拥有者具有读、写、执行权限，属组用户具有读、执行权限、其他用户具有读、执行权限。而passwd 的权限值为“755”。

6、为 test 目录下的 tar.manual 文件建立一个硬链接 tar.manual.4 ，一个软链接 tar.manual.5,

运行 ls -l。总结一下软链接和硬链接的异同。



如图所示，分别为tar.manual 建立了一个硬链接 tar.manual.4 ，一个软链接 tar.manual.5

关于硬链接和软链接的区别，硬链接可认为是一个文件拥有两个文件名;

而软链接则是系统新建一个链接文件，此文件指向其所要指的文件

若源文件改名，则软连接不可用（找不到源文件）；硬链接可正常访问到源文件；

7、找到正则表达式至少三份不同的资料，掌握正则表达式基本规则，了解正则表达式在 vi，

grep，.net，java，python,C，shell 程序中的应用。

正则表达式定义了字符串的模式。

正则表达式可以用来搜索、编辑或处理文本。

正则表达式并不仅限于某一种语言，但是在每种语言中有细微的差别。

Vi: :%s/\<id\>/\=line(“.”) => 将各行的id字符串替换为行号

Python: 在Python中，从1.5开始增加的re模块，让Python拥有全部的正则表达式功能。

Eg: 寻找字符串位置：  
 import re print(re.search('www', 'www.runoob.com').span()) # 在起始位置匹配

Java: 使用pattern类对正则表达式进行编译

Eg：字符串匹配：

boolean isMatch = Pattern.matches(".\*runoob.\*", "I am noob " + "from runoob.com.");

## **实验总结：**

通过这次的实验，我对Linux的内存分配和打包操作更加的熟练，同时也复习了各个程序中正则的使用方式。

## 实验四

## **实验内容：**

1、用 ps 了解 Linux 当前进程，说明选项：a、u、x、f 的意义。

建立普通用户，用普通用户登录系统，说明普通用户运行的进程有几个，状态代号是什么，

代表什么状态？

提示：得到普通用户运行的进程用 ps aux|grep 当前用户的 UID

或者你还有更好的方法。状态就是 STAT 列。

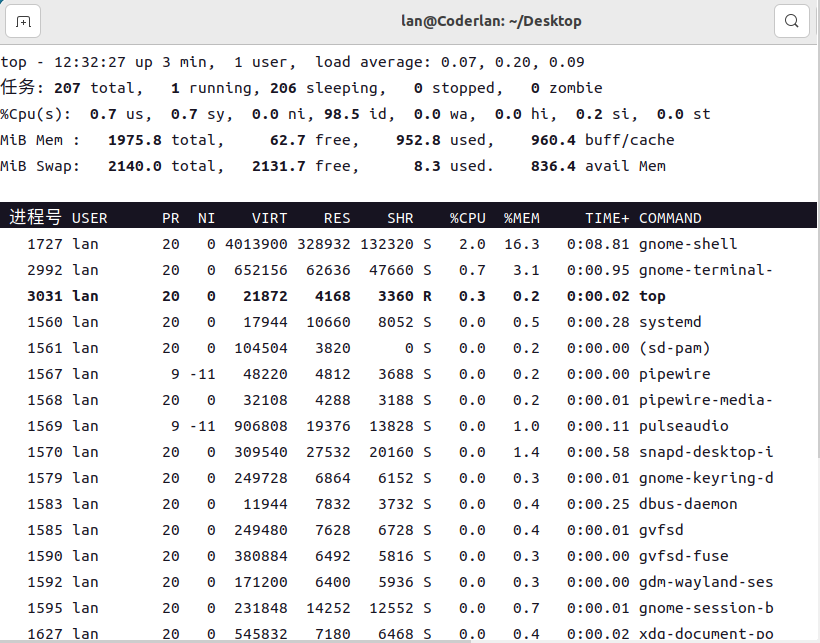
"ps -a" 显示一个终端的所有进程，除了会话引线

"ps -f" 全部列出，通常和其他选项联用

"ps -u" 选择有效的用户id或者是用户名

"ps -x" 显示没有控制终端的进程，同时显示各个命令的具体路径

使用top -u lan命令查看lan用户的进程,top可以动态的显示用户的进程信息

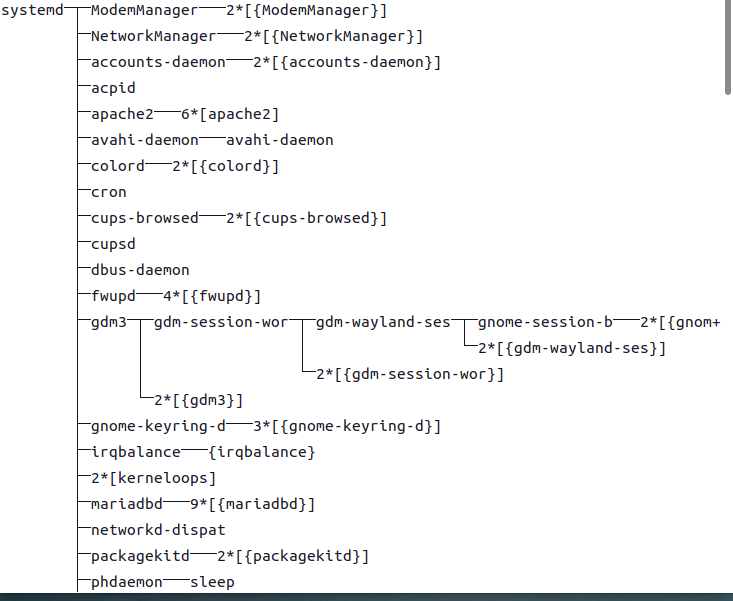


总共有207个，状态代号有S和R， R ：该程序目前正在运作，或者是可被运作

S ：该程序目前正在睡眠当中 (可说是 idle 状态)，但可被某些讯号 (signal) 唤醒。T=跟踪/停止,Z=僵尸进程)

2、用 ps auxf 和 pstree 了解进程之间的关系，把两条命令输出的有父子关系的进程关系图

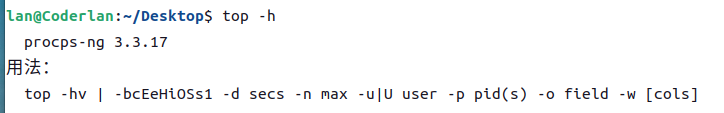
画出来。



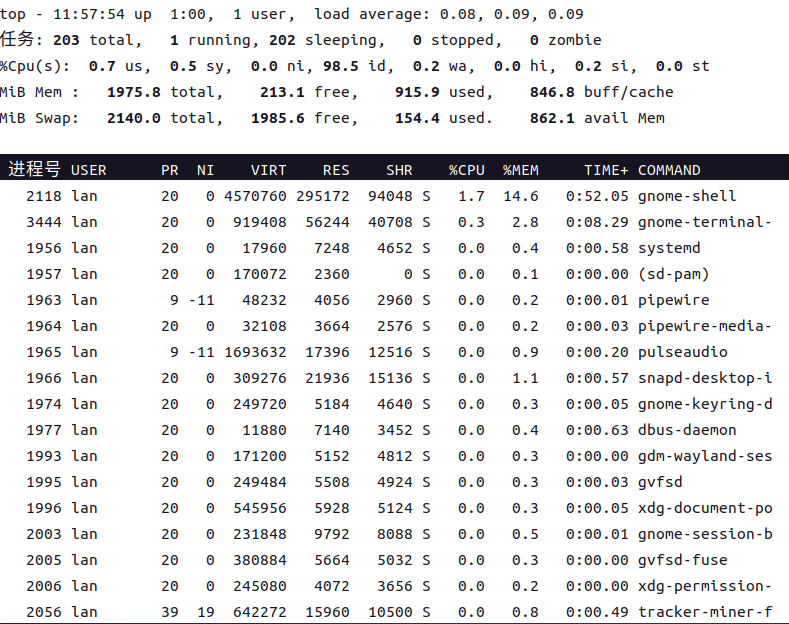
3、运行 top 命令，用 h 了解当前版本的 linux 的各项快捷键。

重点掌握使用 u，F（注意大小写）

top可以动态获取进程信息，每三秒刷新一次。



其中u/U 进程列表只看指定用户的进程，大致的界面信息如下：



F的

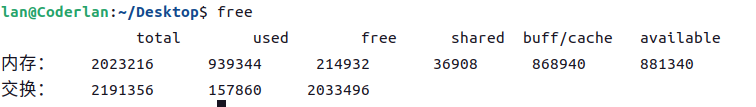
4、linux 获得资源情况的命令很多，

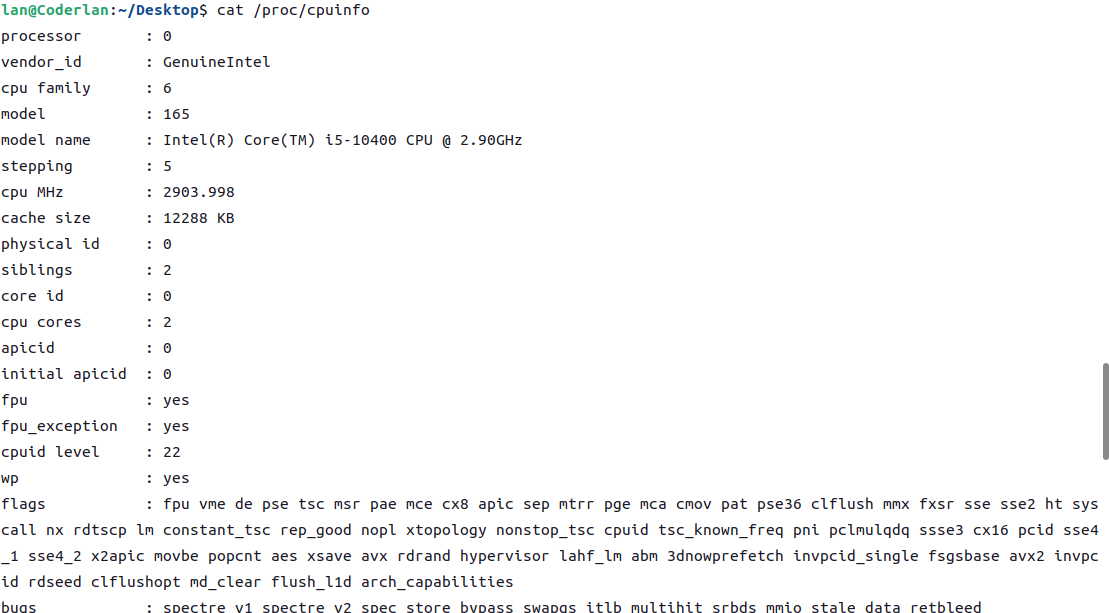
free 了解内存资源情况

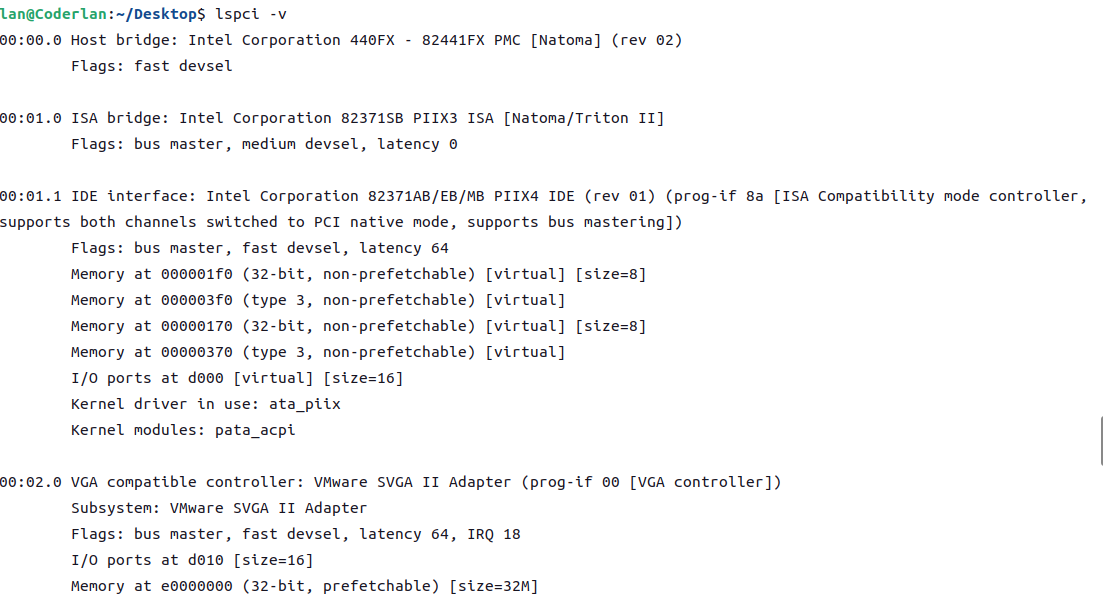
了解当前 cpu 信息：cat /proc/cpuinfo

了解硬件情况：lspci -v

实验上述命令，把得到的信息记录下来，重点了解得到的信息含义。







5、实验顺序执行和并行执行。

例如： date ; cal

date & cal

请大家自己设计两个命令进行顺序执行和并行执行的实验。

顺序执行：

并行执行：

Ls & cd /



6、实验有条件执行。

当前目录下先运行 mkdir test && man ps > ./test/ps.man

再运行 mkdir test && man top > ./test/top.man

上述命令是否都运行完成，为什么？

再试试 mkdir test || mkdir test1

列目录看看 test1 是否建立，重点了解 || 的含义。

自己设计两个实验比较 &&和|| 的不同。

7、实验 kill 进程

运行某个程序，用 ps 得到它的 PID ，并用 kill 关闭它。

自己查资料了解 kill 一个进程可能带来的问题。

8、列出你当前运行的 linux 版本启动后运行的守护进程。

提示：看看目录/etc/rc2.d (ubuntu),/etc/rc3.d(redhat 或者 SUSE 或者 fc6）中的文件。

或者用 ps 查看进程。

用 man 命令了解这些守护进程的作用，记录下来。

## **实验总结：**