

# 桂林理工大学大数据培训中心

## LINUX 测试题

### 一 • 选择题 (每题 3 分, 共计 30 分)

1.Linux 文件权限一共 10 位长度, 分成四段, 第三段表示的内容是( **C** )

- A. 文件类型
- B. 文件所有者的权限
- C. 文件所有者所在组的权限
- D. 其他用户的权限

2.以下描述哪个是不正确的? ( **D** )

- A. ps -aux 查看所有用户所有进程; ps -ef 查看子父进程之间的关系
- B. /etc/profile 文件对所有用户设置; ~/.bash\_profile 只对当前用户设置
- C. cmd1&&cat f1.txt > f2.txt, 该命令表示当命令 cmd1 成功执行, 则将 f1.txt 复制成 f2.txt
- D. kill -9 和 kill -15 后面跟进程号都可以杀死进程, kill -15 是强行杀死一个进程

3.如何删除目录/tmp 下的所有文件及子目录 ( **D** )。

- A、del /tmp/\*
- B、rm -rf /tmp
- C、rm -Ra /tmp/\*
- D、rm -rf /tmp/\*

4.电信用户报告说, 有一个号码为 35298286 的固定电话的话单市话费用计算错误, 请你协助在 LS\_CALL\_200605030052 这个文本话单文件中查找固定电话 35298286 在 5 月 3 日这天打出的所有

市话话单的原始信息, 请问选择下列的哪行命令可以满足用户提出的这个要求? ( **D** )

- A、find 35298286 LS\_CALL\_200605030052
- B、more 35298286 LS\_CALL\_200605030052
- C、cmp 35298286 LS\_CALL\_200605030052
- D、grep 35298286 LS\_CALL\_200605030052

5. 有一个备份程序 mybackup，需要在周一至周五下午 1 点和晚上 8 点各运行一次，下面哪

条 crontab 的项可以完成这项工作？ ( **C** )

- A. 0 13,20 \* \* 1,5 mybackup
- B. 0 13,20 \* \* 1,2,3,4,5 mybackup
- C. \* 13,20 \* \* 1-5 mybackup
- D. 0 13,20 1,5 \* \* mybackup

6.chmod a=rwx /opt/test 该命令修改权限设置的结果是什么？ ( **B** )

- A. 731
- B. 777
- C. 640
- D. 650

7.你使用命令“vim/etc/inittab 查看该文件的内容，你不小心改动了一些内容，为了防止系统出问

题，你不想保存所修改内容，你应该如何操作 ( **B** )

- A、在末行模式下，键入：wq
- B、在末行模式下，键入：q!
- C、在末行模式下，键入：x!
- D、在编辑模式下，键入“ESC”键直接退出 vim

8.如果想列出当前目录以及子目录下所有扩展名为.txt 的文件，那么您可以使用的命令是

( **B** )。

- A. ls \*.txt
- B. find ./ -name "\*.txt"
- C. ls -d .txt
- D. find ./ ".txt"

9.哪个目录存放用户密码信息 ( **B** )

- A、/boot
- B、/etc
- C、/var
- D、/dev

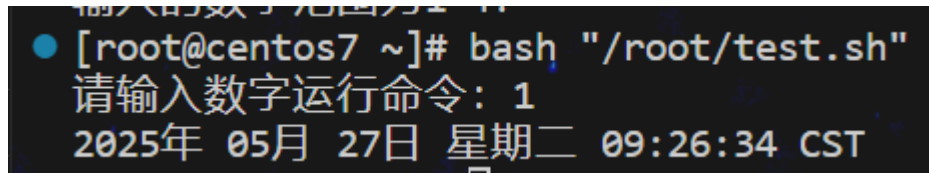
10.什么命令解压 tar 文件？ ( **B** )

- A. tar -czvf filename.tgz
- B. tar -xzvf filename.tgz
- C. tar -tzvf filename.tgz
- D. tar -dzvf filename.tgz

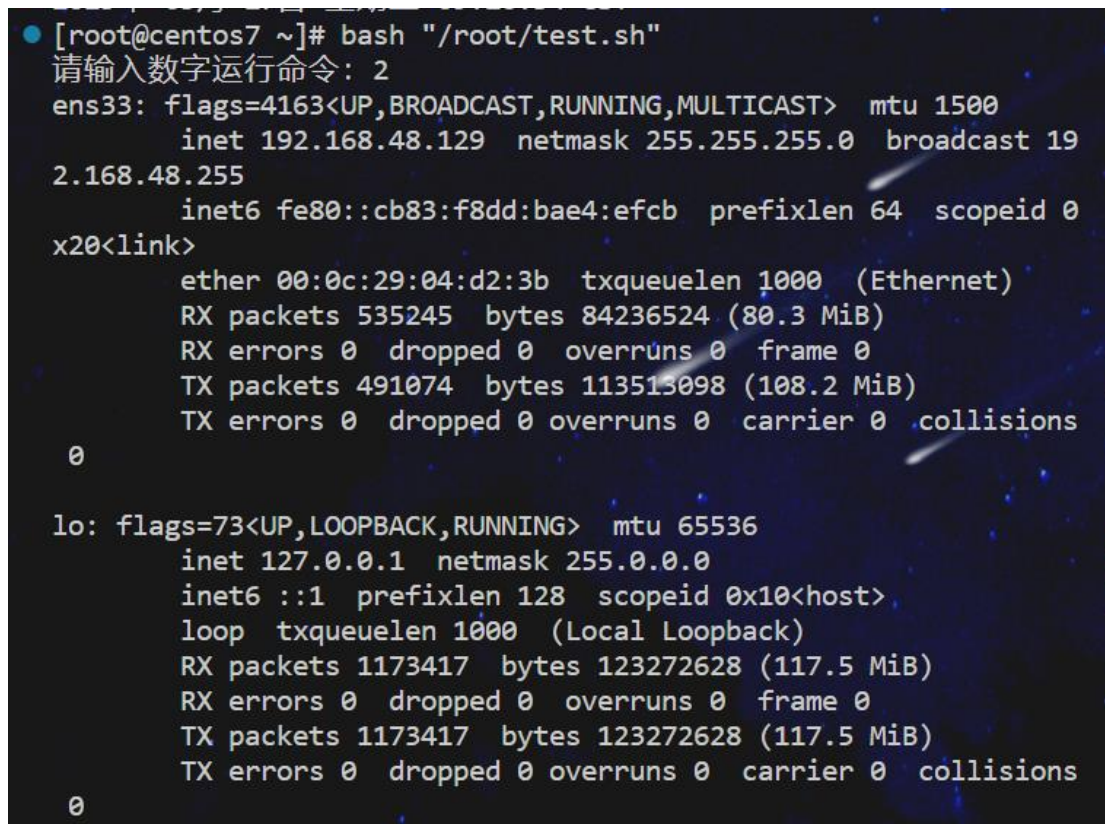
## 二、编程题（共 70 分）

1. 根据输入的数字运行命令：(1:查看当前时间 2:查看网卡信息 3:查看当前用户 4:查看当前目录下的文件)输入其他数字则给出提示控制输入的数字范围。附上所有代码和截图。(10 分)

```
read -p "请输入数字运行命令: " num
if [ $num -eq 1 ]
then date
elif [ $num -eq 2 ]
then ifconfig
elif [ $num -eq 3 ]
then whoami
elif [ $num -eq 4 ]
then ls -l | grep "^-"
else echo "输入的数字范围为 1~4!"
fi
```



```
[root@centos7 ~]# bash "/root/test.sh"
请输入数字运行命令: 1
2025年 05月 27日 星期二 09:26:34 CST
```



```
[root@centos7 ~]# bash "/root/test.sh"
请输入数字运行命令: 2
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.48.129 netmask 255.255.255.0 broadcast 192.168.48.255
    inet6 fe80::cb83:f8dd:bae4:efcb prefixlen 64 scopeid 0
x20<link>
    ether 00:0c:29:04:d2:3b txqueuelen 1000 (Ethernet)
    RX packets 535245 bytes 84236524 (80.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 491074 bytes 113515098 (108.2 MiB)
    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 1173417 bytes 123272628 (117.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1173417 bytes 123272628 (117.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions
    0
```

```

● [root@centos7 ~]# bash "/root/test.sh"
请输入数字运行命令： 3
root

```

```

● [root@centos7 ~]# bash "/root/test.sh"
请输入数字运行命令： 4
-rw-----. 1 root root 1774 5月 16 09:55 anaconda-ks
.cfg
-rw-r--r-- 1 root root 43 5月 22 13:58 hosts
-rw-r--r-- 1 root root 1822 5月 16 09:55 initial-set
up-ks.cfg
-rw-r--r-- 1 root root 66 5月 23 19:18 ip.txt
-rw-r--r-- 1 root root 5002 5月 16 19:34 man_db.conf
-rw----- 1 root root 2762850 5月 22 11:43 messages
-rw-r--r-- 1 root root 126 5月 23 19:34 names.txt
-rw-r--r-- 1 root root 2517 5月 21 11:24 passwd
-rw-r--r-- 1 root root 2304 5月 19 10:47 ptest
-rwxr-xr-x 1 root root 20 5月 21 17:37 test.py
-rwxr-xr-x 1 root root 229 5月 27 09:25 test.sh
-rw-r--r-- 1 root root 435 5月 23 18:42 wangka.txt

```

```

● [root@centos7 ~]# bash "/root/test.sh"
请输入数字运行命令： 5
输入的数字范围为1~4!

```

2. 编写一个脚本，读取学生分数，结果输出学生的等级的个数（A:90+, B:80+, C: 70+, D: 60+, F: 60 以下）。附上所有代码和截图 (10 分)

```

● [root@centos7 shell]# ./kaoshi.sh
A等:3人
B等:4人
C等:3人
D等:3人
F等:3人

```

```
s='awk -v FS="," '{print $NF}' /root/students.txt | sed '/,/d'
```

```
for i in $s
```

```
do
```

```
if [ $i -ge 90 ]
```

```
then A=$((A+1))
```

```
elif [ $i -ge 80 ]
```

```
then B=$((B+1))
```

```
elif [ $i -ge 70 ]
```

```
then C=$((C+1))
```

```
elif [ $i -ge 60 ]
```

```
then D=$((D+1))
```

```
else F=$((F+1))
```

```
fi
```

```

done
echo "A 等:$A 人"
echo "B 等:$B 人"
echo "C 等:$C 人"
echo "D 等:$D 人"
echo "F 等:$F 人"

```



```

[root@centos7 ~]# bash "/root/kaoshi.sh"
A等:3人
B等:4人
C等:3人
D等:4人
F等:2人
[root@centos7 ~]#

```

3. 编写一个函数，传入一个参数，参数为一个字符串变量 `str`，其内容为“关羽，张飞，大乔，刘备，谷爱凌，曹操，夏侯淳，小乔，李典，周瑜，孙坚”，逐行显示出来每个名字，并且在显示大乔和小乔的时候追加是靓女，显示谷爱凌的时候追加是冠军，显示其他姓名时显示是靓仔，最后再打印输出靓女、靓仔、冠军的数量。

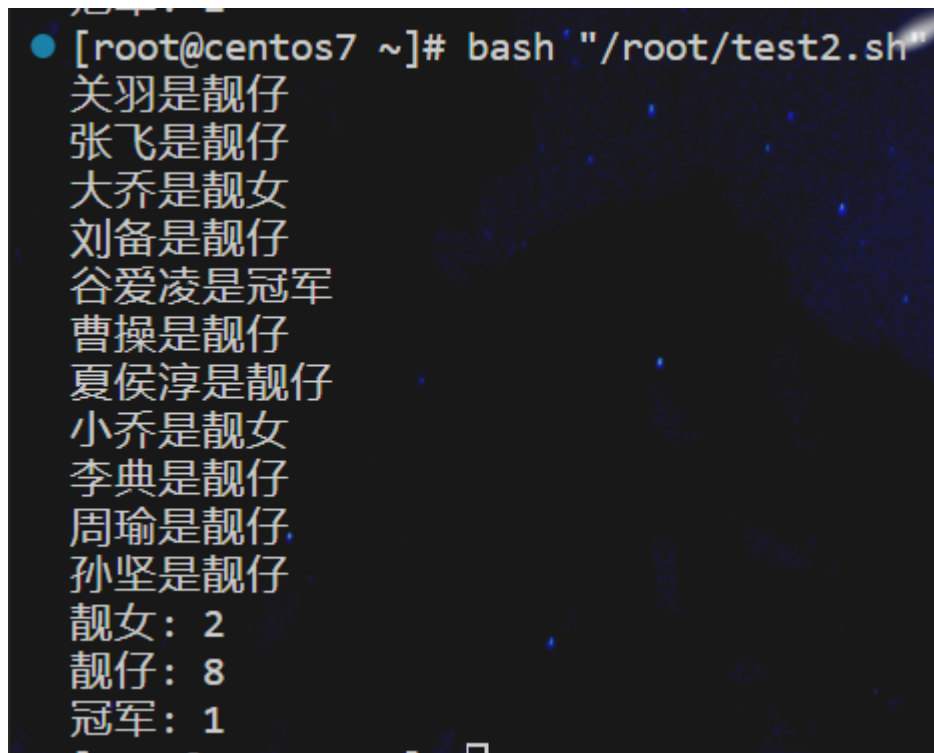
附上所有代码，并将运行结果截图。(25 分)

```

func(){
    IFS=' '
    for i in $1
    do
        if [ $i == "大乔" -o $i == "小乔" ]
        then
            echo "$i 是靓女"
            girl=$((girl+1))
        elif [ $i == "谷爱凌" ]
        then
            echo "$i 是冠军"
            top=$((top+1))
        else
            echo "$i 是靓仔"
            boy=$((boy+1))
        fi
    done
    echo "靓女: $girl"
    echo "靓仔: $boy"
    echo "冠军: $top"
}

```

```
str="关羽，张飞，大乔，刘备，谷爱凌，曹操，夏侯淳，小乔，李典，周瑜，孙坚"  
func "$str"
```



```
[root@centos7 ~]# bash "/root/test2.sh"  
关羽是靓仔  
张飞是靓仔  
大乔是靓女  
刘备是靓仔  
谷爱凌是冠军  
曹操是靓仔  
夏侯淳是靓仔  
小乔是靓女  
李典是靓仔  
周瑜是靓仔  
孙坚是靓仔  
靓女: 2  
靓仔: 8  
冠军: 1
```

4. 编写 shell 脚本，将数据文件 movie.txt，清洗成 movie.csv 格式（注意分数需要降序排序，标题里面需要保留空格），把 csv 文件使用 mysqlcsv.sh（需要附上此文件代码）导入到数据库中。

附上所有代码，并将处理结果及导入数据库结果截图。(25 分)

```
sed 's/\[//g' /root/movie.txt | #掐头  
sed 's/}\[//g' | #去尾  
sed 's/}, {\n/g' | #换行  
sed "s/'/null/g" | #置空值  
sed "s/'//g" | #删引号  
#删除元素说明  
sed 's/rank://g' |  
sed 's/cover_url://g' |  
sed 's/is_playable://g' |  
sed 's/id://g' |  
sed 's/title://g' |  
sed 's/url://g' |  
sed 's/release_date://g' |  
sed 's/actor_count://g' |  
sed 's/vote_count://g' |  
sed 's/score://g' |  
sed 's/is_watched://g' |
```

```
sort -t "," -nrk 10 |
#以逗号分割，添加列名，保存到 csv
awk -v FS="," -v OFS="," 'BEGIN{print "id,title,release_date,score"}{print $4,$5,$7,$10}' \
>/root/sh/movie.csv
```

```
sh >  movie.csv >  data
1 id,title,release_date,score
2 35853587, 进击的巨人 最终季 完结篇 前篇, 2023-03-04, 9.5
3 1418019, 大闹天宫, 1961, 9.4
4 1291561, 千与千寻, 2019-06-21, 9.4
5 2131459, 机器人总动员, 2008-06-27, 9.3
6 1307315, 哪吒闹海, 1979-05-19, 9.2
7 1428581, 天书奇谭, 2021-11-05, 9.2
8 1291583, 天空之城, 1992-05-01, 9.2
9 1291560, 龙猫, 2018-12-14, 9.2
10 25662329, 疯狂动物城, 2016-03-04, 9.2
11 20495023, 寻梦环游记, 2017-11-24, 9.1
12
```

```
#设置 mysql 连接参数
mvhost="127.0.0.1" #主机名
mvprot=3306 #端口
mvuser="root" #用户名
mvpass="root123456" #密码
mov="test" #表空间
```

```
#如果 movie 表存在则删除
sql1="drop table if exists movie"
mysql -h$mvhost -P$mvprot -u$mvuser -p$mvpass $mov -e "$sql1"
```

```
#获取第一行（列名）
names='head -n +1 /root/sh/movie.csv | sed 's/,/ /g`
#找出最后一个列名
lastname='echo "$names" | awk '{print $NF}'`
```

```
#如果 movie 表不存在则新建
sql2="create table if not exists movie("
for i in $names
do
    if [ $i == "$lastname" ]
    then
        sql2="$sql2$i varchar(200))"
    else
        sql2="$sql2$i varchar(200),"
    fi
```



done

```
mysql -h$mvhost -P$mvprot -u$mvuser -p$mvpass $mov -e "$sql2"
```

#导入数据

```
sql3="LOAD DATA INFILE '/usr/local/mysql/data/movie.csv' INTO TABLE movie  
CHARACTER SET utf8
```

```
FIELDS TERMINATED BY ','
```

```
LINES TERMINATED BY '\n'
```

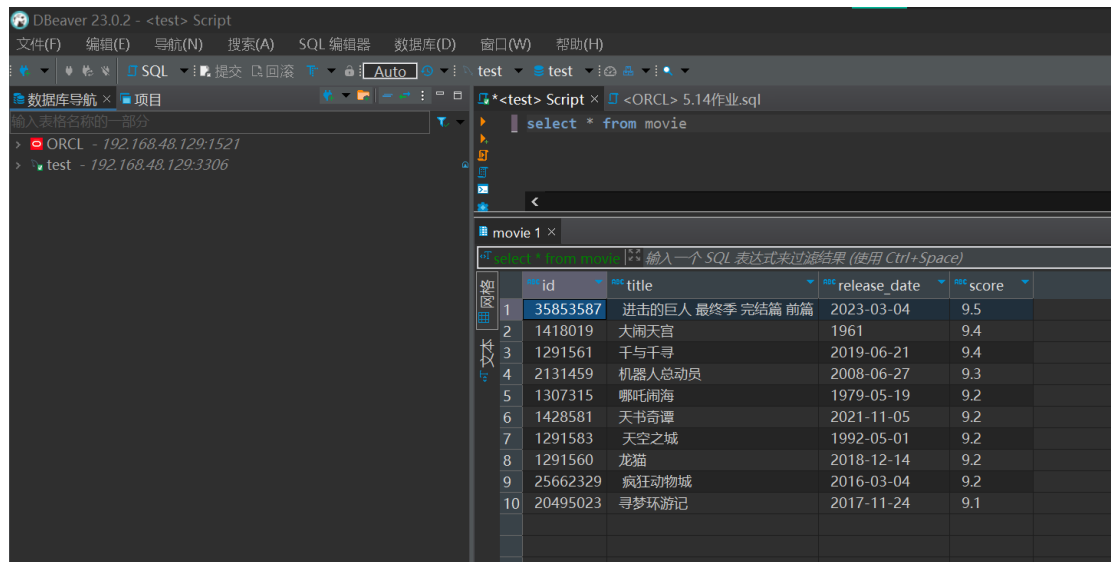
```
IGNORE 1 LINES"
```

```
mysql -h$mvhost -P$mvprot -u$mvuser -p$mvpass $mov -e "$sql3"
```

#查看表和数据

```
sql4="select * from movie"
```

```
mysql -h$mvhost -P$mvprot -u$mvuser -p$mvpass $mov -e "$sql4"
```



The screenshot shows the DBeaver 23.0.2 interface. The top menu bar includes '文件(F)', '编辑(E)', '导航(N)', '搜索(A)', 'SQL 编辑器', '数据库(D)', '窗口(W)', and '帮助(H)'. The main window is titled 'test' and contains a SQL script: 'select \* from movie'. Below the script, the results are displayed in a table with columns: 'id', 'title', 'release\_date', and 'score'. The table contains 10 rows of data. A large, diagonal watermark '大数据培训中心' is visible across the bottom right of the image.

	id	title	release_date	score
1	35853587	进击的巨人 最终季 完结篇 前篇	2023-03-04	9.5
2	1418019	大闹天宫	1961	9.4
3	1291561	千与千寻	2019-06-21	9.4
4	2131459	机器人总动员	2008-06-27	9.3
5	1307315	哪吒闹海	1979-05-19	9.2
6	1428581	天书奇谭	2021-11-05	9.2
7	1291583	天空之城	1992-05-01	9.2
8	1291560	龙猫	2018-12-14	9.2
9	25662329	疯狂动物城	2016-03-04	9.2
10	20495023	寻梦环游记	2017-11-24	9.1