

# Lab01 Assignment

班级：192115 学号：19373257 姓名：黄泽桓

## 实验准备

- 请安装一个合适的 Linux 系统，你安装的 Linux 发行版及版本号是什么？内核版本号是什么？

截图：

```
Linux version 5.8.0-44-generic (buildd@lgw01-amd64-054) (gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0, GNU ld (GNU Binutils for Ubuntu) 2.34) #50~20.04.1-Ubuntu SMP
Wed Feb 10 21:07:30 UTC 2021
```

答案：我安装的 Linux 发行版为 Ubuntu，版本号为 20.04，内核版本号为 Linux 5.8.0.

- 查看你的根目录下有哪几个子目录，每个子目录主要用来做什么用的？

截图：

```
root@huan-Legion-Y7000-2019:/home/huan# cd /
root@huan-Legion-Y7000-2019:/# ls
bin    cdrom  etc    lib    lib64  lost+found  mnt    root  sbin  srv  tmp  var
boot  dev   home  lib32  libx32  media      proc   run   snap  sys  usr
```

答案：我的根目录下有 23 个子目录，下面列出主要架构的作用

子目录	作用
/bin	二进制可执行命令，多是重要系统文件
/dev	存放设备文件
/root	超级用户的主目录
/home	普通用户主目录的基点，比如 /home/user
/usr	存放软件
/opt	存放可选的安装的软件
/tmp	存储临时文件
/var	存放比如各种服务的日志文件
/sbin	存放标准系统管理程序
/lib	标准程序设计库，类似 windows 的 .dll 文件

- 查看自己的 ip 地址，并 ping 一下 baidu.com 看网络是否连通？

命令：ip addr and ping baidu.com

截图：ip address: 10.136.78.14

```
huan@huan-Legion-Y7000-2019:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp7s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 98:fa:9b:0a:42:1f brd ff:ff:ff:ff:ff:ff
3: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 3c:f0:11:41:96:05 brd ff:ff:ff:ff:ff:ff
    inet 10.136.78.14/16 brd 10.136.255.255 scope global dynamic noprefixroute wlp0s20f3
        valid_lft 2920sec preferred_lft 2920sec
    inet6 2001:da8:bc:8b20:f40b:c095:e47c:a5e1/64 scope global temporary dynamic
        valid_lft 258511sec preferred_lft 83612sec
    inet6 2001:da8:bc:8b20:e933:237c:13eb:526a/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 258511sec preferred_lft 172111sec
    inet6 fe80::afa8:622f:9d19:1813/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
huan@huan-Legion-Y7000-2019:~$
```

ping baidu.com:

```
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=9 ttl=46 时间=10.9 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=10 ttl=46 时间=11.3 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=11 ttl=46 时间=11.7 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=12 ttl=46 时间=11.4 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=13 ttl=46 时间=11.8 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=14 ttl=46 时间=8.80 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=15 ttl=46 时间=10.9 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=16 ttl=46 时间=11.3 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=17 ttl=46 时间=11.5 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=18 ttl=46 时间=11.8 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=19 ttl=46 时间=10.9 毫秒
64 比特, 来自 39.156.69.79 (39.156.69.79): icmp_seq=20 ttl=46 时间=11.7 毫秒
```

- 用软件安装命令下载 build-essential。

命令：sudo apt install build-essential

截图：

```
huan@huan-Legion-Y7000-2019:~$ sudo apt install build-essential
[sudo] huan 的密码：
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
build-essential 已经是最新版 (12.8ubuntu1.1)。
build-essential 已设置为手动安装。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 10 个软件包未被升级。
```

## 1. Linux 命令操作

- Linux 命令行操作，请用你学到的 Linux 命令，实现以下操作，给出每一步你的命令行截图。
  - 用 wget 从 <https://github.com/BUAA-SE-2021/patpat/releases/download/v0.1.3/patpat-linux-amd64> 下载你们的 Linux 版 OOP 课 Java 自助评测机 patpat
  - <https://github.com/BUAA-SE-2021/sp-labs/lab01/lab01.zip> 处下载实验压缩包

- 解压lab01.zip。
- 进入lab01目录,进入子目录etc,打印当前路径并在当前路径下创建名为a1的目录,并在a1目录中创建名为b1的目录。
- 进入b1目录中,创建两个文件a.txt, b.txt。
- 返回子目录etc,删除目录a1。
- 将etc目录下所有以tmp开头的文件移动到lab01目录下的Download目录下的tmp目录中。
- 查看tmp目录下a1005.cpp的内容。
- 查看tmp目录下a1009.cpp的前十行和后十行。
- 将tmp目录下的所有文件打成一个tar包,并命名为并命名为tmp.tar.gz。
- 返回lab01目录,列出当前目录下的文件大小。
- 用命令找出空目录并将空目录删除。

命令:

```
# 不完整的部分自行补充,填写必要注释
wget -O patpat https://github.com/BUAA-SE-2021/patpat/releases/download/v0.1.3/patpat-linux-amd64 # 下载
wget -O lab01.zip https://github.com/BUAA-SE-2021/sp-labs/lab01/lab01.zip # 下载实验资料
unzip -n lab01.zip # 解压
cd lab01/etc # 进入目录
pwd # 打印当前路径
mkdir a1 # 创建目录`a1`
mkdir a1/b1 # 创建子目录`b1`
cd a1/b1 # 进入`b1`目录中
touch a.txt # 创建文件`a.txt`
touch b.txt # 创建文件`b.txt`
cd ../../ # 返回子目录`etc`
rm -rf a1 # 删除`a1`
cd c1
mv tmp* ../../Download/tmp # 移动
cd ../../Download/tmp
cat a1005.cpp # 查看
head -n 10 a1009.cpp # 查看前十行
tail -n 10 a1009.cpp # 后十行
tar cvf tmp.tar.gz * # 打包
cd ../../ # 返回`lab01`目录
du -sh # 列出当前目录下的文件大小
find -type d -empty # 用命令找出空目录
rmdir ./upload # 删除空目录
```

必要的实验截图(如查看前十行、查看文件大小的展示效果)

```
huan@huan-Legion-V7000-2019:~/Lab$ wget -O patpat https://github.com/BUAA-SE-2021/patpat/releases/download/v0.1.3/patpat-linux-amd64
huan@huan-Legion-V7000-2019:~/Lab$ wget -O lab01.zip https://github.com/BUAA-SE-2021/sp-labs/lab01/lab01.zip
```

```
huan@huan-Legion-Y7000-2019:~/Lab$ unzip -n lab01.zip
Archive:  lab01.zip
  creating:  lab01/
  creating:  lab01/Download/
  creating:  lab01/Download/tmp/
  inflating: lab01/Download/tmp/a1005.cpp
  inflating: lab01/Download/tmp/a1009.cpp
  creating:  lab01/etc/
  creating:  lab01/etc/c1/
  extracting: lab01/etc/c1/sc.txt
  extracting: lab01/etc/c1/tmp1.c
  extracting: lab01/etc/c1/tmp2.cpp
  extracting: lab01/etc/c1/tmptmp.txt
  creating:  lab01/upload/
```

```
huan@huan-Legion-Y7000-2019:~/Lab/lab01$ cd etc
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc$ pwd
/home/huan/Lab/lab01/etc
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc$ mkdir a1
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc$ cd a1
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/a1$ mkdir b1
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/a1$
```

```
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc$ cd a1
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/a1$ ls
b1
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/a1$ cd ../
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc$ cd a1/b1
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/a1/b1$ touch a.txt
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/a1/b1$ touch b.txt
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/a1/b1$ ls -ll a.txt
-rw-rw-r-- 1 huan huan 0 3月  6 00:32 a.txt
```

```
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/a1/b1$ cd ../../
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc$ rm -rf a1
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc$ ls
c1
```

```
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc$ cd c1
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/c1$ ls
sc.txt  tmp1.c  tmp2.cpp  tmptmp.txt
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/c1$ mv tmp* ../../Download/tmp
huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/c1$ ls
sc.txt
```

```

huan@huan-Legion-Y7000-2019:~/Lab/lab01/etc/c1$ cd ../../Download/tmp
huan@huan-Legion-Y7000-2019:~/Lab/lab01/Download/tmp$ ls
a1005.cpp  a1009.cpp  tmp1.c  tmp2.cpp  tmptmp.txt
huan@huan-Legion-Y7000-2019:~/Lab/lab01/Download/tmp$ cat a1005.cpp
#include <iostream>
#include <cstdio>
#include <algorithm>

int hash[10500];
void add(int num){
    if(num%2==0){
        num/=2;
    }
    else num=(num*3+1)/2;
    while (num!=1)
    {
        if(hash[num]==0){
            hash[num]=1;
        }
        if(num%2==0){
            num/=2;
        }
        else {
            num=num*3+1;

```

```

huan@huan-Legion-Y7000-2019:~/Lab/lab01/Download/tmp$ head -n 10 a1009.cpp
#include <iostream>
#include <cstdio>

using namespace std;

char str[100][80];
int main()
{
    int i=0;
    while (~scanf("%s",str+i))

```

```

huan@huan-Legion-Y7000-2019:~/Lab/lab01/Download/tmp$ tail -n 10 a1009.cpp
{
    i++;
}
for(int j=i-1;j>=0;j--){
    if(j==i-1)
        printf("%s",str+j);
    else printf(" %s",str+j);
}
return 0;
}

```

```

huan@huan-Legion-Y7000-2019:~/Lab/lab01/Download/tmp$ tar cvf tmp.tar.gz *
a1005.cpp
a1009.cpp
tmp1.c
tmp2.cpp
tmptmp.txt
huan@huan-Legion-Y7000-2019:~/Lab/lab01/Download/tmp$ ls
a1005.cpp  a1009.cpp  tmp1.c  tmp2.cpp  tmp.tar.gz  tmptmp.txt

```

```

huan@huan-Legion-Y7000-2019:~/Lab/lab01$ du -sh
44K      .

```

## 2. vi 模式

- vi 编辑器有哪几种模式？简述这几种模式间如何互相切换？

模式：

- 1 命令模式
- 2 插入模式
- 3 末行模式

如何切换：**vi** <fileName> 后进入命令模式，**:** 进入末行模式，输入 **i** 进入插入模式，插入模式 **ESC** 进入命令模式，末行模式回车进入命令模式，命令模式 **:wq** 保存并退出

### 3. vi 命令

写出以下一系列操作使用的**命令**（底行模式的命令加上 **:** 或 **/**）：

#### 3.1. 用 **vi** 在当前用户家目录下新建文件 **testfile** 并打开

```
vi ~/testfile
```

#### 3.2. 设置显示行号

```
:set number
```

#### 3.3. 进入 **insert mode**，输入 **3** 行文本

```
This is the first line.  
This is the second line.  
This is the third line.
```

```
1 This is the first line.  
2 This is the second line.  
3 This is the third line.
```

#### 3.4. 返回到 **command mode**，将光标移动到第 **2** 行，复制这一行

```
yy
```

#### 3.5. 移动光标到文档最后一行，将复制内容粘贴到这一行后面

```
p
```

```
1 This is the first line.  
2 This is the second line.  
3 This is the third line.  
4 This is the second line.
```

### 3.6. 移动光标到第一行，删除第一行

```
dd
```

```
1 This is the second line.  
2 This is the third line.  
3 This is the second line.
```

### 3.7. 从文档开头开始查找 **second**，然后查找下一个

```
/second  
n
```

### 3.8. 将全部 **second** 替换为 **fourth**，替换过程不需要询问

```
:1,$s/second/fourth/g
```

```
This is the fourth line.  
This is the third line.  
This is the fourth line.
```

### 3.9. 取消显示行号

```
set nonumber
```

```
This is the fourth line.  
This is the third line.  
This is the fourth line.
```

### 3.10. 保存修改并退出 **vi**

```
:wq
```

### 3.11. 再次用 **vi** 打开 **testfile**，另存为 **testfile2**



```
:w testfile2
```

## 4. POSIX

- 什么是 POSIX 标准？哪些操作系统支持 POSIX 标准？

可移植操作系统接口，包括了系统调用接口(API)，和实时扩展[C语言]。支持 POSIX 的系统有 Linux, Solaris, Windows 等等。

## 5. LSB

- 什么是 LSB 标准？分析它和 POSIX 标准的异同。

LSB 是保证了操作系统同应用程序之间的良好结合，保证**应用程序在系统间迁移时不用重新编译**，且保持稳定；而 POSIX 规定了系统调用接口，为的是一个系统编写的程序，能够在**其他支持 POSIX 的系统上编译执行**。

## 6. 实验感想

以前我所掌握的 Linux 使用技巧较为零碎，通过本次课程、实验和自我整理总结，我对 Linux 的使用 (包括 vim) 掌握得更加系统、熟练。

截图是真的累