实验 03：文本信息处理

一、实验目的

1、掌握文本查看的方法；
2、掌握文本编辑的方法；
3、掌握文本信息处理相关命令。

二、实验学时

2 学时

扫码看操作演示

三、实验类型

验证性

四、实验需求

1、硬件

每人配备计算机 1 台。

2、软件

安装 VMware WorkStation Pro 或 Oracle VM VirtualBox 软件，安装 Mobaxterm 软件。

3、网络

本地主机与虚拟机能够访问互联网，虚拟机网络不使用 DHCP 服务。

4、工具

无。

五、实验任务

1、完成文本内容的查看；
2、完成文本内容的编辑；
3、完成文本信息处理的操作；
4、使用文本信息处理工具进行日志分析。

六、实验环境

1、本实验需要 VM 1 台；
2、本实验 VM 配置信息如下表所示；

虚拟机配置	操作系统配置
虚拟机名称：VM-Lab-03-Task-01-172.31.0.131 内存：2GB CPU：1 颗，1 核心 虚拟磁盘：20GB 网卡：1 块，NAT	主机名：Lab-03-Task-01 IP 地址：172.31.0.131 子网掩码：255.255.255.0 网关：172.31.0.254 DNS：172.31.0.254

3、本实验拓扑图。

无

4、本实验操作演示视频。

本实验操作演示视频为视频集的第 3 集：

https://www.bilibili.com/video/BV1iH4y1c7ft?p=3

七、实验内容及步骤

1、文本内容查看

（1）短文本查看

通过 cat 命令可查看文本内容。

```
# 查看/etc/目录下的 profile 文件的内容
[root@Lab-03-Task-01 ~]# cat /etc/profile
# ----------------------profile 文件----------------------
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what y
ou
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as t
his
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}
# 此处省略了部分提示信息
# ----------------------profile 文件----------------------
```

```
# 对 profile 文件的内容的非空白行进行编号
[root@Lab-03-Task-01 ~]# cat -b /etc/profile
# ----------profile 文件----------
1  # /etc/profile

2  # System wide environment and startup programs, for login setup
3  # Functions and aliases go in /etc/bashrc

4  # It's NOT a good idea to change this file unless you know what you
5  # are doing. It's much better to create a custom.sh shell script in
6  # /etc/profile.d/ to make custom changes to your environment, as this
7  # will prevent the need for merging in future updates.

8  pathmunge () {
9      case ":${PATH}:" in
10         *:"$1":*)
11             ;;
12         *)
13             if [ "$2" = "after" ] ; then
14                 PATH=$PATH:$1
15             else
16                 PATH=$1:$PATH
17             fi
18     esac
19 }
# 此处省略了部分提示信息
# ----------profile 文件----------

# 对 profile 文件的所有内容行进行编号
[root@Lab-03-Task-01 ~]# cat -n /etc/profile
# ----------profile 文件----------
1  # /etc/profile
2
3  # System wide environment and startup programs, for login setup
4  # Functions and aliases go in /etc/bashrc
5
6  # It's NOT a good idea to change this file unless you know what you
7  # are doing. It's much better to create a custom.sh shell script in
8  # /etc/profile.d/ to make custom changes to your environment, as this
9  # will prevent the need for merging in future updates.
10
```

```
11  pathmunge () {
12      case ":${PATH}:" in
13          *:"$1":*)
14              ;;
15          *)
16              if [ "$2" = "after" ] ; then
17                  PATH=$PATH:$1
18              else
19                  PATH=$1:$PATH
20              fi
21      esac
22  }
# 此处省略了部分提示信息
# --------------------profile 文件--------------------
```

（2）长文本内容查看

通过 more 命令可分页查看较长的文本内容。

```
# 查看/etc/profile 文件内容
[root@Lab-03-Task-01 ~]# more -dc /etc/profile
# --------------------profile 文件--------------------
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what y
ou
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as t
his
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}
# 此处省略部分内容
# --------------------profile 文件--------------------

# 查看/etc/profile 文件内容，每五行显示一次，在显示后再清屏
[root@Lab-03-Task-01 ~]# more -c -5 /etc/profile
# --------------------profile 文件--------------------
```

```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

--更多--(6%)
# --------------------profile 文件--------------------
```

（3）文本头内容查看

通过 head 命令可查看文件的开头内容，默认显示前 10 行。

```
# 查看前 10 行
[root@Lab-03-Task-01 ~]# head /etc/profile
# --------------------profile 文件--------------------
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what y
ou
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as t
his
# will prevent the need for merging in future updates.

# --------------------profile 文件--------------------
# 查看前 2 行,并展示文件名
[root@Lab-03-Task-01 ~]# head -v -n 2 /etc/profile
```

（4）文本末尾内容查看

通过 tail 命令可查看文本的尾部内容，默认显示最后 10 行。

```
# 查看最后 10 行
[root@Lab-03-Task-01 ~]# tail /etc/passwd
# /etc/profile
# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc
# It's NOT a good idea to change this file unless you know what y
ou
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as t
his
# will prevent the need for merging in future updates.
# 查看最后 2 行,并展示文件名
[root@Lab-03-Task-01 ~]# tail -v -n 2 /etc/passwd
# --------------------passwd 文件--------------------
==> /etc/passwd <==
systemd-oom:x:992:992:systemd Userspace OOM Killer:/:/usr/sbin/nol
ogin
centoslab:x:1000:1002::/home/centoslab:/bin/bash
# --------------------passwd 文件--------------------
```

（5）通过 grep 命令可搜索指定文件，并显示匹配的行。

```
# 查看/etc/passwd 包含 user 的文本行的行数
[root@Lab-03-Task-01 ~]# grep -c user /etc/passwd
1

# 查看/etc/passwd 包含 user 的文本行，并显示出行号
[root@Lab-03-Task-01 ~]# grep -n user /etc/passwd
18:chrony:x:997:994:chrony system user:/var/lib/chrony:/sbin/nologi

# 查看/etc/passwd 以 user 开头的文本行
[root@Lab-03-Task-01 ~]# grep '^\user' /etc/passwd
```

2、文本内容编辑

（1）使用 sed 对日志文件进行编辑操作。

```
# 使用 sed 命令在第二行后面追加测试代码
[root@Lab-03-Task-01 ~]# sed '2a #test123456789' /var/log/messag
es
# ----------messages 文件----------
Jul 17 09:25:01 qs-dev-plat systemd[1]: Starting Check pmlogger ins
tances are running...
Jul 17 09:25:01 qs-dev-plat systemd[1]: Started Check pmlogger inst
ances are running.
#test123456789
Jul 17 09:25:03 qs-dev-plat systemd[1]: pmlogger_check.service: Succ
eeded.
# 此处省略了部分提示信息
# ----------messages 文件----------
```

（2）使用 sort 对日志文件进行编辑操作。

```
# 使用 sort 命令对/var/log/messages 排序
[root@Lab-03-Task-01 ~]# sort /var/log/messages
# ----------messages 文件----------
Jul 17 09:10:35 venus journal: Runtime journal is using 8.0M (max a
llowed 90.9M, trying to leave 136.4M free of 901.6M available → cu
rrent limit 90.9M).
Jul 18 15:50:00 venus kernel: Initializing cgroup subsys cpuset
Jul 15 14:19:59 mars journal: Runtime journal is using 8.0M (max all
owed 90.9M, trying to leave 136.4M free of 901.6M available → cur
rent limit 90.9M).
Jul 15 14:19:59 mars kernel: #011RCU restricting CPUs from NR_CP
US=5120 to nr_cpu_ids=128.
# 此处省略了部分提示信息
# ----------messages 文件----------
```

（3）使用 awk 对日志文件进行编辑操作。

```
# 输出包含 "MySQL" 的行
[root@Lab-03-Task-01 ~]# awk '/MySQL/ ' /var/log/messages
# ----------messages 文件----------
Jul 15 14:19:00 Saturn systemd[1]: Starting MySQL Server...
Jul 15 14:19:09 Saturn systemd[1]: Started MySQL Server.
```

```
Jul 15 14:19:11 Saturn systemd[1]: Stopping MySQL Server...
Jul 15 14:19:15 Saturn systemd[1]: Stopped MySQL Server.
Jul 15 14:19:22 Saturn systemd[1]: Starting MySQL Server...
Jul 15 14:19:25 Saturn systemd[1]: Started MySQL Server.
Jul 15 14:19:30 Saturn dnf[17828]: MySQL 8.0 Community Server
                    5.9 kB/s | 2.6 kB     00:00
Jul 15 14:19:59 Saturn dnf[17828]: MySQL Connectors Community
                     5.4 kB/s | 2.6 kB     00:00
Jul 15 14:19:59 Saturn dnf[17828]: MySQL Tools Community
                    4.8 kB/s | 2.6 kB     00:00
# --------------------messages 文件--------------------

#从文件中找出长度大于 360 的行
[root@Lab-03-Task-01 ~]# awk 'length>360' /var/log/messages
# --------------------messages 文件--------------------
Jul 15 14:07:16 Saturn systemd: systemd 252-15.el9 running in syste
m mode (+PAM +AUDIT +SELINUX -APPARMOR +IMA +SMACK +S
ECCOMP +GCRYPT +GNUTLS +OPENSSL +ACL +BLKID +CURL +ELF
UTILS -FIDO2 +IDN2 -IDN -IPTC +KMOD +LIBCRYPTSETUP +LIBFDIS
K +PCRE2 -PWQUALITY +P11KIT -QRENCODE +TPM2 +BZIP2 +LZ4
+XZ +ZLIB +ZSTD -BPF_FRAMEWORK +XKBCOMMON +UTMP +SYS
VINIT default-hierarchy=unified)
Jul 15 14:19:22 Saturn systemd: systemd 252-15.el9 running in syste
m mode (+PAM +AUDIT +SELINUX -APPARMOR +IMA +SMACK +S
ECCOMP +GCRYPT +GNUTLS +OPENSSL +ACL +BLKID +CURL +ELF
UTILS -FIDO2 +IDN2 -IDN -IPTC +KMOD +LIBCRYPTSETUP +LIBFDIS
K +PCRE2 -PWQUALITY +P11KIT -QRENCODE +TPM2 +BZIP2 +LZ4
+XZ +ZLIB +ZSTD -BPF_FRAMEWORK +XKBCOMMON +UTMP +SYS
VINIT default-hierarchy=unified)
Jul 15 14:23:59 Saturn containerd[798]: time="2023-08-12T12:08:37.7
67774118+08:00" level=info msg="skip loading plugin \"io.container
d.snapshotter.v1.aufs\"..." error="aufs is not supported (modprobe a
ufs failed: exit status 1 \"modprobe: FATAL: Module aufs not found
in directory /lib/modules/5.14.0-333.el9.x86_64\\n\"): skip plugin" typ
e=io.containerd.snapshotter.v1
# --------------------messages 文件--------------------
```

（4）使用 uniq 对日志文件进行编辑操作。

```
# 使用 uniq 删除重复行
[root@Lab-03-Task-01 ~]# uniq /var/log/messages
# --------------------messages 文件--------------------
Jul 15 14:19:59 mars journal: Runtime journal is using 8.0M (max all
owed 90.9M, trying to leave 136.4M free of 901.6M available → cur
rent limit 90.9M).
Jul 15 14:19:59 mars kernel: Initializing cgroup subsys cpuset
Jul 15 14:19:59 mars kernel: Initializing cgroup subsys cpu
Jul 15 14:19:59 mars kernel: Initializing cgroup subsys cpuacct
# --------------------messages 文件--------------------

# 综合应用
```

```
[root@Lab-03-Task-01 ~]# sed 's/\[.*$//' /var/log/messages | sed 's/.\{35\}//' | sort | uniq -c
# --------------------messages 文件--------------------
 sort | uniq -c
      6 -01 auditd
     86 -01 augenrules
      3 -01 bootctl
    119 -01 chronyd
      3 -01 dbus-broker-lau
    140 -01 dracut
      6 -01 dracut-cmdline
     12 -01 dracut-initqueue
      8 -01 kdumpctl
      9 -01 kernel:
      3 -01 kernel: #011Rude variant of Tasks RCU enabled.
      3 -01 kernel: #011Tracing variant of Tasks RCU enabled
# 此处省略了部分提示信息
# --------------------messages 文件--------------------
```

3、文本信息处理

（1）使用 vi 编辑/var/log/message 日志内容。

```
# 在/var/log/messages 文件中写入“文本信息处理”的测试文字
[root@Lab-03-Task-01 ~]# vi /var/log/messages
# --------------------messages 文件--------------------
文本信息处理
Jul  7 10:04:43 Project-Number-Task-01 kernel: Linux version 5.14.0-333.el9.x86_
64 (mockbuild@x86-05.stream.rdu2.redhat.com) (gcc (GCC) 11.4.1 20230605 (Red Hat
 11.4.1-2), GNU ld version 2.35.2-42.el9) #1 SMP PREEMPT_DYNAMIC Wed Jun 28 09:4
7:27 UTC 2023
Jul  7 10:04:43 Project-Number-Task-01 kernel: The list of certified hardware an
d cloud instances for Red Hat Enterprise Linux 9 can be viewed at the Red Hat Ec
osystem Catalog, https://catalog.redhat.com.
Jul  7 10:04:43 Project-Number-Task-01 kernel: Command line: BOOT_IMAGE=(hd0,msd
os1)/vmlinuz-5.14.0-333.el9.x86_64 root=/dev/mapper/cs_miwifi--r3600--srv-root r
o crashkernel=1G-4G:192M,4G-64G:256M,64G-:512M resume=/dev/mapper/cs_miwifi--r36
00--srv-swap rd.lvm.lv=cs_miwifi-r3600-srv/root rd.lvm.lv=cs_miwifi-r3600-srv/sw
ap
# 此处省略了部分提示信息
# --------------------messages 文件--------------------
```

（2）使用 nano 编辑/var/log/message 日志内容。

```
# 下载 nano
yum install -y nano
# 在/var/log/messages 文件中写入“nano 文本信息处理”的测试文字
[root@Lab-03-Task-01 ~]# nano /var/log/messages
# ----------messages 文件----------
nano 文本信息处理
Jul 7 10:04:43 Project-Number-Task-01 kernel: Linux version 5.14.0-333.el9.x86_
64 (mockbuild@x86-05.stream.rdu2.redhat.com) (gcc (GCC) 11.4.1 20230605 (Red Hat
 11.4.1-2), GNU ld version 2.35.2-42.el9) #1 SMP PREEMPT_DYNAMIC Wed Jun 28 09:4
7:27 UTC 2023
Jul 7 10:04:43 Project-Number-Task-01 kernel: The list of certified hardware an
d cloud instances for Red Hat Enterprise Linux 9 can be viewed at the Red Hat Ec
osystem Catalog, https://catalog.redhat.com.
Jul 7 10:04:43 Project-Number-Task-01 kernel: Command line: BOOT_IMAGE=(hd0,msd
os1)/vmlinuz-5.14.0-333.el9.x86_64 root=/dev/mapper/cs_miwifi--r3600--srv-root r
o crashkernel=1G-4G:192M,4G-64G:256M,64G-:512M resume=/dev/mapper/cs_miwifi--r36
00--srv-swap rd.lvm.lv=cs_miwifi-r3600-srv/root rd.lvm.lv=cs_miwifi-r3600-srv/sw
ap
# 此处省略了部分提示信息
# ----------messages 文件----------
# 使用键盘快捷键 ctrl+O 写入
# 使用键盘快捷键 ctrl+x 离开
```

4、使用文本信息处理工具进行日志分析

（1）linux 系统内核和系统日志文件一般由 rsyslog 软件包提供，目录位置 ：/etc/rsyslog.conf。结合已学命令进行了解系统中关于日志文件的设置。

```
[root@Lab-03-Task-01 ~]# grep -v “^$” /etc/rsyslog.conf
# ----------rsyslog.conf----------
# rsyslog configuration file

# For more information see /usr/share/doc/rsyslog-*/rsyslog_conf.html
# or latest version online at http://www.rsyslog.com/doc/rsyslog_conf.html
# If you experience problems, see http://www.rsyslog.com/doc/troubleshoot.html

#### MODULES ####

module(load="imuxsock"    # provides support for local system log
```

```
ging (e.g. via logger command)
        SysSock.Use="off") # Turn off message reception via local lo
g socket;
                                # local messages are retrieved through im
journal now.
module(load="imjournal"                 # provides access to the syste
md journal
        StateFile="imjournal.state") # File to store the position in the
 journal
#module(load="imklog") # reads kernel messages (the same are rea
d from journald)
#module(load="immark") # provides --MARK-- message capability
# 此处省略部分内容
# --------------------rsyslog.conf--------------------

# 查看前 10 行
[root@Lab-03-Task-01 ~]# head /etc/rsyslog.conf
# --------------------rsyslog.conf--------------------
# rsyslog configuration file

# For more information see /usr/share/doc/rsyslog-*/rsyslog_conf.ht
ml
# or latest version online at http://www.rsyslog.com/doc/rsyslog_con
f.html
# If you experience problems, see http://www.rsyslog.com/doc/troub
leshoot.html

#### MODULES ####

module(load="imuxsock"    # provides support for local system log
ging (e.g. via logger command)
        SysSock.Use="off") # Turn off message reception via local lo
g socket;
# --------------------rsyslog.conf--------------------

# 查看最后 10 行
[root@Lab-03-Task-01 ~]# tail /etc/rsyslog.conf
# --------------------rsyslog.conf--------------------
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up agai
n.
#queue.filename="fwdRule1"         # unique name prefix for spool f
iles
#queue.maxdiskspace="1g"           # 1gb space limit (use as much
as possible)
#queue.saveonshutdown="on"         # save messages to disk on sh
utdown
#queue.type="LinkedList"           # run asynchronously
#action.resumeRetryCount="-1"     # infinite retries if host is down
```

```
# Remote Logging (we use TCP for reliable delivery)
# remote_host is: name/ip, e.g. 192.168.0.1, port optional e.g. 10514
#Target="remote_host" Port="XXX" Protocol="tcp")
# ----------rsyslog.conf----------
# 使用 vi 进行编辑/etc/rsyslog.conf
[root@Lab-03-Task-01 ~]# vi /etc/rsyslog.conf
```

（2）查看历史操作命令

使用 history 命令查看历史操作命令。

```
[root@Lab-03-Task-01 ~]# history 20 #显示最近 20 个命令记录
# ----------history----------
   62  hostnamectl set-name Lab-02-Task-01
   63  hostnamectl set-hostname Lab-02-Task-01
   64  reboot
   65  groupadd labs
   66  useradd centoslab
   67  passwd centoslab
   68  cat /etc/profile
   69  cat -b /etc/profile
   70  cat -n /etc/profile
   71  head /etc/profile
   72   tail -v -n 2 /etc/passwd
   73  history
   74  grep -c user /etc/passwd
   75   grep -n user /etc/passwd
   76  grep '^\user' /etc/passwd
   77  more /var/log/messages
   78  vi /var/log/message
   79  vi /var/log/messages
   80  sed 's/\[.*$//' /var/log/messages | sed 's/.\{35\}//' | sort | uni
q -c
   81  history 20
# ----------history----------
# 将当前历史命令缓冲区命令写入历史命令文件中
[root@Lab-03-Task-01 ~]# history -w
```

八、实验考核

实验考核分为【实验随堂查】和【实验线上考】两个部分。

实验随堂查：每个实验设置 2-5 考核点。完成实验任务后，任课教师随机选择一个考核点，学生现场进行演示和汇报讲解。

实验线上考：每个实验设置 10 道客观题。通过线上考核平台（如课堂派）进行作答。

1、实验随堂查

本实验随堂查设置 2 个考核点，具体如下：

考核点 1：创建文本文件，至少用三种方法将内容（自定义）写入文件

考核点 2：只显示 vi /var/log/messages 文件中关于 Linux 的内容

2、实验线上考

本实验线上考共 10 题，题型为单选、多选、判断、填空等题型。