東北大學秦皇島分校
Northeastern University at Qinhuangdao

Linux 操作系统与内核分析

实验报告

院　　别	计算机与通信工程学院
专业名称	计算机科学与技术
班级学号	20188068，180235
学生姓名	孔天欣

2021 年 4 月 22 日

目　　录

1 实验一 Linux 基本命令

1.1 实验名称

Linux 基本命令

1.2 实验内容

1. 查看 Linux 的内核版本

```
[root@izbp1icdyevcrfsbmd4woaz /]# uname -a
Linux izbp1icdyevcrfsbmd4woaz 3.10.0-514.26.2.el7.x86_64 #1 SMP Tue Jul 4 15:04:05 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
```

2. 查看网卡信息，可以看到 IP 地址是 172.16.199.141

```
[root@izbp1icdyevcrfsbmd4woaz /]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.199.141  netmask 255.255.240.0  broadcast 172.16.207.255
        ether 00:16:3e:11:ea:79  txqueuelen 1000  (Ethernet)
        RX packets 14680169  bytes 7040388982 (6.5 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13610260  bytes 4928876799 (4.5 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1  (Local Loopback)
        RX packets 1059  bytes 72855 (71.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1059  bytes 72855 (71.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

3. 搜索以 asswd 作为后缀的文件

```
[root@izbp1icdyevcrfsbmd4woaz /]# find / -name "?asswd" | more
/usr/bin/passwd
/etc/passwd
/etc/pam.d/passwd
/etc/mosquitto/passwd
```

4. 查看文件 passwd 的内容

```
[root@izbp1icdyevcrfsbmd4woaz /]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
```

5. 通过管道过滤查找关键字“mysql”

```
[root@izbp1icdyevcrfsbmd4woaz /]# cat /etc/passwd | grep "mysql"
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/false
```

6. 创建目录 test1 和 test2

```
[root@izbp1icdyevcrfsbmd4woaz /]# mkdir test1
[root@izbp1icdyevcrfsbmd4woaz /]# mkdir test2
[root@izbp1icdyevcrfsbmd4woaz /]# ls
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  test1  test2  tmp  usr  var
```

7. 创建文本文件 mytext

```
[root@izbp1icdyevcrfsbmd4woaz /]# cd ./test1
[root@izbp1icdyevcrfsbmd4woaz test1]# touch mytext
[root@izbp1icdyevcrfsbmd4woaz test1]# ls
mytext
```

8. 编辑文件 mytext

```
[root@izbp1icdyevcrfsbmd4woaz test1]# vi mytext
[root@izbp1icdyevcrfsbmd4woaz test1]# cat mytext
hello world!
```

9. 复制文件

```
[root@izbp1icdyevcrfsbmd4woaz test1]# cp mytext /test2
[root@izbp1icdyevcrfsbmd4woaz test1]# ls /test2
mytext
[root@izbp1icdyevcrfsbmd4woaz test1]# mv /test2/mytext /test2/mytext2
[root@izbp1icdyevcrfsbmd4woaz test1]# ls /test2
mytext2
```

10. 删除/test1/mytext 文件

```
[root@izbp1icdyevcrfsbmd4woaz test1]# rm -f mytext
[root@izbp1icdyevcrfsbmd4woaz test1]# ls
```

11. 彻底删除/test1 目录

```
[root@izbp1icdyevcrfsbmd4woaz test1]# cd ../
[root@izbp1icdyevcrfsbmd4woaz /]# rm -r test1
rm: remove directory 'test1'? y
[root@izbp1icdyevcrfsbmd4woaz /]# ls
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  test2  tmp  usr  var
```

12. 强制删除/test2 中所有文件及目录

```
[root@izbp1icdyevcrfsbmd4woaz /]# rm -rf test2/
[root@izbp1icdyevcrfsbmd4woaz /]# ls
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
```

13. 新建用户 testuser

```
[root@izbp1icdyevcrfsbmd4woaz /]# passwd testuser
Changing password for user testuser.
New password:
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

14. 切换并测试用户

```
[root@izbp1icdyevcrfsbmd4woaz /]# su testuser
[testuser@izbp1icdyevcrfsbmd4woaz /]$ sudo mkdir /test

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for testuser:
testuser is not in the sudoers file.  This incident will be reported.
```

15. 修改用户权限，在系统管理员权限下将 testuser 用户添加至“wheel”组。

```
[root@izbp1icdyevcrfsbmd4woaz ~]# usermod -a -G wheel testuser
[root@izbp1icdyevcrfsbmd4woaz ~]# su testuser
[testuser@izbp1icdyevcrfsbmd4woaz root]$ sudo mkdir /test
[sudo] password for testuser:
[testuser@izbp1icdyevcrfsbmd4woaz root]$ ls
ls: cannot open directory .: Permission denied
[testuser@izbp1icdyevcrfsbmd4woaz root]$ ;s /
bash: syntax error near unexpected token `;'
[testuser@izbp1icdyevcrfsbmd4woaz root]$ ls /
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  test  tmp  usr  var
```

16. 查看用户组，可以看到 testuser 在 wheel 组。

```
[root@izbp1icdyevcrfsbmd4woaz ~]# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
kmem:x:9:
wheel:x:10:testuser
```

17. 压缩文件 test.tar

```
[root@izbp1icdyevcrfsbmd4woaz /]# tar cvf test.tar test/
test/
[root@izbp1icdyevcrfsbmd4woaz /]# ls /
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  test  test.tar  tmp  usr  var
```

18. 解压文件 test.tar

```
[root@izbp1icdyevcrfsbmd4woaz /]# rm -rf test
[root@izbp1icdyevcrfsbmd4woaz /]# tar xvf test.tar
test/
[root@izbp1icdyevcrfsbmd4woaz /]# ls
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  test  test.tar  tmp  usr  var
```

19. 压缩文件 test.tar.gz

```
[root@izbp1icdyevcrfsbmd4woaz /]# rm -f test.tar
[root@izbp1icdyevcrfsbmd4woaz /]# tar zvcf test.tar.gz test/
test/
```

18. 解压文件 test.tar.gz

```
[root@izbp1icdyevcrfsbmd4woaz /]# rm -r /test
rm: remove directory '/test'? y
[root@izbp1icdyevcrfsbmd4woaz /]# tar zxvf test.tar.gz
test/
[root@izbp1icdyevcrfsbmd4woaz /]# ls
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  test  test.tar.gz  tmp  usr  var
```

1.3 实验总结

通过本次实验，本人初步掌握了 Linux 的基本命令，包括查看内核和网卡信息、进行目录和文件的生成和删除，压缩和解压以及进行用户的增删和修改权限等功能，了解了更多关于 Linux 的终端操作技巧。

2 实验二 Linux 系统管理

2.1 实验名称

Linux 系统管理

2.2 实验内容

1. 在虚拟机使用 ifconfig 查看网卡信息，网卡名为 ens33

```
[strutnut@localhost ~]$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.100  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::d9a2:b87:92e9:64ee  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:ee:88:08  txqueuelen 1000  (Ethernet)
        RX packets 12270  bytes 17601581 (16.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1080  bytes 80563 (78.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. 查看网卡配置文件路径，ifcfg-ens33 即网卡文件

```
[strutnut@localhost ~]$ ls /etc/sysconfig/network-scripts/
ifcfg-ens33   ifdown-isdn     ifup          ifup-plip     ifup-tunnel
ifcfg-lo      ifdown-post     ifup-aliases  ifup-plusb    ifup-wireless
ifdown        ifdown-ppp      ifup-bnep     ifup-post     init.ipv6-global
ifdown-bnep   ifdown-routes   ifup-eth      ifup-ppp      network-functions
ifdown-eth    ifdown-sit      ifup-ib       ifup-routes   network-functions-ipv6
ifdown-ib     ifdown-Team     ifup-ippp     ifup-sit
ifdown-ippp   ifdown-TeamPort ifup-ipv6     ifup-Team
ifdown-ipv6   ifdown-tunnel   ifup-isdn     ifup-TeamPort
[strutnut@localhost ~]$ vi /etc/sysconfig/network-scripts/ifcfg-ens33
[strutnut@localhost ~]$
```

3. 修改文件，配置静态 IP 为 192.168.0.100

```
strutnut@localhost:/home/strutnut
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
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=7d06c45e-3170-4e9d-9558-c0cdff3725a4
DEVICE=ens33
ONBOOT=yes

IPADDR=192.168.0.100
NETMASK=255.255.255.0
GATEWAY=192.168.0.1
DNS1=8.8.8.8
DNS2=114.114.114.114
~
~
~
~
-- INSERT --
```

4. 重启网络服务

```
[strutnut@localhost ~]$ service network restart
Restarting network (via systemctl):                        [  确定  ]
```

5. 查看系统开放端口

```
[root@localhost strutnut]# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1312/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1525/master
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN      5456/sshd: strutnut
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1/systemd
tcp        0      0 192.168.122.1:53        0.0.0.0:*               LISTEN      1970/dnsmasq
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1310/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN      1312/cupsd
tcp6       0      0 ::1:25                  :::*                    LISTEN      1525/master
tcp6       0      0 ::1:6010                :::*                    LISTEN      5456/sshd: strutnut
tcp6       0      0 :::111                  :::*                    LISTEN      1/systemd
tcp6       0      0 :::22                   :::*                    LISTEN      1310/sshd
udp        0      0 0.0.0.0:996             0.0.0.0:*                           824/rpcbind
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           827/avahi-daemon: r
udp        0      0 192.168.122.1:53        0.0.0.0:*                           1970/dnsmasq
udp        0      0 0.0.0.0:67              0.0.0.0:*                           1970/dnsmasq
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1/systemd
udp        0      0 0.0.0.0:43224           0.0.0.0:*                           827/avahi-daemon: r
udp        0      0 127.0.0.1:323           0.0.0.0:*                           850/chronyd
udp6       0      0 :::996                  :::*                                824/rpcbind
udp6       0      0 :::111                  :::*                                1/systemd
udp6       0      0 ::1:323                 :::*                                850/chronyd
```

6. 查看端口 111 所对应服务

```
[root@localhost strutnut]# cat /etc/services | grep -w 111
sunrpc          111/tcp         portmapper rpcbind      # RPC 4.0 portmapper TCP
sunrpc          111/udp         portmapper rpcbind      # RPC 4.0 portmapper UDP
```

7. 查看进程

```
[root@localhost strutnut]# ps
  PID TTY          TIME CMD
 5505 pts/2    00:00:00 su
 5514 pts/2    00:00:00 bash
 5597 pts/2    00:00:00 ps
```

8. 查看详细进程信息

```
[root@localhost strutnut]# ps -aux
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.2  0.2 193940  7100 ?        Ss   10:50   0:05 /usr/lib/systemd/systemd --switched-root --system --deserialize 2
root          2  0.0  0.0      0     0 ?        S    10:50   0:00 [kthreadd]
root          4  0.0  0.0      0     0 ?        S<   10:50   0:00 [kworker/0:0H]
root          6  0.1  0.0      0     0 ?        S    10:50   0:03 [ksoftirqd/0]
root          7  0.0  0.0      0     0 ?        S    10:50   0:00 [migration/0]
root          8  0.0  0.0      0     0 ?        S    10:50   0:00 [rcu_bh]
root          9  0.1  0.0      0     0 ?        S    10:50   0:03 [rcu_sched]
root         10  0.0  0.0      0     0 ?        S<   10:50   0:00 [lru-add-drain]
root         11  0.0  0.0      0     0 ?        S    10:50   0:00 [watchdog/0]
root         12  0.0  0.0      0     0 ?        S    10:50   0:00 [watchdog/1]
root         13  0.0  0.0      0     0 ?        S    10:50   0:00 [migration/1]
root         14  0.0  0.0      0     0 ?        S    10:50   0:00 [ksoftirqd/1]
root         16  0.0  0.0      0     0 ?        S<   10:50   0:00 [kworker/1:0H]
root         17  0.0  0.0      0     0 ?        S    10:50   0:00 [watchdog/2]
root         18  0.0  0.0      0     0 ?        S    10:50   0:00 [migration/2]
root         19  0.0  0.0      0     0 ?        S    10:50   0:00 [ksoftirqd/2]
root         21  0.0  0.0      0     0 ?        S<   10:50   0:00 [kworker/2:0H]
root         23  0.0  0.0      0     0 ?        S    10:50   0:00 [kdevtmpfs]
```

9. 按内存使用排序（升序）

```
[root@localhost strutnut]# ps -auxw --sort=rss
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          2  0.0  0.0      0     0 ?        S    10:50   0:00 [kthreadd]
root          4  0.0  0.0      0     0 ?        S<   10:50   0:00 [kworker/0:0H]
root          6  0.1  0.0      0     0 ?        S    10:50   0:03 [ksoftirqd/0]
root          7  0.0  0.0      0     0 ?        S    10:50   0:00 [migration/0]
root          8  0.0  0.0      0     0 ?        S    10:50   0:00 [rcu_bh]
root          9  0.1  0.0      0     0 ?        S    10:50   0:03 [rcu_sched]
root         10  0.0  0.0      0     0 ?        S<   10:50   0:00 [lru-add-drain]
root         11  0.0  0.0      0     0 ?        S    10:50   0:00 [watchdog/0]
root         12  0.0  0.0      0     0 ?        S    10:50   0:00 [watchdog/1]
```

10. 按内存使用排序（降序）

```
[root@localhost strutnut]# ps -auxw --sort=-rss
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
strutnut   2581  4.5  7.0 3684860 202788 ?      Sl   10:58   1:08 /usr/bin/gnome-shell
root       3285  3.1  5.6 593260 162180 ?       SN   10:59   0:44 /usr/bin/python /usr/share/PackageKit/helpers/yum/yumBackend.py get-updates r
strutnut   3015  0.4  2.1 1423788 62276 ?       Sl   10:58   0:05 /usr/bin/gnome-software --gapplication-service
root       1523  1.4  1.6 343924 48464 tty1     Ssl+ 10:51   0:28 /usr/bin/X :0 -background none -noreset -audit 4 -verbose -auth /run/gdm/auth
strutnut   3231  0.7  1.1 766248 33800 ?        Sl   10:59   0:11 /usr/libexec/gnome-terminal-server
strutnut   2868  0.1  1.0 900452 30408 ?        Sl   10:58   0:02 nautilus-desktop --force
root        951  0.0  1.0 359024 29752 ?        Ssl  10:50   0:01 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid
strutnut   2676  0.0  0.9 893492 27244 ?        Sl   10:58   0:00 /usr/libexec/goa-daemon
strutnut   2925  0.8  0.9 641232 26756 ?        Sl   10:58   0:13 /usr/bin/vmtoolsd -n vmusr
strutnut   2972  0.1  0.7 1040640 20708 ?       Sl   10:58   0:02 /usr/libexec/evolution-addressbook-factory-subprocess --factory all --bus-nam
```

11. 按 CPU 使用排序（升序）

```
[root@localhost strutnut]# ps -auxw --sort=%cpu
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           2  0.0  0.0      0     0 ?        S    10:50   0:00 [kthreadd]
root           4  0.0  0.0      0     0 ?        S<   10:50   0:00 [kworker/0:0H]
root           7  0.0  0.0      0     0 ?        S    10:50   0:00 [migration/0]
root           8  0.0  0.0      0     0 ?        S    10:50   0:00 [rcu_bh]
root          10  0.0  0.0      0     0 ?        S<   10:50   0:00 [lru-add-drain]
root          11  0.0  0.0      0     0 ?        S    10:50   0:00 [watchdog/0]
root          12  0.0  0.0      0     0 ?        S    10:50   0:00 [watchdog/1]
root          13  0.0  0.0      0     0 ?        S    10:50   0:00 [migration/1]
root          14  0.0  0.0      0     0 ?        S    10:50   0:00 [ksoftirqd/1]
root          16  0.0  0.0      0     0 ?        S<   10:50   0:00 [kworker/1:0H]
root          17  0.0  0.0      0     0 ?        S    10:50   0:00 [watchdog/2]
root          18  0.0  0.0      0     0 ?        S    10:50   0:00 [migration/2]
root          19  0.0  0.0      0     0 ?        S    10:50   0:00 [ksoftirqd/2]
```

12. 按 CPU 使用排序（降序）

```
[root@localhost strutnut]# ps -auxw --sort=-%cpu
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
strutnut   2581  4.3  7.0 3684860 202788 ?      Sl   10:58   1:08 /usr/bin/gnome-shell
root       3285  3.0  5.6 593260 162180 ?       SN   10:59   0:44 /usr/bin/python /usr/share/PackageKit/helpers/yum/yumBackend.py
root       1523  1.3  1.6 343924 48464 tty1     Ssl+ 10:51   0:28 /usr/bin/X :0 -background none -noreset -audit 4 -verbose -auth
strutnut   2925  0.8  0.9 641232 26756 ?        Sl   10:58   0:13 /usr/bin/vmtoolsd -n vmusr
strutnut   3231  0.7  1.1 766248 33800 ?        Sl   10:59   0:11 /usr/libexec/gnome-terminal-server
root        829  0.4  0.2 324680  6908 ?        Ssl  10:50   0:09 /usr/bin/vmtoolsd
strutnut   2594  0.4  0.2 1285496 7380 ?        S<l  10:58   0:07 /usr/bin/pulseaudio --start --log-target=syslog
strutnut   3015  0.3  2.1 1423788 62276 ?       Sl   10:58   0:05 /usr/bin/gnome-software --gapplication-service
```

13. 查看系统进程信息

```
top - 11:25:07 up 34 min,  4 users,  load average: 0.00, 0.05, 0.20
Tasks: 229 total,   1 running, 228 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.9 us,  5.6 sy,  0.0 ni, 92.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  2859892 total,   186480 free,   956432 used,  1716980 buff/cache
KiB Swap:  2097148 total,  2097148 free,        0 used.  1642328 avail Mem

   PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
  5728 root      20   0  162020   2264   1528 R   5.9  0.1   0:00.03 top
     1 root      20   0  193940   7100   4236 S   0.0  0.2   0:05.36 systemd
     2 root      20   0       0      0      0 S   0.0  0.0   0:00.02 kthreadd
     4 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
     6 root      20   0       0      0      0 S   0.0  0.0   0:04.20 ksoftirqd/0
     7 root      rt   0       0      0      0 S   0.0  0.0   0:00.21 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   0:03.27 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   0:00.16 watchdog/0
    12 root      rt   0       0      0      0 S   0.0  0.0   0:00.02 watchdog/1
    13 root      rt   0       0      0      0 S   0.0  0.0   0:00.32 migration/1
    14 root      20   0       0      0      0 S   0.0  0.0   0:00.35 ksoftirqd/1
    16 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:0H
```

14. 终止进程 5974 并确认

```
[root@localhost strutnut]# ps -ef | grep 5974
root       5974   5514  0 11:29 pts/2    00:00:00 ./redis-cli
root       6115   6071  0 11:31 pts/3    00:00:00 grep --color=auto 5974
[root@localhost strutnut]# kill 5974
[root@localhost strutnut]# ps -ef | grep 5974
root       6117   6071  0 11:31 pts/3    00:00:00 grep --color=auto 5974
```

15. 查看磁盘使用情况

```
[root@localhost strutnut]# df -h
文件系统                 容量  已用  可用 已用% 挂载点
devtmpfs                 1.4G     0  1.4G    0% /dev
tmpfs                    1.4G     0  1.4G    0% /dev/shm
tmpfs                    1.4G   13M  1.4G    1% /run
tmpfs                    1.4G     0  1.4G    0% /sys/fs/cgroup
/dev/mapper/centos-root   17G  8.9G  8.2G   53% /
/dev/sda1               1014M  184M  831M   19% /boot
tmpfs                    280M   48K  280M    1% /run/user/1000
/dev/sr0                 4.4G  4.4G     0  100% /run/media/strutnut/CentOS 7 x86_64
tmpfs                    280M     0  280M    0% /run/user/0
```

16. 查看目录所占空间

```
[root@localhost strutnut]# du -s /home/strutnut
421916  /home/strutnut
[root@localhost strutnut]# du -sm /home/strutnut
413     /home/strutnut
```

2.3 实验总结

通过本次实验，本人掌握了 Linux 系统管理的相关命令，包括配置静态 IP，查看活跃进程以及关键词排序，终止进程以及查看磁盘使用情况这些功能，本人的 Linux 操作系统使用熟练度得以提升。

3 实验三 服务器配置与管理

3.1 实验名称

服务器配置与管理

3.2 实验内容

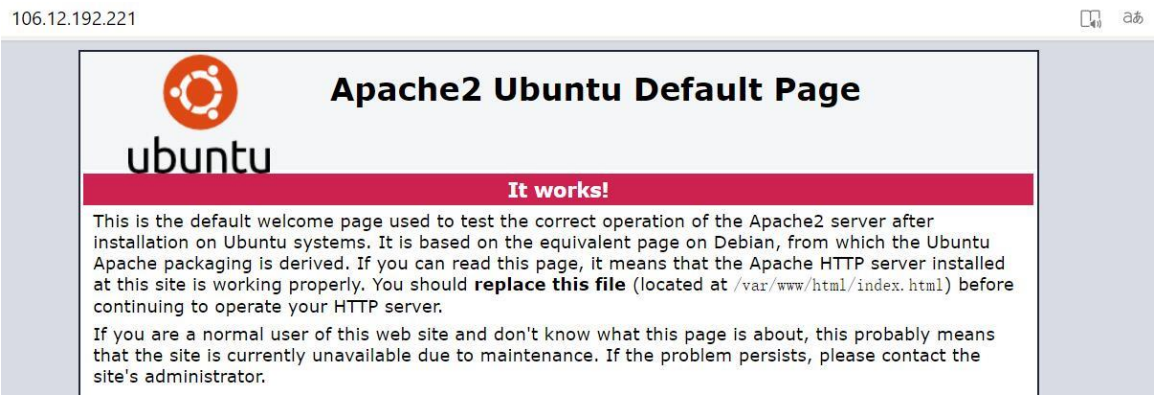
1. 在线安装 Apache 服务器

```
root@instance-gwa3tobu:/home# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 194 not upgraded.
Need to get 1,729 kB of archives.
After this operation, 6,985 kB of additional disk space will be used.
```

2. 手动启动 apache 服务

```
root@instance-gwa3tobu:/home# /etc/init.d/apache2 start
[ ok ] Starting apache2 (via systemctl): apache2.service.
```

3. 测试 apache



4. 安装 MariaDB 服务

```
root@instance-gwa3tobu:/home# apt-get install mariadb-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  mariadb-client
0 upgraded, 1 newly installed, 0 to remove and 220 not upgraded.
Need to get 12.8 kB of archives.
After this operation, 66.6 kB of additional disk space will be used.
Get:1 http://mirrors.baidubce.com/ubuntu bionic-updates/universe amd64 mariadb-client all 1:10.1.47-0ubuntu0.18.04.1 [12.8
Fetched 12.8 kB in 0s (794 kB/s)
Selecting previously unselected package mariadb-client.
(Reading database ... 100168 files and directories currently installed.)
Preparing to unpack .../mariadb-client_1%3a10.1.47-0ubuntu0.18.04.1_all.deb ...
Unpacking mariadb-client (1:10.1.47-0ubuntu0.18.04.1) ...
Setting up mariadb-client (1:10.1.47-0ubuntu0.18.04.1) ...
```

5. 测试数据库

```
root@instance-gwa3tobu:/home# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 46
Server version: 10.1.47-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

6. 安装 PHP

```
root@instance-gwa3tobu:/home# apt-get install php7.3 libapache2-mod-php7.3  php7.3-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  net-tools
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libargon2-1 libpcre2-8-0 libsodium23 php-common php7.3-cli php7.3-common php7.3-json php7.3-opcache php7.3-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php7.3 libargon2-1 libpcre2-8-0 libsodium23 php-common php7.3 php7.3-cli php7.3-common php7.3-json php7.3-mysql php7.3-opcache
  php7.3-readline
0 upgraded, 12 newly installed, 0 to remove and 10 not upgraded.
Need to get 4,046 kB of archives.
After this operation, 18.3 MB of additional disk space will be used.
```

7. 测试 PHP 页面

106.12.192.221/test.php

PHP Version 7.3.27-9+ubuntu18.04.1+deb.sury.org+1

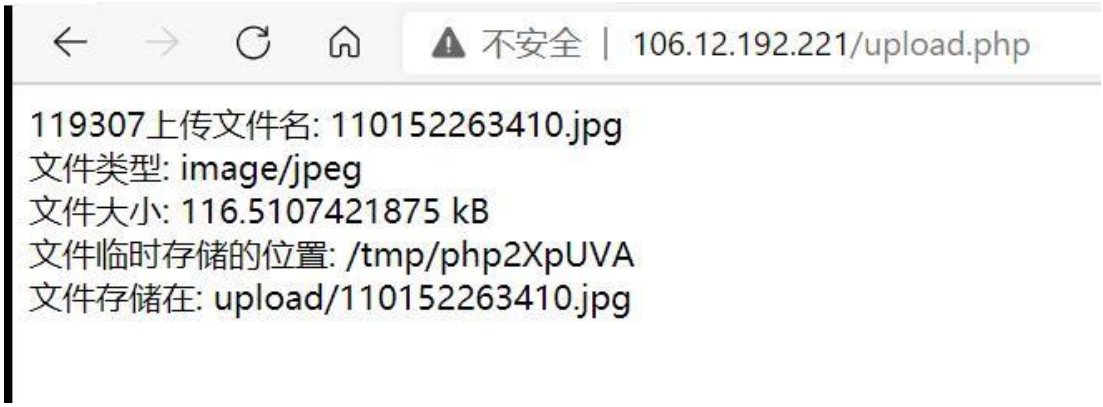
System	Linux instance-gwa3tobu 4.15.0-128-generic #131-Ubuntu SMP Wed Dec 9 06:57:35 UTC 2020 x86_64
Build Date	Feb 23 2021 15:10:08
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mysqli.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini

7. 测试上传文件页面

不安全 | 106.12.192.221/upload.html

Filename: 选择文件 未选择文件

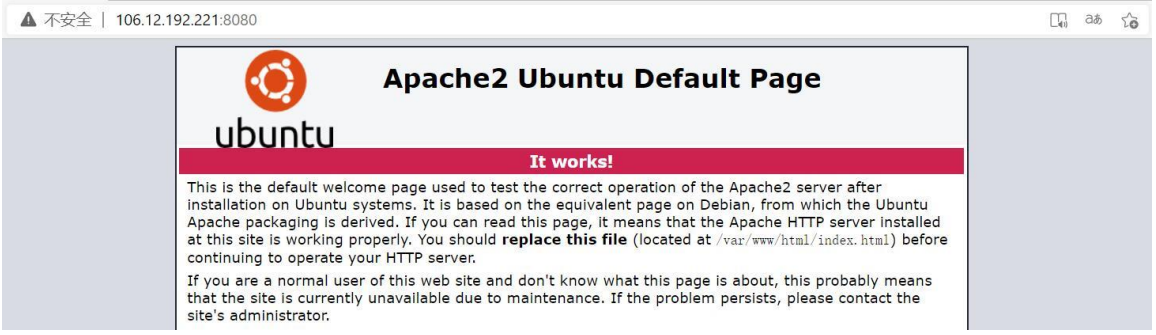
Submit

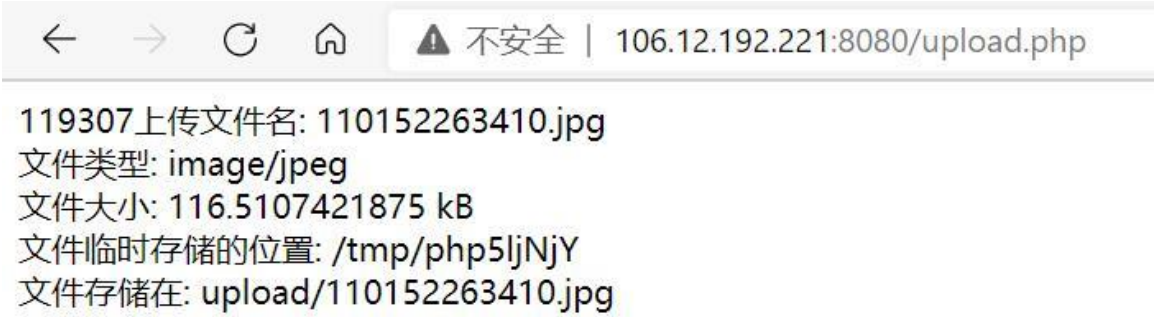


8. 修改默认端口和网站根目录测试，并上传文件

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080

<IfModule ssl_module>
        Listen 443
</IfModule>
```

```
<VirtualHost *:8080>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www2/html
```





3.3 实验总结

通过本次实验，本人成功学会了服务器的基本配置，包括 Apache、数据库 MariaDB、以及 PHP 环境的安装与配置，同时成功部署了部分网页文件在 Apache 中，并能够通过 IP 地址访问，初步实现了文件的上传功能以及 Apache 的参数配置方法。

4 实验四 Linux Shell 编程

4.1 实验名称

Linux Shell 编程

4.2 实验内容

1. helloworld.sh，会在终端输出 helloworld！

```
root@VM-0-7-ubuntu:/home/ubuntu/temp# bash helloworld.sh
Hello World !
root@VM-0-7-ubuntu:/home/ubuntu/temp# chmod 744 helloworld.sh
root@VM-0-7-ubuntu:/home/ubuntu/temp# ./helloworld.sh
Hello World !
```

2. system_info.sh，会在终端输出系统时间、系统持续运行时间、系统负载、内存使用等系统信息。

```
root@VM-0-7-ubuntu:/home/ubuntu/temp# bash system_info.sh
System time: 2021-04-04 09:00:23
Running time: up 1 week, 3 days, 16 hours, 59 minutes
Load average: 0.00 0.02 0.04
Used memory: 1.4Gi / 1.9Gi
```

3. network_monitor.sh，会在终端输出网卡输入输出数据流量，并根据流量判断网卡状态。

```
root@VM-0-7-ubuntu:/home/ubuntu/temp# bash network_monitor.sh
IP: 172.18.0.1
Input bytes1: 9675329148 Output bytes1: 10700563650
Input bytes2: 9675441886 Output bytes2: 10700694528
Network traffic is on the rise.
```

4. for 条件循环，会依次输出 1~5 的数字。

```
root@VM-0-7-ubuntu:/home/ubuntu/temp# bash loop.sh
The value is: 1
The value is: 2
The value is: 3
The value is: 4
The value is: 5
```

5. CPU_monitor.sh，会在终端显示 CPU 的型号，核数，以及 CPU 负载。

```
root@VM-0-7-ubuntu:/home/ubuntu/temp# bash CPU_monitor.sh&
[1] 2391467
root@VM-0-7-ubuntu:/home/ubuntu/temp# tail -f cpu_monitor.txt
model name      : Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz
cpu cores       : 1

Total data:
user nice system idle iowait irq softirq
2007948 12953 1163396 88283569 206783 0 144938
2007952 12953 1163399 88283760 206784 0 144939
2007958 12953 1163404 88283947 206784 0 144940
2007963 12953 1163407 88284137 206784 0 144941
2007966 12953 1163410 88284329 206784 0 144941
```

4.3 实验总结

通过本次实验，本人初步掌握了 Shell 编程的基本原理，掌握了 Shell 脚本的语法基础和运行方式，并成功运行了通过 Shell 脚本语言书写的一系列程序，进而对 Linux 命令的拓展和组合有了更加深入的理解。

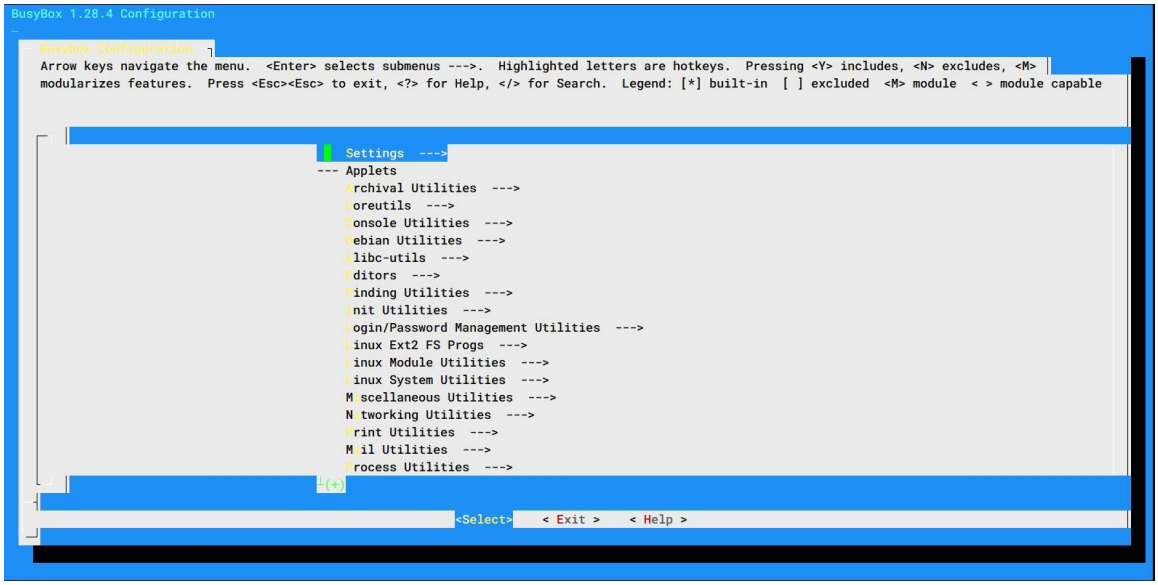
5 实验五 Linux 内核编译与运行

5.1 实验名称

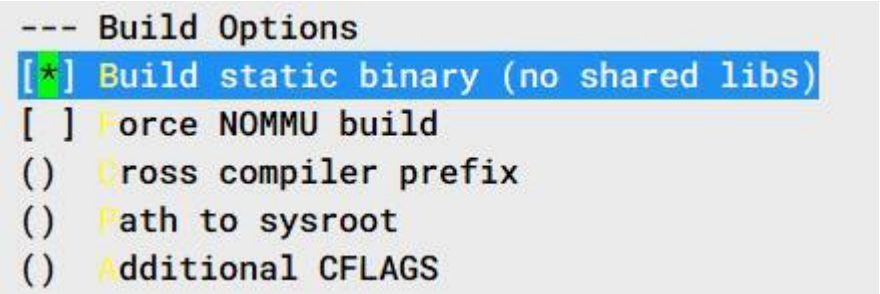
Linux 内核编译与运行

5.2 实验内容

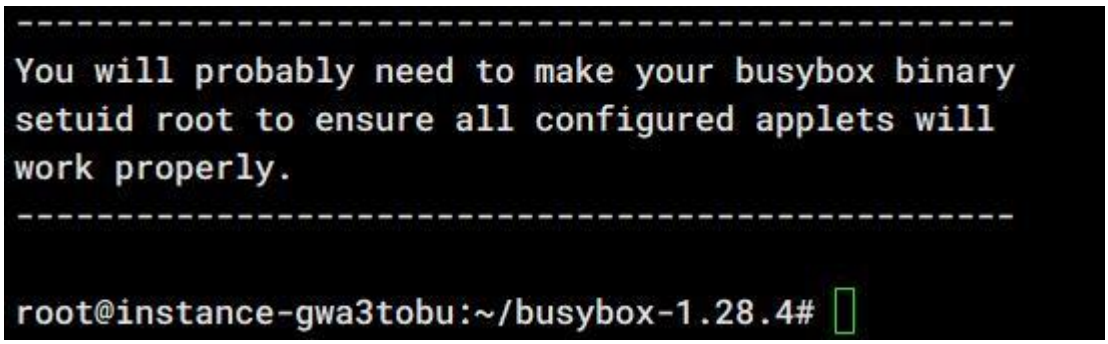
1. 图形化编译最小文件系统



2. 静态编译配置



3. make install



4. 生成_install 目录

```
root@instance-gwa3tobu:~/busybox-1.28.4# ls
0086-dpkg-fix-symlink-creation-closes-10941.patch    busybox_unstripped.out  init            Makefile.help           selinux
0122-nsenter-Rename-network-option-to-net.patch      Config.in               _install        make_single_applets.sh  shell
0123-nsenter-fix-parsing-of-t-S-and-G-options.patch  configs                 INSTALL         miscutils               size_single_applets.sh
applets                                              console-tools           klibc-utils     modutils                sysklogd
applets_sh                                           coreutils               libbb           networking              testsuite
arch                                                 debianutils             libpwdgrp       NOFORK_NOEXEC.lst       TODO
archival                                             docs                    LICENSE         printutils              TODO_unicode
AUTHORS                                              e2fsprogs               loginutils      procps                  util-linux
busybox                                              editors                 mailutils       qemu_multiarch_testing
busybox.links                                        examples                Makefile        README
busybox_unstripped                                   findutils               Makefile.custom runit
busybox_unstripped.map                               include                 Makefile.flags  scripts
```

5. 创建目录

```
root@instance-gwa3tobu:~/linux-5.4.5/_install# ls
bin  dev  etc  linuxrc  -p  sbin  usr
```

6. 创建并编辑 fstab,inittab,rcS 文件

```
root@instance-gwa3tobu:~/linux-5.4.5/_install/etc# ls
fstab  init.d  inittab
root@instance-gwa3tobu:~/linux-5.4.5/_install/etc# cat fstab
proc /proc proc defaults 0 0
tmpfs /tmp tmpfs defaults 0 0
sysfs /sys sysfs defaults 0 0
tempfs /dev tmpfs defaults 0 0
debugfs /sys/kernel/debug debugfs defaults 0 0
root@instance-gwa3tobu:~/linux-5.4.5/_install/etc# cat inittab
::sysinit:/etc/init.d/rcS
::respawn:-/bin/sh
::askfirst:-/bin/sh
::ctrlaltdel:/bin/umount -a -r
root@instance-gwa3tobu:~/linux-5.4.5/_install/etc# cat init.d/rcS
mkdir -p /proc
mkdir -p /tmp
mkdir -p /sys
mkdir -p /mnt
/bin/mount -a
mkdir -p /dev/pts
mount -t devpts devpts /dev/pts
echo /sbin/mdev > /proc/sys/kernel/hotplug
mdev -s
```

7. 创建节点

```
root@instance-gwa3tobu:~/linux-5.4.5/_install/dev# mknod console c 5
root@instance-gwa3tobu:~/linux-5.4.5/_install/dev# mknod null c 1 3
root@instance-gwa3tobu:~/linux-5.4.5/_install/dev# ls
console  null
```

8. 编译内核

```
root@instance-gwa3tobu:~/linux-5.4.5# export ARCH=arm
root@instance-gwa3tobu:~/linux-5.4.5# export CROSS_COMPILE=arm-linux-gnueabi-
root@instance-gwa3tobu:~/linux-5.4.5# make vexpress_defconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  HOSTCC  scripts/kconfig/confdata.o
  HOSTCC  scripts/kconfig/expr.o
  LEX     scripts/kconfig/lexer.lex.c
  YACC    scripts/kconfig/parser.tab.[ch]
  HOSTCC  scripts/kconfig/lexer.lex.o
  HOSTCC  scripts/kconfig/parser.tab.o
  HOSTCC  scripts/kconfig/preprocess.o
  HOSTCC  scripts/kconfig/symbol.o
  HOSTLD  scripts/kconfig/conf
#
# configuration written to .config
#
```

8. 路径配置

```
[ ] Enable deprecated sysfs features to support old userspace tools
[ ] Kernel->user space relay support (formerly relayfs)
[*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
(_install) Initramfs source file(s)
(0)   User ID to map to 0 (user root) (NEW)
(0)   Group ID to map to 0 (group root) (NEW)
```

```
(0x0) Compressed ROM boot loader BSS address
[ ] Use appended device tree blob to zImage (EXPERIMENTAL)
()  Default kernel command string
[ ] Kexec system call (EXPERIMENTAL)
[ ] Build kdump crash kernel (EXPERIMENTAL)
```

```
    Memory split (3G/1G user/kernel split)  --->
(8) Maximum number of CPUs (2-32)
-*- Support for hot-pluggable CPUs
[*] Support for the ARM Power State Coordination Interface (PSCI)
    Timer frequency (100 Hz)  --->
[ ] Compile the kernel in Thumb-2 mode
[*] Runtime patch udiv/sdiv instructions into __aeabi_{u}idiv()
-*- Use the ARM EABI to compile the kernel
[ ]   Allow old ABI binaries to run with this kernel (EXPERIMENTAL)
[*] High Memory Support
```

9. 内核编译成功

```
  SHIPPED arch/arm/boot/compressed/hyp-stub.S
  AS      arch/arm/boot/compressed/hyp-stub.o
  SHIPPED arch/arm/boot/compressed/lib1funcs.S
  AS      arch/arm/boot/compressed/lib1funcs.o
  SHIPPED arch/arm/boot/compressed/ashldi3.S
  AS      arch/arm/boot/compressed/ashldi3.o
  SHIPPED arch/arm/boot/compressed/bswapsdi2.S
  AS      arch/arm/boot/compressed/bswapsdi2.o
  LD      arch/arm/boot/compressed/vmlinux
  OBJCOPY arch/arm/boot/zImage
  Kernel: arch/arm/boot/zImage is ready
root@instance-gwa3tobu:~/linux-5.4.5#
```

```
root@instance-gwa3tobu:~/linux-5.4.5# make dtbs
  DTC     arch/arm/boot/dts/vexpress-v2p-ca5s.dtb
  DTC     arch/arm/boot/dts/vexpress-v2p-ca9.dtb
  DTC     arch/arm/boot/dts/vexpress-v2p-ca15-tc1.dtb
  DTC     arch/arm/boot/dts/vexpress-v2p-ca15_a7.dtb
```

10. 启动 QEMU

```
/ # pwd
/
/ # uname -a
Linux (none) 5.4.5 #1 SMP Sat Apr 10 12:42:05 CST 2021 armv7l GNU/Linux
/ # ls
bin      etc      mnt      sbin     tmp      ?p
dev      linuxrc  proc     sys      usr
/ # poweroff
The system is going down NOW!
Sent SIGTERM to all processes
Terminated
Sent SIGKILL to all processes
Requesting system poweroff
Flash device refused suspend due to active operation (state 20)
Flash device refused suspend due to active operation (state 20)
reboot: Power down
root@instance-gwa3tobu:~/linux-5.4.5#
```

5.3 实验总结

通过本次实验，本人成功掌握了 Linux 内核镜像的编译方法，同时还初步掌握了使用 BusyBox，QEMU 等工具，在 Ubuntu 系统中成功编译 Linux 内核，建立了 Linux 虚拟机，加深了对 Linux 内核的理解和掌握程度。

6 实验六 Linux 内核模块编程

6.1 实验名称

Linux 内核模块编程

6.2 实验内容

1. 编写.c 文件和 Makefile 文件并编译、查看文件

```
root@instance-gwa3tobu:~/cpb# ls
hello_module.c  Makefile
root@instance-gwa3tobu:~/cpb# make
make -C /lib/modules/4.15.0-128-generic/build M=/root/cpb modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-128-generic'
  CC [M]  /root/cpb/hello_module.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /root/cpb/hello_module.mod.o
  LD [M]  /root/cpb/hello_module.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-128-generic'
root@instance-gwa3tobu:~/cpb# ls
hello_module.c  hello_module.ko  hello_module.mod.c  hello_module.mod.o  hello_module.o  Makefile  modules.order  Module.symvers
```

2. file 命令检查编译模块

```
root@instance-gwa3tobu:~/cpb# file hello_module.ko
hello_module.ko: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), BuildID[sha1]=d3d0fbaa9a264c4e925e07e164b9c965931f1b6d, not stripped
```

3. modinfo 命令检查编译模块

```
root@instance-gwa3tobu:~/cpb# modinfo hello_module.ko
filename:       /root/cpb/hello_module.ko
alias:          hello
description:    hello kernel module
author:         Mr Yu
license:        GPL
srcversion:     43E4E354AB7BF7C3C668E81
depends:
retpoline:      Y
name:           hello_module
vermagic:       4.15.0-128-generic SMP mod_unload
```

4. insmod 命令插入模块并查看

```
root@instance-gwa3tobu:~/cpb# insmod hello_module.ko
root@instance-gwa3tobu:~/cpb# lsmod
Module                  Size  Used by
hello_module           16384  0
binfmt_misc            20480  1
btrfs                1155072  0
zstd_compress         163840  1 btrfs
```

5. 模块目录

```
root@instance-gwa3tobu:~/cpb# ls /sys/module/
8250          button          drm_kms_helper  ghash_clmulni_intel  ipv6      module      processor     serio_raw     tpm_tis         watchdog
acpi          cirrus          dynamic_debug   glue_helper          jfs       mousedev    psmouse       sg            tpm_tis_core    workqueue
acpi_cpufreq  configfs        edac_core       gpiolib_acpi         joydev    msdos       pstore        spurious      ttm             xen_acpi_processor
acpiphp       cpufreq         edd             hello_module         kdb       netpoll     pvpanic       srcutree      ufs             xen_blkfront
aesni_intel   cpuidle         efivars         hfs                  kernel    ntfs        qemu_fw_cfg   sr_mod        uhci_hcd        xen_netfront
```

6. 查看输出

```
nfined" name="/usr/bin/man" pid=11408 comm="apparmor_parser"
[600527.763554] hello_module: loading out-of-tree module taints kernel.
[600527.763619] hello_module: module verification failed: signature and/or required key missing - tainting kernel
[600527.763724] This is hello_module, welcome to Linux kernel
root@instance-gwa3tobu:~/cpb#
```

7. 卸载模块

```
[600527.763619] hello_module: module verification failed: signature and/or required key missing - tainting kernel
[600527.763724] This is hello_module, welcome to Linux kernel
[600761.446831] see you next time!
root@instance-gwa3tobu:~/cpb#
```

8. 模块参数默认值

```
[600527.763724] This is hello_module, welcome to Linux kernel
[600761.446831] see you next time!
[601129.895198] my linux kernel module init.
[601129.895199] module parameter = 10
root@instance-gwa3tobu:~/cpb2#
```

9. 赋值重新加载模块

```
[601214.744743] see you next time!
[601222.867143] my linux kernel module init.
[601222.867144] module parameter = 100
root@instance-gwa3tobu:~/cpb2#
```

6.2 实验总结

通过本次实验，本人成功掌握了简单的内核模块编译方法，并理解了模块的主要结构和内容。同时，还能够完成简单的模块参数编译，理解内核模块传参的正确方式，从而对 Linux 内核模块程序设计有了更加深入的理解。

7 实验七 Linux 内存管理

7.1 实验名称

Linux 内存管理

7.2 实验内容

1. make 编译

```
root@instance-gwa3tobu:~/mm# make
make -C /lib/modules/4.15.0-128-generic/build   M=/root/mm modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-128-generic'
  CC [M]  /root/mm/vma_test.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /root/mm/vma_test.mod.o
  LD [M]  /root/mm/vma_test.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-128-generic'
```

2. 查看 VMA 信息，可以看到第一个 VMA 区域的长度为 15437824 字节。

```
[1119080.950907]  Examining vma's for pid=12913, command=mysqldn
[1119080.950910] mm_struct addr = 0x000000007691c209
[1119080.950911] vmas: vma start end lengthn
[1119080.950913]      1:         401eea32 55805825f000 558059118000 15437824n
[1119080.950914]      2:         69a0ff2d 558059318000 5580593c2000   696320n
[1119080.950915]      3:         3f9a89d5 5580593c2000 558059474000   729088n
[1119080.950916]      4:         14edc7b7 558059474000 558059cf3000  8908800n
[1119080.950916]      5:         912c25ef 7f6d44400000 7f6d45800000 20971520n
[1119080.950917]      6:         b46a7faa 7f6d45bf5000 7f6d45bf6000     4096n
[1119080.950918]      7:         dae40db2 7f6d45bf6000 7f6d463f6000  8388608n
[1119080.950919]      8:         14414494 7f6d463f6000 7f6d463f7000     4096n
[1119080.950920]      9:         f6dba9ec 7f6d463f7000 7f6d46bf7000  8388608n
[1119080.950920]     10:         8b77bc05 7f6d46bf7000 7f6d46bf8000     4096n
[1119080.950921]     11:         f733836f 7f6d46bf8000 7f6d473f8000  8388608n
[1119080.950922]     12:         32513085 7f6d473f8000 7f6d473f9000     4096n
[1119080.950923]     13:         328647e8 7f6d473f9000 7f6d47bf9000  8388608n
[1119080.950923]     14:         1a18f0ff 7f6d47bf9000 7f6d47bfa000     4096n
[1119080.950924]     15:          2552f33 7f6d47bfa000 7f6d483fa000  8388608n
```

3. 使用 cat 查看 VMA 信息，可以看到第一个 VMA 区域的大小为 15076KB，和上图的 15437824 / 1024 = 15076 KB 一致，因此程序输出结果正确。

```
root@instance-gwa3tobu:~/mm# cat /proc/12913/smaps | head -n 30
55805825f000-558059118000 r-xp 00000000 fc:01 788407                     /usr/sbin/mysqld
Size:              15076 kB
KernelPageSize:        4 kB
MMUPageSize:           4 kB
Rss:                3024 kB
Pss:                3024 kB
Shared_Clean:          0 kB
Shared_Dirty:          0 kB
Private_Clean:      3024 kB
Private_Dirty:         0 kB
Referenced:         3024 kB
Anonymous:             0 kB
LazyFree:              0 kB
AnonHugePages:         0 kB
ShmemPmdMapped:        0 kB
Shared_Hugetlb:        0 kB
Private_Hugetlb:       0 kB
Swap:                  0 kB
SwapPss:               0 kB
Locked:                0 kB
VmFlags: rd ex mr mw me dw sd
```

7.3 实验总结

通过本次实验，本人成功了解了 Linux 内存空间地址的查看和管理方法，同时能够掌握部分 VMA 相关的操作，最后能够通过编写一个内核模块，遍历用户进程中的 VMA 来查看相关信息并验证其正确性。

8 实验八 Linux 设备驱动

8.1 实验名称

Linux 设备驱动

8.2 实验内容

1. 查看设备号

```
246 bsg
247 hmm_device
248 watchdog
249 rtc
250 dax
251 dimmctl
252 ndctl
253 tpm
254 gpiochip
300 my_cdev_driver

Block devices:
```

2. 执行测试程序

```
root@instance-gwa3tobu:~/lab8# ./test
84
104
105
115
32
105
115
32
109
121
root@instance-gwa3tobu:~/lab8#
```

3. 通过 dmesg 查看日志信息，可以看到，将 /dev/mycdev 读了 10 个字节到 buf 中。

```
[1206488.250976] mycdev driver is now starting!
[1206488.250981] register successfully!
[1206661.686901] reader: 10 bytes was read.
root@instance-gwa3tobu:~/lab8#
```

8.3 实验总结

通过本次实验，本人初步了解 Linux 设备管理的方式，同时能够理解 Linux 设备驱动框架的原理以及 Linux 设备驱动相关的 API 的功能和作用，最后能够编写出一个简单的字符设备驱动程序，实现基本的操作和方法。