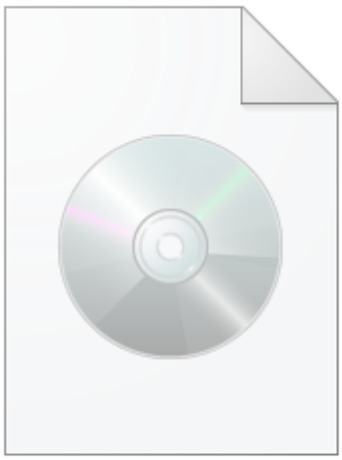
1 (30分) 用虚拟机方式安装CentOS 7 操作系统，并掌握LINUX平台下的基本命令

> 要求：
>
> (1)尽可能的多测试LINUX平台的各种命令。
>
> (2)学习使用管道符组合使用命令
>
> 评分标准：实验的命令必须有运行结果截图，否则扣5分。

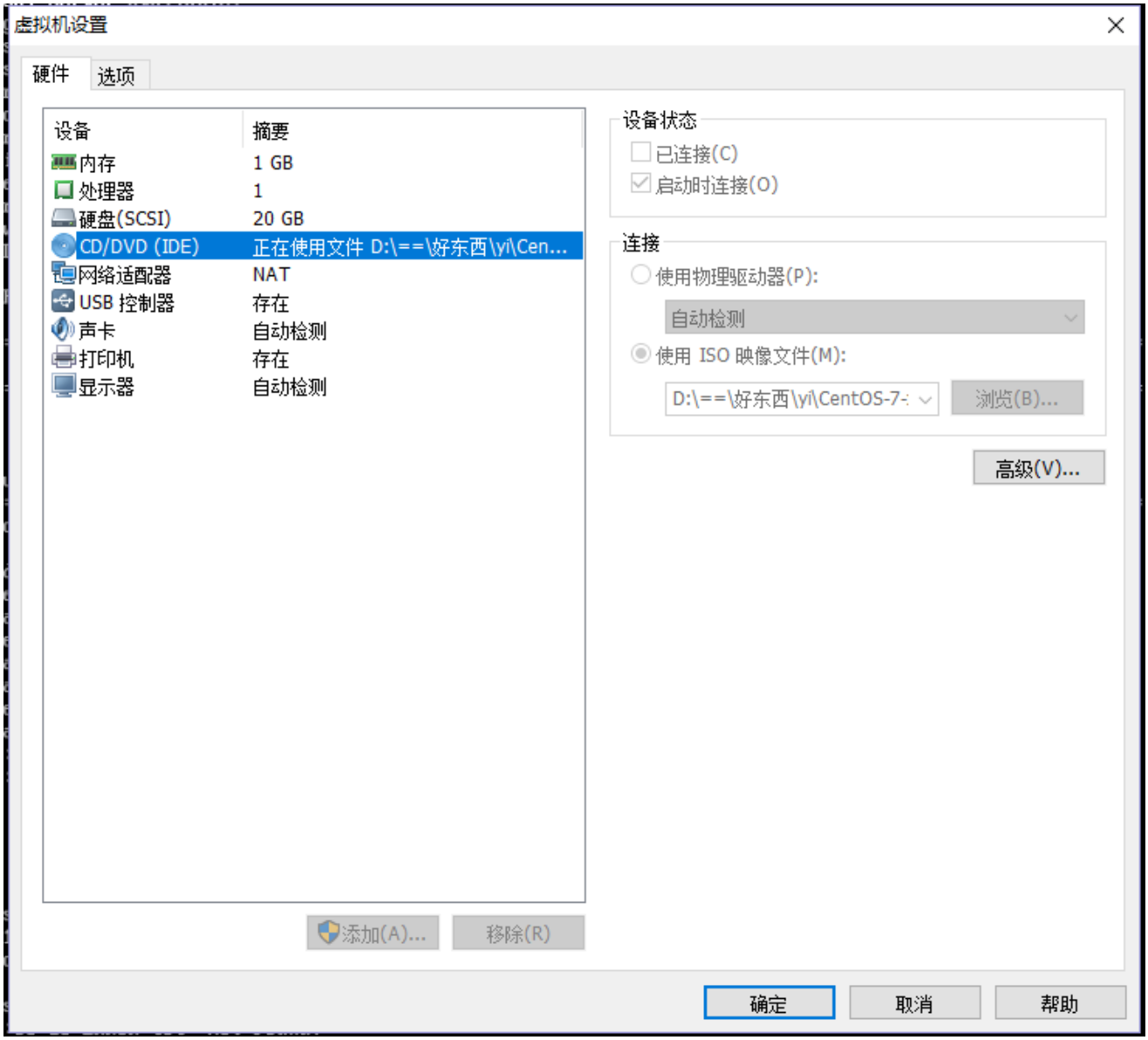
1.1 安装CentOS 7

官网下载CentOS 7镜像（网络及内存原因下载minimal版本）



CentOS-7-x86_64-Minimal-1810.iso

在VMWARE上新建CentOS虚拟机，挂载镜像



启动虚拟机，进行minimal安装，配置账号密码，语言时区等各种信息，最后安装完成没有图形界面只有命令行，所以使用shell连接工具连接上虚拟机进行下一步操作。

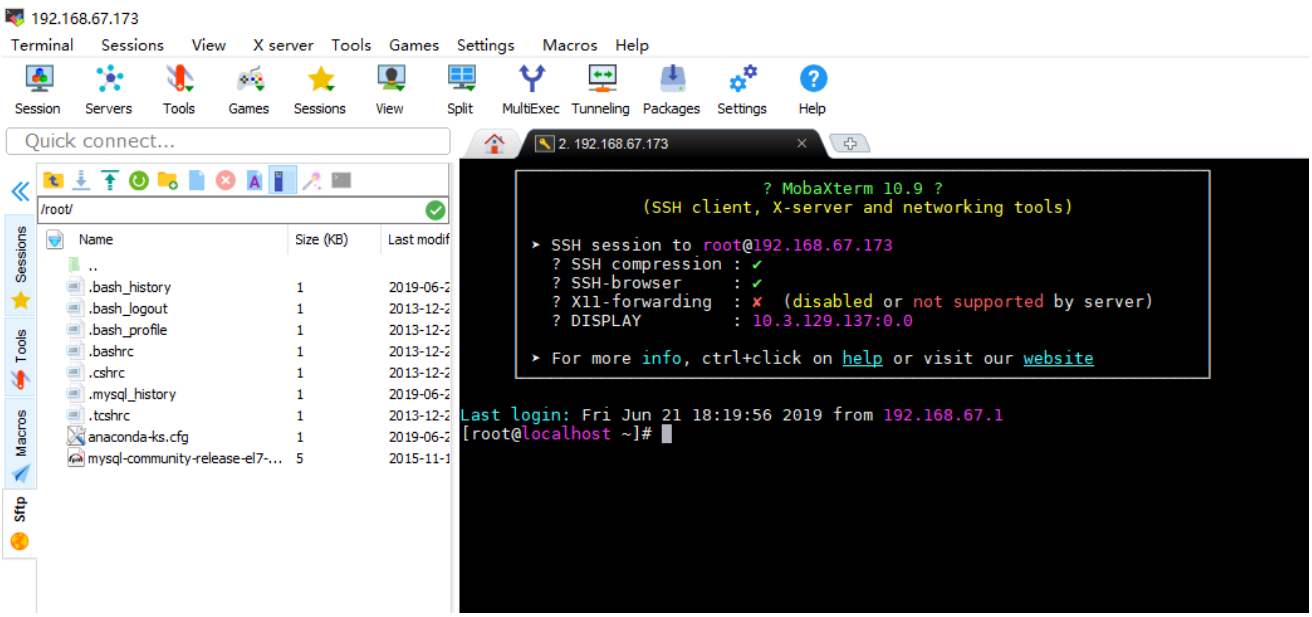
查看ens33网卡的配置 `vi /etc/sysconfig/network-scripts/ifcfg-ens33` ，发现 CentOS 7 默认不启动网卡（ONBOOT=no），改为yes后保存退出。

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=f2082275-6cad-4a7d-b7b1-1ec19caf3439
DEVICE=ens33
ONBOOT=yes
~
~
~
```

重启网络服务 `sudo service network restart` ，输入 `ip addr` 查看ip

```
[root@localhost ~]# sudo service network restart
Restarting network (via systemctl):                        [  OK  ]
[root@localhost ~]# 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:41:93:23 brd ff:ff:ff:ff:ff:ff
    inet 192.168.67.173/24 brd 192.168.67.255 scope global noprefixroute dynamic ens33
       valid_lft 1797sec preferred_lft 1797sec
    inet6 fe80::3cd2:fe:8741:3e95/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@localhost ~]#
```

使用MobaXterm新建ssh连接上虚拟机，用root账户登录



1.2 基本命令演示

`pwd [--help][--version]` Linux pwd命令用于显示目前所在的工作目录的绝对路径名称。

`ls [-alrtAFR] [name...]` Linux ls命令用于显示指定工作目录下之内容（列出目前工作目录所含之文件及子目录)。

- -a 显示所有文件及目录 (ls内定将文件名或目录名称开头为"."的视为隐藏档，不会列出)
- -l 除文件名称外，亦将文件型态、权限、拥有者、文件大小等资讯详细列出
- -r 将文件以相反次序显示(原定依英文字母次序)
- -t 将文件依建立时间之先后次序列出
- -A 同 -a ，但不列出 "." (目前目录) 及 ".." (父目录)
- -F 在列出的文件名称后加一符号；例如可执行档则加 "*", 目录则加 "/"
- -R 若目录下有文件，则以下之文件亦皆依序列出

```
[root@localhost ~]# pwd
/root
[root@localhost ~]# ls -a
.  ..  anaconda-ks.cfg  .bash_history  .bash_logout  .bash_profile  .bashrc  .cshrc  mysql-community-release-el7-5.noarch.rpm  .mysql_history  .tcshrc
[root@localhost ~]# ls
anaconda-ks.cfg  mysql-community-release-el7-5.noarch.rpm
[root@localhost ~]# ls -l
total 12
-rw-------. 1 root root 1444 Jun 21 17:16 anaconda-ks.cfg
-rw-r--r--. 1 root root 6140 Nov 12  2015 mysql-community-release-el7-5.noarch.rpm
```

`mkdir [-p] dirName` Linux mkdir命令用于建立名称为 dirName 之子目录。

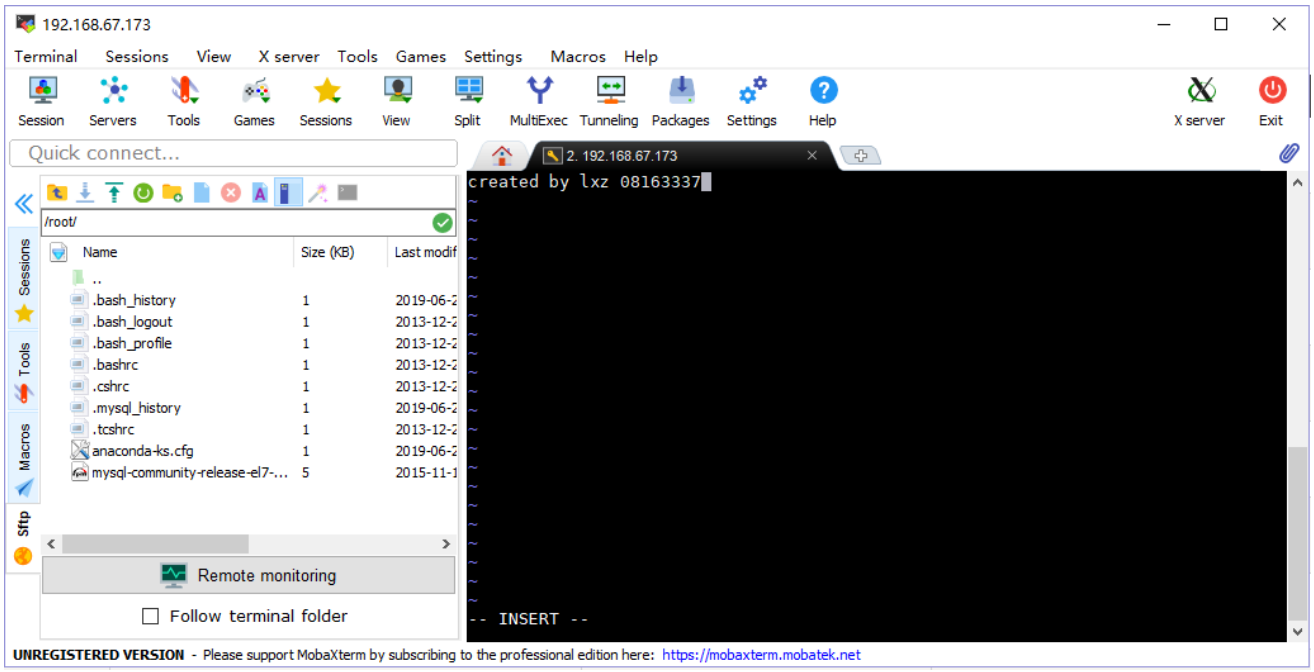
`cd [dirName]` Linux cd命令用于切换当前工作目录至 dirName(目录参数)。

- "~" 也表示为 home 目录 的意思
- "." 则是表示目前所在的目录
- ".." 则表示目前目录位置的上一层目录。

`touch [-acfm][-d<日期时间>][-r<参考文件或目录>] [-t<日期时间>][--help][--version][文件或目录…]` Linux touch命令用于修改文件或者目录的时间属性，包括存取时间和更改时间。若文件不存在，系统会建立一个新的文件。

```
[root@localhost ~]# mkdir test
[root@localhost ~]# cd test
[root@localhost test]# touch lxz
[root@localhost test]# ls
lxz
```

`vi` linux下文本编辑器，insert进行编辑，esc退出编辑模式， `:wq` 保存退出



`cat [-AbeEnstTuv] [--help] [--version] fileName` cat 命令用于连接文件并打印到标准输出设备上。

- -n 或 --number：由 1 开始对所有输出的行数编号。
- -b 或 --number-nonblank：和 -n 相似，只不过对于空白行不编号。
- -s 或 --squeeze-blank：当遇到有连续两行以上的空白行，就代换为一行的空白行。
- -v 或 --show-nonprinting：使用 ^ 和 M- 符号，除了 LFD 和 TAB 之外。
- -E 或 --show-ends : 在每行结束处显示 $。
- -T 或 --show-tabs: 将 TAB 字符显示为 ^I。
- -A, --show-all：等价于 -vET。
- -e：等价于"-vE"选项；
- -t：等价于"-vT"选项；

`cmp [-clsv][-i <字符数目>][--help][第一个文件][第二个文件]` Linux cmp命令用于比较两个文件是否有差异。

当相互比较的两个文件完全一样时，则该指令不会显示任何信息。若发现有所差异，预设会标示出第一个不同之处的字符和列数编号。若不指定任何文件名称或是所给予的文件名为"-"，则cmp指令会从标准输入设备读取数据。

`diff [-abBcdefHilnNpPqrstTuvwy][-<行数>][-C <行数>][-D <巨集名称>][-I <字符或字符串>][-S <文件>][-W <宽度>][-x <文件或目录>][-X <文件>][--help][--left-column][--suppress-common-line][文件或目录1][文件或目录2]`
Linux diff命令用于比较文件的差异。

diff以逐行的方式，比较文本文件的异同处。如果指定要比较目录，则diff会比较目录中相同文件名的文件，但不会比较其中子目录。

```
[root@localhost test]# vi lxz
[root@localhost test]# cat lxz
This is a test file
created by lxz 08163337
[root@localhost test]# cat -n lxz > lxz2
[root@localhost test]# cat lxz2
     1  This is a test file
     2  created by lxz 08163337
[root@localhost test]# cmp lxz lxz2
lxz lxz2 differ: byte 1, line 1
[root@localhost test]# diff lxz lxz2
1,2c1,2
< This is a test file
< created by lxz 08163337
---
>      1        This is a test file
>      2        created by lxz 08163337
[root@localhost test]#
```

`stat [文件或目录]` Linux stat命令用于显示inode内容。

`file [-bcLvz][-f <名称文件>][-m <魔法数字文件>...][文件或目录...]` Linux file命令用于辨识文件类型。

```
[root@localhost test]# stat lxz
  File: ‘lxz’
  Size: 44              Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 50596369    Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Context: unconfined_u:object_r:admin_home_t:s0
Access: 2019-06-21 20:57:39.818609451 +0800
Modify: 2019-06-21 20:53:21.831617078 +0800
Change: 2019-06-21 20:53:21.838617078 +0800
 Birth: -
[root@localhost test]# stat lxz2
  File: ‘lxz2’
  Size: 58              Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 50596367    Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Context: unconfined_u:object_r:admin_home_t:s0
Access: 2019-06-21 20:58:18.991608293 +0800
Modify: 2019-06-21 20:58:14.883608414 +0800
Change: 2019-06-21 20:58:14.883608414 +0800
 Birth: -
[root@localhost test]# file lxz
lxz: ASCII text
```

`find path -option [-print] [-exec -ok command] {} \;` Linux find命令用来在指定目录下查找文件。任何位于参数之前的字符串都将被视为欲查找的目录名。如果使用该命令时，不设置任何参数，则find命令将在当前目录下查找子目录与文件。并且将查找到的子目录和文件全部进行显示。

`mv [options] source dest` `mv [options] source... directory` Linux mv命令用来为文件或目录改名、或将文件或目录移入其它位置。

- -i: 若指定目录已有同名文件，则先询问是否覆盖旧文件;
- -f: 在mv操作要覆盖某已有的目标文件时不给任何指示;

```
[root@localhost test]# find . lxz
.
./lxz
./lxz2
lxz
[root@localhost test]# mv /var/www/html/* .
[root@localhost test]# ls
info.php  lxz  lxz2  sql.php
```

`paste [-s][-d <间隔字符>][--help][--version][文件...]` Linux paste命令用于合并文件的列。

`patch [-bceEflnNRstTuvZ][-B <备份字首字符串>][-d <工作目录>][-D <标示符号>][-F <监别列数>][-g <控制数值>][-i <修补文件>][-o <输出文件>][-p <剥离层级>][-r <拒绝文件>][-V <备份方式>][-Y <备份字首字符串>][-z <备份字尾字符串>][--backup-if -mismatch][--binary][--help][--nobackup-if-mismatch][--verbose][原始文件 <修补文件>] 或 path [-p <剥离层级>] < [修补文件]` Linux patch命令用于修补文件。

patch指令让用户利用设置修补文件的方式，修改，更新原始文件。倘若一次仅修改一个文件，可直接在指令列中下达指令依序执行。如果配合修补文件的方式则能一次修补大批文件，这也是Linux系统核心的升级方法之一。

`split [--help][--version][-<行数>][-b <字节>][-C <字节>][-l <行数>][要切割的文件][输出文件名]` Linux split命令用于将一个文件分割成数个。

```
[root@localhost test]# paste lxz lxz2
This is a test file          1  This is a test file
created by lxz 08163337      2  created by lxz 08163337
[root@localhost test]# split -1 lxz
[root@localhost test]# ls
info.php  lxz  lxz2  sql.php  xaa  xab
[root@localhost test]# cat xaa
This is a test file
[root@localhost test]# cat xab
created by lxz 08163337
[root@localhost test]#
```

`cp [options] source dest` `cp [options] source... directory` Linux cp命令主要用于复制文件或目录。

- -a：此选项通常在复制目录时使用，它保留链接、文件属性，并复制目录下的所有内容。其作用等于dpR参数组合。
- -d：复制时保留链接。这里所说的链接相当于Windows系统中的快捷方式。
- -f：覆盖已经存在的目标文件而不给出提示。
- -i：与-f选项相反，在覆盖目标文件之前给出提示，要求用户确认是否覆盖，回答"y"时目标文件将被覆盖。
- -p：除复制文件的内容外，还把修改时间和访问权限也复制到新文件中。
- -r：若给出的源文件是一个目录文件，此时将复制该目录下所有的子目录和文件。
- -l：不复制文件，只是生成链接文件。

`which [文件...]` Linux which命令用于查找文件。

which指令会在环境变量$PATH设置的目录里查找符合条件的文件。

`whereis [-bfmsu][-B <目录>...][-M <目录>...][-S <目录>...][文件...]` Linux whereis命令用于查找文件。

该指令会在特定目录中查找符合条件的文件。这些文件应属于原始代码、二进制文件，或是帮助文件。

该指令只能用于查找二进制文件、源代码文件和man手册页，一般文件的定位需使用locate命令。

```
[root@localhost test]# cp lxz lxzz
[root@localhost test]# ls
info.php  lxz  lxz2  lxzz  sql.php  xaa  xab
[root@localhost test]# diff lxz lxzz
[root@localhost test]# which bash
/usr/bin/bash
[root@localhost test]# whereis bash
bash: /usr/bin/bash /usr/share/man/man1/bash.1.gz
[root@localhost test]#
```

`grep [-abcEFGhHilLnqrsvVwxy][-A<显示列数>][-B<显示列数>][-C<显示列数>][-d<进行动作>][-e<范本样式>][-f<范本文件>][--help][范本样式][文件或目录...]` Linux grep命令用于查找文件里符合条件的字符串。

grep指令用于查找内容包含指定的范本样式的文件，如果发现某文件的内容符合所指定的范本样式，预设grep指令会把含有范本样式的那一列显示出来。若不指定任何文件名称，或是所给予的文件名为"-"，则grep指令会从标准输入设备读取数据。

`join [-i][-a<1或2>][-e<字符串>][-o<格式>][-t<字符>][-v<1或2>][-1<栏位>][-2<栏位>][--help][--version][文件1][文件2]` Linux join命令用于将两个文件中，指定栏位内容相同的行连接起来。

找出两个文件中，指定栏位内容相同的行，并加以合并，再输出到标准输出设备。

```
[root@localhost test]# grep 08163337 l*
lxz:created by lxz 08163337
lxz2:     2     created by lxz 08163337
lxzz:created by lxz 08163337
```

`wc [-clw][--help][--version][文件...]` Linux wc命令用于计算字数。

利用wc指令我们可以计算文件的Byte数、字数、或是列数，若不指定文件名称、或是所给予的文件名为"-"，则wc指令会从标准输入设备读取数据。

`chmod [-cfvR] [--help] [--version] mode file...` Linux/Unix 的文件调用权限分为三级 : 文件拥有者、群组、其他。利用 chmod 可以藉以控制文件如何被他人所调用。

- u 表示该文件的拥有者，g 表示与该文件的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示这三者皆是。
- + 表示增加权限、- 表示取消权限、= 表示唯一设定权限。
- r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该文件是个子目录或者该文件已经被设定过为可执行。

- -c : 若该文件权限确实已经更改，才显示其更改动作
- -f : 若该文件权限无法被更改也不要显示错误讯息
- -v : 显示权限变更的详细资料
- -R : 对目前目录下的所有文件与子目录进行相同的权限变更(即以递回的方式逐个变更)
- --help : 显示辅助说明
- --version : 显示版本

```
[root@localhost test]# wc lxz
 2  9 44 lxz
[root@localhost test]# wc sql.php
  9  17 193 sql.php
[root@localhost test]# chmod 777 lxz
[root@localhost test]# ll
total 28
-rw-r--r--. 1 root root  21 Jun 21 18:24 info.php
-rwxrwxrwx. 1 root root  44 Jun 21 20:53 lxz
-rw-r--r--. 1 root root  58 Jun 21 20:58 lxz2
-rw-r--r--. 1 root root  44 Jun 21 21:37 lxzz
-rw-r--r--. 1 root root 193 Jun 21 18:43 sql.php
-rw-r--r--. 1 root root  20 Jun 21 21:33 xaa
-rw-r--r--. 1 root root  24 Jun 21 21:33 xab
[root@localhost test]#
```

`df [选项]... [FILE]...` Linux df命令用于显示目前在Linux系统上的文件系统的磁盘使用情况统计。

`du [-abcDhHklmsSx][-L <符号连接>][-X <文件>][--block-size][--exclude=<目录或文件>][--max-depth=<目录层数>][--help][--version][目录或文件]` Linux du命令用于显示目录或文件的大小。

du会显示指定的目录或文件所占用的磁盘空间。

```
[root@localhost test]# df -l
Filesystem              1K-blocks    Used Available Use% Mounted on
/dev/mapper/centos-root  17811456 1837660  15973796  11% /
devtmpfs                   485780       0    485780   0% /dev
tmpfs                      497948       0    497948   0% /dev/shm
tmpfs                      497948    7912    490036   2% /run
tmpfs                      497948       0    497948   0% /sys/fs/cgroup
/dev/sda1                 1038336  135376    902960  14% /boot
tmpfs                       99592       0     99592   0% /run/user/0
[root@localhost test]# du -a
4       ./lxz
4       ./lxz2
4       ./info.php
4       ./sql.php
4       ./xaa
4       ./xab
4       ./lxzz
28      .
[root@localhost test]#
```

`ps [options] [--help]` Linux ps命令用于显示当前进程 (process) 的状态。

`top [-] [d delay] [q] [c] [S] [s] [i] [n] [b]` Linux top命令用于实时显示 process 的动态。

```
[root@localhost test]# ps
   PID TTY          TIME CMD
 22544 pts/1    00:00:00 bash
 25412 pts/1    00:00:00 ps
[root@localhost test]# top
top - 22:01:48 up  4:44,  4 users,  load average: 0.00, 0.01, 0.05
Tasks: 118 total,   1 running, 117 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   995896 total,    67448 free,   302580 used,   625868 buff/cache
KiB Swap:  2097148 total,  2097140 free,        8 used.   465396 avail Mem

   PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
     1 root      20   0  128164   6736   4176 S  0.3  0.7   0:05.75 systemd
     2 root      20   0       0      0      0 S  0.0  0.0   0:00.02 kthreadd
     3 root      20   0       0      0      0 S  0.0  0.0   0:01.45 ksoftirqd/0
     5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
     7 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
     8 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
     9 root      20   0       0      0      0 S  0.0  0.0   0:02.82 rcu_sched
    10 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 lru-add-drain
    11 root      rt   0       0      0      0 S  0.0  0.0   0:00.18 watchdog/0
    13 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
    14 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 netns
    15 root      20   0       0      0      0 S  0.0  0.0   0:00.01 khungtaskd
    16 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 writeback
    17 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kintegrityd
    18 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 bioset
    19 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 bioset
    20 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 bioset
    21 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kblockd
    22 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 md
    23 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 edac-poller
    24 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 watchdogd
    30 root      20   0       0      0      0 S  0.0  0.0   0:00.41 kswapd0
    31 root      25   5       0      0      0 S  0.0  0.0   0:00.00 ksmd
    32 root      39  19       0      0      0 S  0.0  0.0   0:00.21 khugepaged
    33 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 crypto
    41 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kthrotld
    43 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kmpath_rdacd
    44 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kaluad
    45 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kpsmoused
    47 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 ipv6_addrconf
    60 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 deferwq
    91 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kauditd
  1710 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 nfit
  1737 root       0 -20       0      0      0 S  0.0  0.0   0:00.01 ata_sff
  1749 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 mpt_poll_0
  1752 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 mpt/0
  1810 root      20   0       0      0      0 S  0.0  0.0   0:00.00 scsi_eh_0
  1811 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 scsi_tmf_0
  1838 root      20   0       0      0      0 S  0.0  0.0   0:00.01 scsi_eh_1
  1846 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 scsi_tmf_1
  1850 root      20   0       0      0      0 S  0.0  0.0   0:00.00 scsi_eh_2
  1852 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 scsi_tmf_2
  2081 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 ttm_swap
  2085 root     -51   0       0      0      0 S  0.0  0.0   0:00.00 irq/16-vmwgfx
  2982 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kdmflush
  2983 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 bioset
```

`free [-bkmotV][-s <间隔秒数>]` Linux free命令用于显示内存状态。

free指令会显示内存的使用情况，包括实体内存，虚拟的交换文件内存，共享内存区段，以及系统核心使用的缓冲区等。

`id [-gGnru][--help][--version][用户名称]` Linux id命令用于显示用户的ID，以及所属群组的ID。

id会显示用户以及所属群组的实际与有效ID。若两个ID相同，则仅显示实际ID。若仅指定用户名称，则显示目前用户的ID。

- -g或--group　显示用户所属群组的ID。

- -G或--groups　显示用户所属附加群组的ID。
- -n或--name　显示用户，所属群组或附加群组的名称。
- -r或--real　显示实际ID。
- -u或--user　显示用户ID。
- -help　显示帮助。
- -version　显示版本信息。

```
[root@localhost test]# free
              total        used        free      shared  buff/cache   available
Mem:         995896      302496       67524        7980      625876      465476
Swap:       2097148           8     2097140
[root@localhost test]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

`tar [-ABcdgGhiklmMoOpPrRsStuUvwWxzZ][-b <区块数目>][-C <目的目录>][-f <备份文件>][-F <Script文件>][-K <文件>][-L <媒体容量>][-N <日期时间>][-T <范本文件>][-V <卷册名称>][-X <范本文件>][-<设备编号><存储密度>][--after-date=<日期时间>][--atime-preserve][--backuup=<备份方式>][--checkpoint][--concatenate][--confirmation][--delete][--exclude=<范本样式>][--force-local][--group=<群组名称>][--help][--ignore-failed-read][--new-volume-script=<Script文件>][--newer-mtime][--no-recursion][--null][--numeric-owner][--owner=<用户名称>][--posix][--erve][--preserve-order][--preserve-permissions][--record-size=<区块数目>][--recursive-unlink][--remove-files][--rsh-command=<执行指令>][--same-owner][--suffix=<备份字尾字符串>][--totals][--use-compress-program=<执行指令>][--version][--volno-file=<编号文件>][文件或目录...]` Linux tar命令用于备份文件。

tar是用来建立，还原备份文件的工具程序，它可以加入，解开备份文件内的文件。

```
[root@localhost test]# tar -czvf test.tar.gz lxz
lxz
[root@localhost test]# tar -tzvf test.tar.gz
-rwxrwxrwx root/root        44 2019-06-21 20:53 lxz
[root@localhost test]# tar -xzvf test.tar.gz
lxz
[root@localhost test]#
```

1.3 管道符组合使用命令

利用Linux所提供的管道符"|"将两个命令隔开，管道符左边命令的输出就会作为管道符右边命令的输入。连续使用管道意味着第一个命令的输出会作为 第二个命令的输入，第二个命令的输出又会作为第三个命令的输入，依此类推。

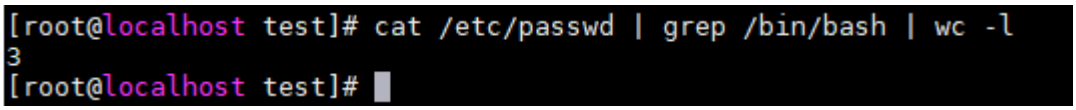
- 管道命令只处理前一个命令正确输出，不处理错误输出。
- 管道命令右边命令，必须能够接收标准输入流命令才行。

将文件lxz中的小写字母全部转换成大写字母 `cat lxz |tr a-z A-Z`

```
[root@localhost test]# cat lxz
This is a test file
created by lxz 08163337
[root@localhost test]# cat lxz |tr a-z A-Z
THIS IS A TEST FILE
CREATED BY LXZ 08163337
[root@localhost test]#
```

```
cat /etc/passwd | grep /bin/bash | wc -l
```

这条命令使用了两个管道，利用第一个管道将cat命令（显示passwd文件的内容）的输出送给grep命令，grep命令找出含有“/bin /bash”的所有行；第二个管道将grep的输出送给wc命令，wc命令统计出输入中的行数。这个命令的功能在于找出系统中有多少个用户使用bash。



管道命令与重定向区别：

- 左边的命令应该有标准输出 | 右边的命令应该接受标准输入

  左边的命令应该有标准输出 > 右边只能是文件

  左边的命令应该需要标准输入 < 右边只能是文件

- 管道触发两个子进程执行"|"两边的程序；而重定向是在一个进程内执行

2 (30分) shell编程

> 要求：
>
> (1)知道如何执行shell程序
>
> (2)在shell脚本中要体现条件控制（如if结构和条件分支）
>
> (3)在shell脚本中要体现循环（for,while和until循环）
>
> (4)掌握shell程序的调试
>
> 评分标准：所写程序代码，不符合下列要求的扣1-20分
>
> (1)有效语句不少于20行(符合编程规范)
>
> (2)if语句不少于4个
>
> (3)循环嵌套不少于1层

2.1 脚本编写

脚本功能：读取文件，判断是否有段落内部的不当分行并输出这个段落应该连续的内容。

```
#!/bin/bash

ARGCOUNT=1       # 全局变量，一个参数
OFF=0            # 标记状态
ON=1
E_WRONGARGS=85
file="$1"        # 输入目标文件名
lineno=1         # 行数从一开始
Flag=$OFF        # 空行标记

if [ $# -ne "$ARGCOUNT" ] # 输入参数错误时报错退出
then
  echo "Usage: `basename $0` FILENAME"

  exit $E_WRONGARGS
```

```
  fi

  file_read ()     # 读取文件函数
  {
  while read line  # 循环读取每一行
  do

    if [[ "$line" =~ ^[a-z] ]]
    then
      if [[ $Flag -eq $ON ]]
       then
       echo -n "$lineno::    " # 前一行为空行，且这一行以小写字母开头，输出行号和内容
       echo "$line"
      fi
    fi


    if [[ "$line" =~ ^$ ]]
       then
       Flag=$ON   #空行，设置标志
    else
       Flag=$OFF
    fi

    ((lineno++))

  done
  } < $file  # 将文件作为函数输入

  file_read

  exit $?
```

2.2 运行及调试

运行遇到问题 `/bin/bash: bad interpreter: Text file busy` 是因为直接使用shell工具自带的文件传输和编辑软件编写所以占用文件，所以下载lsof命令进行查看。

```
[root@localhost test]# ./hnt.sh
-bash: ./hnt.sh: /bin/bash: bad interpreter: Text file busy
[root@localhost test]# yum install lsof
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.163.com
 * extras: mirrors.163.com
 * updates: mirrors.163.com
Resolving Dependencies
--> Running transaction check
---> Package lsof.x86_64 0:4.87-6.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch                Version                Repository                Size
================================================================================
Installing:
 lsof                   x86_64              4.87-6.el7             base                      331 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 331 k
Installed size: 927 k
Is this ok [y/d/N]: y
Downloading packages:
lsof-4.87-6.el7.x86_64.rpm                                  | 331 kB  00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : lsof-4.87-6.el7.x86_64                                      1/1
  Verifying  : lsof-4.87-6.el7.x86_64                                      1/1

Installed:
  lsof.x86_64 0:4.87-6.el7

Complete!
[root@localhost test]# lsof | grep hnt.sh
sftp-serv 22539        root    3w      REG              253,0      3455   50596373 /root/test/hnt.sh
sftp-serv 22539        root    4r      REG              253,0      3455   50596373 /root/test/hnt.sh
```

`kill 22539` 停止进程后再运行成功

编辑测试文件，随便复制了一篇文章，将某些完整的段落中故意加入空行。

```
[root@localhost test]# vi example
[root@localhost test]# cat example
“Misogyny” seems a straightforward word. In dictionaries, it is “hatred of women”. In its etymology are the Greek verb misein, to

hate, and gyne, women. The word, like the sentiment, has been around for a long time. Euripides, an ancient Greek playwright, was called a misogynes, or woman-hater. (“Well, in his tragedies,
yes,” his peer Sophocles is said to have quipped, “but in bed at any rate he was a philogynes.”) The first known use of “misogynist” in English is from 1620—by a female group counter-attacking
 against a screed called “The Arraignment of Lewd, Idle, Froward [sic], and Unconstant Women”.

In fact, very few interesting words are quite so stable. As they are used, their meanings drift. Furthermore, they need not remain true to their etymological roots, a belief known to linguists
 as the “etymological fallacy”. The word “person”, for instance, comes from the Latin for “mask”; the word “tragedy” may derive from the Greek for “goat-song”. Over time, words evolve.

Much of that process is random. But it is also possible to make a conscious effort

to shift how a word is used. One such bid is under way for “misogyny”. For decades, feminists have expanded its connotations beyond the idea of “hatred of women”. Recently Kate Manne, a philos
opher at Cornell University, has become the voice of that campaign. She thinks the notion of a hatred for all women deep in the psychology of some men is philosophically untestable. In any cas
e, few men, she says, really hate all women. Instead of misogyny meaning something men feel, she says it should designate something women face.

Ms Manne distinguishes between sexist beliefs and systemic prejudice. For instance, the idea that women have certain innate characteristics (being loving and nurturing, say) and natural roles
 that derive from them (wife, mother) is sexist. It is when women fail to behave as they “should” that her version of misogyny comes into play—when men punish them for being too sexually active
 (or not enough), for neglecting their domestic responsibilities or for claiming “male” roles such as leadership. Her misogyny is the enforcement structure of sexism.

In her recent book “Down Girl”, Ms Manne argues for an “ameliorative” approach to concepts (one she draws from another philosopher, Sally Haslanger), whereby they are made fit for philosophica
l scrutiny. The vindictive psychology of some men is beyond such analysis, but the expectations widely imposed on women, and how non-conformists are treated, can be probed, and maybe even chan
ged.

What words mean is generally determined another way: most linguists believe that they simply mean what people use them to mean. As virtually all modern lexicographers acknowledge, dictionaries
 are there to register actual usage, not to tell the mass of people that they are deploying a word incorrectly. If philosophers or activists want dictionaries to include a new meaning, they ha
ve to get people to use the word that way.

Sometimes they succeed. In 2012 Julia Gillard, Australia’s prime minister, gave her renowned “misogyny speech”, lambasting her rival Tony Abbott for referring to Ms Gillard “making an honest w
oman of herself”, and for posing by a sign reading “ditch the witch”. Traditionalists pounced; Mr Abbott didn’t hate all women, they said, so Ms Gillard obviously didn’t know what misogyny mea
nt. In response, Macquarie, an Australian dictionary publisher, expanded its definition of the word to include “entrenched prejudice against women”.

There are other ways to wage a social struggle on the lexical front. Inventing a word is one; Ms Manne has written about “himpathy”, which she uses to describe outbreaks of disproportionate co
ncern for the future of a man accused of harassment, rape or other violence towards women. The term is pointed and memorable, and is spreading online.

Repurposing an existing word is harder; the inertia of the older meaning must be overcome. But this can be done, as (more intentionally than Ms Gillard) theorists and activists managed with “q
ueer”. Whether inventing or repurposing words, in refusing to

kowtow to inherited concepts Ms Manne is emulating Friedrich Nietzsche, who said that philosophers “must no longer accept concepts as a gift, nor merely purify and polish them, but first make
 and create them, present them and make them convincing”. Sound argument is needed to persuade other philosophers of such intellectual leaps; to enlist the wider world, a compelling vocabulary
 is vital.
[root@localhost test]# chmod +x chaifen.sh
[root@localhost test]# ./chaifen.sh example
3::   hate, and gyne, women. The word, like the sentiment, has been around for a long time. Euripides, an ancient Greek playwright, was called a misogynes, or woman-hater. (“Well, in his trage
dies, yes,” his peer Sophocles is said to have quipped, “but in bed at any rate he was a philogynes.”) The first known use of “misogynist” in English is from 1620—by a female group counter-att
acking against a screed called “The Arraignment of Lewd, Idle, Froward [sic], and Unconstant Women”.
9::   to shift how a word is used. One such bid is under way for “misogyny”. For decades, feminists have expanded its connotations beyond the idea of “hatred of women”. Recently Kate Manne, a
philosopher at Cornell University, has become the voice of that campaign. She thinks the notion of a hatred for all women deep in the psychology of some men is philosophically untestable. In a
ny case, few men, she says, really hate all women. Instead of misogyny meaning something men feel, she says it should designate something women face.
23::   kowtow to inherited concepts Ms Manne is emulating Friedrich Nietzsche, who said that philosophers “must no longer accept concepts as a gift, nor merely purify and polish them, but firs
t make and create them, present them and make them convincing”. Sound argument is needed to persuade other philosophers of such intellectual leaps; to enlist the wider world, a compelling voca
bulary is vital.
[root@localhost test]#
```

赋予可执行权限 `chmod +x chaifen.sh` 带上参数运行shell脚本 `./chaifen.sh example` 发现脚本输出了段落中的异常空行信息。一个更简短的例子：

```
[root@localhost test]# cat chaifen
This is line one of an example paragraph, bla, bla, bla.
This is line two, and line three should follow on next line, but

there is a blank line separating the two parts of the paragraph.
[root@localhost test]# ./chaifen.sh chaifen
4::   there is a blank line separating the two parts of the paragraph.
[root@localhost test]#
```

接下来进行调试

- -n 只读取shell脚本，但不实际执行
- -x 进入跟踪方式，显示所执行的每一条命令
- -c "string" 从strings中读取命令
- -v 一边执行脚本，一边将执行过的脚本命令打印到标准错误输出

“-n”可用于测试shell脚本是否存在语法错误，但不会实际执行命令。在shell脚本编写完成之后，实际执行之前，首先使用“-n”选项来测试脚本是否存在语法错误是一个很好的习惯。因为某些shell脚本在执行时会对系统环境产生影响，比如生成或移动文件等，如果在实际执行才发现语法错误，将不得不手工做一些系统环境的恢复工作才能继续测试这个脚本。

```
[root@localhost test]# sh -n ./chaifen.sh
[root@localhost test]# sh -v ./chaifen.sh chaifen
#!/bin/bash
# find-splitpara.sh
#  Finds split paragraphs in a text file,
#+ and tags the line numbers.


ARGCOUNT=1       # Expect one arg.
OFF=0            # Flag states.
ON=1
E_WRONGARGS=85

file="$1"        # Target filename.
lineno=1         # Line number. Start at 1.
Flag=$OFF        # Blank line flag.

if [ $# -ne "$ARGCOUNT" ]
then
  echo "Usage: `basename $0` FILENAME"
  exit $E_WRONGARGS
fi

file_read ()     # Scan file for pattern, then print line.
{
while read line
do

  if [[ "$line" =~ ^[a-z] ]]
    then
    if [[ $Flag -eq $ON ]]
     then  # Line begins with lowercase character, following blank line.
     echo -n "$lineno::   "
     echo "$line"
    fi
  fi


  if [[ "$line" =~ ^$ ]]
     then       #  If blank line,
     Flag=$ON   #+ set flag.
  else
     Flag=$OFF
  fi

  ((lineno++))

done
} < $file  # Redirect file into function's stdin.

file_read
4::   there is a blank line separating the two parts of the paragraph.


exit $?
```

“-c”选项使shell解释器从一个字符串中而不是从一个文件中读取并执行shell命令。当需要临时测试一小段脚本的执行结果时，可以使用这个选项，如下所示：`sh -c 'a=1;b=2;let c=$a+$b;echo "c=$c"'`

"-x"选项可用来跟踪脚本的执行，是调试shell脚本的强有力工具。“-x”选项使shell在执行脚本的过程中把它实际执行的每一个命令行显示出来，并且在行首显示一个"+"号。 "+"号后面显示的是经过了变量替换之后的命令行的内容，有助于分析实际执行的是什么命令。 “-x”选项使用起来简单方便，可以轻松对付大多数的shell调试任务,应把其当作首选的调试手段。

```
[root@localhost test]# sh -x ./chaifen.sh chaifen
+ ARGCOUNT=1
+ OFF=0
+ ON=1
+ E_WRONGARGS=85
+ file=chaifen
+ lineno=1
+ Flag=0
+ '[' 1 -ne 1 ']'
+ file_read
+ read line
+ [[ This is line one of an example paragraph, bla, bla, bla. =~ ^[a-z] ]]
+ [[ This is line one of an example paragraph, bla, bla, bla. =~ ^$ ]]
+ Flag=0
+ (( lineno++ ))
+ read line
+ [[ This is line two, and line three should follow on next line, but =~ ^[a-z] ]]
+ [[ This is line two, and line three should follow on next line, but =~ ^$ ]]
+ Flag=0
+ (( lineno++ ))
+ read line
+ [[ '' =~ ^[a-z] ]]
+ [[ '' =~ ^$ ]]
+ Flag=1
+ (( lineno++ ))
+ read line
+ [[ there is a blank line separating the two parts of the paragraph. =~ ^[a-z] ]]
+ [[ 1 -eq 1 ]]
+ echo -n '4::   '
4::   + echo 'there is a blank line separating the two parts of the paragraph.'
there is a blank line separating the two parts of the paragraph.
+ [[ there is a blank line separating the two parts of the paragraph. =~ ^$ ]]
+ Flag=0
+ (( lineno++ ))
+ read line
+ exit 0
[root@localhost test]#
```

3 (40分) 在Linux下构建LAMP服务器

> 要求：
>
> (1)掌握构建LAMP的创建方法。
>
> (2)编写测试网页，测试运行环境.
>
> 评分标准：实验必须有运行结果截图，否则扣5-15分。

3.1 搭建LAMP

使用yum安装LAMP软件包，执行 `yum install httpd httpd-devel mysql mysql-server mysql-devel php php-mysql php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstring php-snmp php-soap curl curl-devel php-bcmath`

出现错误 `mysql-server没有可用包`

解决方法

- 下载mysql的repo源 `wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm`
- 安装mysql-community-release-el7-5.noarch.rpm包 `rpm -ivh mysql-community-release-el7-5.noarch.rpm`

启动mysql后设置mysql密码

```
service mysql start
mysql -u root
mysql>USE mysql;
mysql>UPDATE user SET Password=PASSWORD('ringfall') WHERE user='root';
mysql>FLUSH PRIVILEGES;
```

允许远程登录mysql数据库

```
mysql -u root -p
mysql>grant all privileges on *.* to 'root'@'%' identified by 'ringfall' with grant option;
```

防火墙设置开启80、443端口

```
firewall-cmd --permanent --zone=public  --add-service=http
firewall-cmd --permanent --zone=public  --add-service=https
firewall-cmd --reload
```

服务开机自启动

```
systemctl enable httpd
systemctl enable mysqld
```

启动服务后可通过 `systemctl status httpd` 命令查看服务器状态

```
[root@localhost ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2019-06-21 18:20:04 CST; 9s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 8654 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─8654 /usr/sbin/httpd -DFOREGROUND
           ├─8655 /usr/sbin/httpd -DFOREGROUND
           ├─8656 /usr/sbin/httpd -DFOREGROUND
           ├─8657 /usr/sbin/httpd -DFOREGROUND
           ├─8658 /usr/sbin/httpd -DFOREGROUND
           └─8659 /usr/sbin/httpd -DFOREGROUND

Jun 21 18:20:04 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Jun 21 18:20:04 localhost.localdomain httpd[8654]: AH00558: httpd: Could not reliably determine the server's fully...sage
Jun 21 18:20:04 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
```

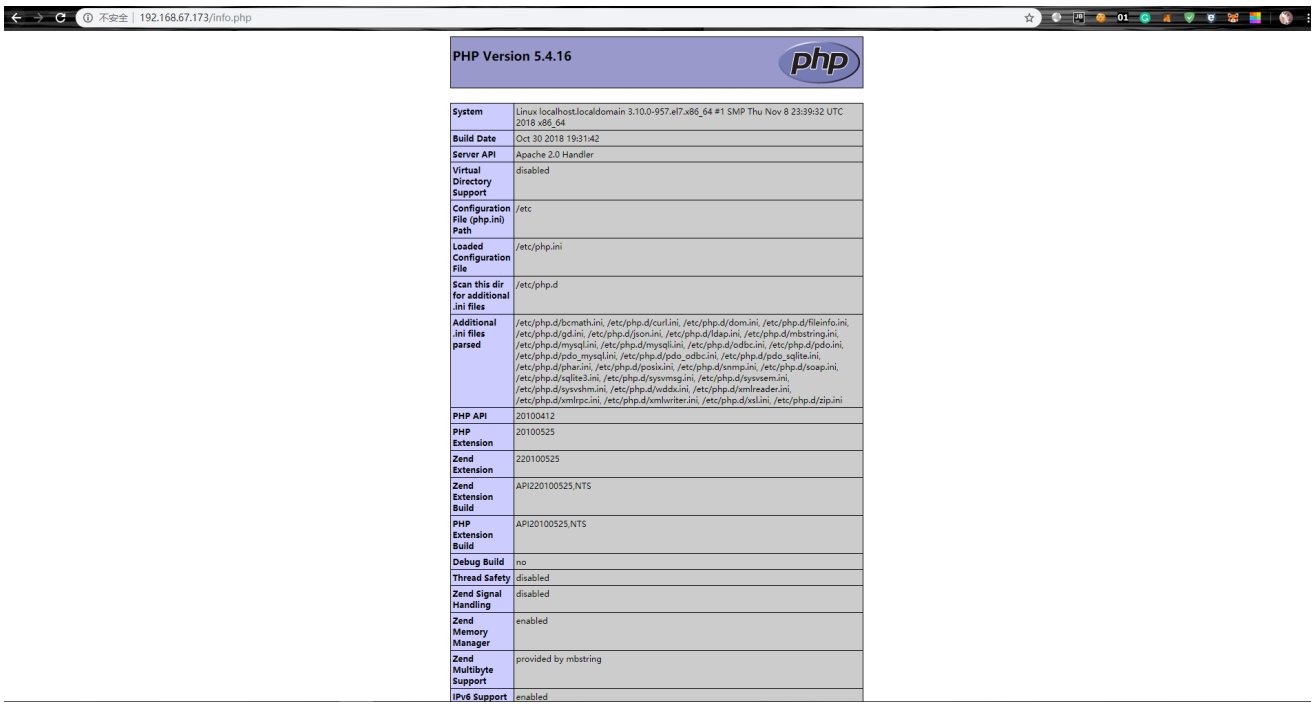
3.2 测试

直接访问ip为Apache默认界面

编写php测试界面

```php
<?php
    phpinfo();
?>
```

访问出现phpinfo信息



编写数据库连接测试界面

```php
<?php
    $mysqli = new mysqli('localhost','root','ringfall');
    if(!$mysqli)  {
        echo"database error";
    }
    else{
        echo"php env successful";
    }
    $mysqli->close();
?>
```

访问发现数据库连接成功

3.3 构建基本PHP登录网站

在数据库中新建数据库和表，插入新的数据

```
mysql> create database rf;
Query OK, 1 row affected (0.01 sec)

mysql> create table rf (id int,name varchar(10),password varchar(10));
ERROR 1046 (3D000): No database selected
mysql> use rf;
Database changed
mysql> create table rf (id int,name varchar(10),password varchar(10));
Query OK, 0 rows affected (0.12 sec)

mysql> insert into rf values(1,"ringfall","ringfall")
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> select * from rf;
+------+----------+----------+
| id   | name     | password |
+------+----------+----------+
|    1 | ringfall | ringfall |
+------+----------+----------+
1 row in set (0.00 sec)
```

使用php编写基本登录、验证和欢迎界面（只实现基本功能，较简略）

```html
<html>
<head>
<title>2333</title>
<meta http-equiv="content-type" content="text/html;charset=utf-8"/>
</head>
<body >
<center>
<h1>LOGIN HERE</h1><br/>
<form action="check.php" method="post">
ID:<input type="text" name="id" /><br/>
PASSWORD:<input type="password"  name="password" /><br/>
<?php
  if(isset($_GET['err'])&&$_GET['err']==1)
  {echo 'wrong  id  or  password';}
?>
<br/>
<input type="submit" value="submit">
</form>
</center>
</body>
</html>
```

```php
<?php
  $id=$_POST['id'];
  $password=$_POST['password'];

  $conn=mysql_connect("127.0.0.1","root","lxz");
  if(!$conn){die("连接失败".mysql_error());}

  mysql_select_db("rfdb",$conn) or die(mysql_error());

  mysql_query("set names utf8");

  $sql="select name,password from rf where id='$id'";
  $res=mysql_query($sql,$conn);

  var_dump($res);

  if($row=mysql_fetch_assoc($res))
  {
    if($row['password']==md5($password))
    {
      $name=$row['name'];
      header("Location:welcome.php?name=$name");
      mysql_free_result($res);
      mysql_close($conn);
      exit();
    }
  }
  header("Location:login.php?err=1");
  mysql_free_result($res);
  mysql_close($conn);
?>
```

```php
<html>
<center><h1>WELCOME~
<?php
$name=$_GET['name'];
echo "$name"?>
</h1></center>
</html>
```

访问登录界面时输入id和密码，错误时会显示 `wrong id or password`

LOGIN HERE

ID:1
PASSWORD:••••••••
wrong id or password
submit

输入正确进入欢迎页面显示用户名

WELCOME~ ringfall