[toc]

课上测试

作业题目：Linux命令实践

学习Linux核心命令

- ls
- who
- pwd
- cd
- man
- whereis
- which
- find
- locate
- grep

作业要求（9’）

- 参考课件，针对上面核心命令重复老师的演示内容（3’）
- 选择自己习惯的 AI 工具， 针对上面核心命令，让 AI 给自己生成至少例子，然后自己进行验证，总结（3’）
- 让 AI工具推荐一些重要命令，然后选择自己认为会高频应用的至少 3个命令（重点推荐 echo，od，sort，bc，ls）进行自主深入学习，每个命令不少于 5 个例子，提交验证，实践过程和总结（3’）

作业提交要求 (1')

0. 记录实践过程和 AI 问答过程，尽量不要截图，给出文本内容
1. github链接
2. (必做)提交作业 markdown文档，命名为”学号-姓名-作业题目.md”
3. (必做)提交作业 markdown文档转成的 PDF 文件，命名为”学号-姓名-作业题目.pdf”

针对上面核心命令重复老师的演示内容：

- `ls`实践内容

  ```
  root@Youer:~# ls
  Desktop    GmSSL     Templates  client    main    snap    testmainso
  Downloads  Pictures  cheat      homework  server  socket
  root@Youer:~# ls .
  Desktop    GmSSL     Templates  client    main    snap    testmainso
  Downloads  Pictures  cheat      homework  server  socket
  root@Youer:~# ls -a
  .              .bashrc  .dotnet      .motd_shown      .vscode-server
  Pictures   homework  socket
  ```

```
..             .cache    .head.h.swo  .profile         Desktop
Templates  main      testmainso
.Xauthority    .cgdb    .head.h.swp  .python_history  Downloads       cheat
server
.bash_history  .ddd     .lesshst     .viminfo         GmSSL           client
snap
root@Youer:~# ls -l
total 92
lrwxrwxrwx  1 root root    13 Sep  2  2022 Desktop -> /root/Desktop
lrwxrwxrwx  1 root root    15 Sep  2  2022 Downloads -> /root/Downloads
drwxr-xr-x 10 root root  4096 Sep  8 14:45 GmSSL
lrwxrwxrwx  1 root root    14 Sep  2  2022 Pictures -> /root/Pictures
lrwxrwxrwx  1 root root    15 Sep  2  2022 Templates -> /root/Templates
drwxr-xr-x  3 root root  4096 Sep 29  2022 cheat
-rwxr-xr-x  1 root root 16472 Dec 14  2022 client
drwxr-xr-x  2 root root  4096 Sep  3  2022 homework
-rwxr-xr-x  1 root root 16048 Dec  4  2022 main
-rwxr-xr-x  1 root root 16464 Dec 14  2022 server
drwxr-xr-x  4 root root  4096 Sep  2  2022 snap
drwxr-xr-x  2 root root  4096 Sep  2  2022 socket
-rwxr-xr-x  1 root root 16160 Dec  4  2022 testmainso
```

- `who`实践内容

```
root@Youer:~# who
root@Youer:~# whoami
root
root@Youer:~# who -a
```

- `pwd`实践内容

```
root@Youer:~# pwd
/root
root@Youer:~# cd GmSSL
root@Youer:~/GmSSL# pwd
/root/GmSSL
```

- `cd`实践内容

```
root@Youer:~/GmSSL# cd /root/GmSSL
root@Youer:~/GmSSL# cd .
root@Youer:~/GmSSL# cd ..
root@Youer:~# cd -
/root/GmSSL
root@Youer:~/GmSSL# cd ~
root@Youer:~# cd -
```

```
/root/GmSSL
root@Youer:~/GmSSL# cd ..
```

- `man`实践内容

```
root@Youer:~# man man
root@Youer:~# man printf
root@Youer:~# man 3 printf
root@Youer:~# man -k sort
alphasort (3)        - scan a directory for matching entries
apt-sortpkgs (1)     - Utility to sort package index files
bsearch (3)          - binary search of a sorted array
bunzip2 (1)          - a block-sorting file compressor, v1.0.8
bzip2 (1)            - a block-sorting file compressor, v1.0.8
comm (1)             - compare two sorted files line by line
qsort (3)            - sort an array
qsort_r (3)          - sort an array
sort (1)             - sort lines of text files
Sort::Naturally (3pm) - - sort lexically, but sort numeral parts numerically
tsort (1)            - perform topological sort
versionsort (3)      - scan a directory for matching entries
XConsortium (7)      - X Consortium information
root@Youer:~# man -k sort |grep 3
alphasort (3)        - scan a directory for matching entries
bsearch (3)          - binary search of a sorted array
qsort (3)            - sort an array
qsort_r (3)          - sort an array
Sort::Naturally (3pm) - - sort lexically, but sort numeral parts numerically
versionsort (3)      - scan a directory for matching entries
root@Youer:~# man -f qsort
qsort (3)            - sort an array
```

- `whereis`实践内容

```
root@Youer:~# whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
```

- `which`实践内容

```
root@Youer:~#  which ls
/usr/bin/ls
```

- `find`实践内容

```
root@Youer:~# find main
main
```

- `locate`实践内容

```
root@Youer:~# locate GmSSL
/mnt/c/Users/xlm20/Downloads/GmSSL-master.zip
/root/GmSSL
/root/GmSSL/.git
/root/GmSSL/.github
/root/GmSSL/.gitignore
……
```

- `grep`实践内容

```
root@Youer:~# grep -nr O_RDWR /usr/include
/usr/include/asm-generic/fcntl.h:22:#define O_RDWR              00000002
/usr/include/linux/dma-heap.h:18:/* Valid FD_FLAGS are O_CLOEXEC, O_RDONLY,
O_WRONLY, O_RDWR */
/usr/include/scsi/cxlflash_ioctl.h:65: * and O_RDWR flags defined in the
fcntl.h header file.
/usr/include/drm/drm.h:839:#define DRM_RDWR O_RDWR
/usr/include/x86_64-linux-gnu/bits/fcntl-linux.h:45:#define O_RDWR
02
```

使用Kimi针对上面核心命令，让AI生成至少例子，然后进行验证，总结：

- 与AI的对话：
    - 对话链接
    - 对话内容如下: User:

```
学习Linux核心命令
-  ls
-  who
-  pwd
-  cd
-  man
-  whereis
-  which
-  find
-  locate
-  grep
针对以上命令，请为我生成一些例子
```

Kimi: 当然，以下是针对你列出的Linux核心命令的一些示例：

1. `ls`：列出目录内容

   ```
   ls             # 列出当前目录下的文件和文件夹
   ls -l          # 以长格式列出详细信息
   ls -a          # 显示所有文件，包括隐藏文件
   ls /path/to/dir  # 列出指定目录下的文件和文件夹
   ```

2. `who`：显示当前登录用户的信息

   ```
   who             # 显示当前登录的用户列表
   who -a          # 显示所有登录用户的信息
   who -m          # 显示最后登录的用户信息
   ```

3. `pwd`：显示当前工作目录的路径

   ```
   pwd             # 显示当前目录的完整路径
   ```

4. `cd`：改变当前目录

   ```
   cd              # 切换到用户的主目录
   cd /path/to/dir  # 切换到指定的目录
   cd ..           # 切换到当前目录的父目录
   ```

5. `man`：查看命令的手册页

   ```
   man ls          # 查看 `ls` 命令的手册页
   man 5 passwd    # 查看与密码文件相关的手册页（第5节）
   ```

6. `whereis`：定位二进制、源和手册页文件的位置

   ```
   whereis java    # 查找 `java` 命令的位置
   whereis ssh     # 查找 `ssh` 相关的文件
   ```

7. `which`：在系统路径中查找执行文件

   ```
   which python    # 查找 `python` 命令的路径
   which vim      # 查找 `vim` 编辑器的路径
   ```

8. `find`：在文件系统中搜索文件

    ```
    find / -name "filename"  # 在整个文件系统中搜索名为 "filename" 的文件
    find /home -type d -name "*logs*"  # 在 /home 目录下搜索所有名为 "*logs*" 的目录
    ```

9. `locate`：使用数据库快速查找文件

    ```
    locate filename  # 查找名为 "filename" 的文件
    locate "*.txt"   # 查找所有扩展名为 ".txt" 的文件
    ```

10. `grep`：搜索文件内容

~~~~
 ```bash
 grep "pattern" filename  # 在 "filename" 文件中搜索 "pattern"
 grep -r "pattern" /path/to/dir  # 在指定目录及其子目录中递归搜索
"pattern"
 ```
~~~~

请注意，这些命令可能需要根据你的具体需求进行调整。例如，你可能需要指定正确的文件名或路径。此外，一些命令（如 `man` 和 `grep`）有大量的选项，可以进一步定制它们的输出和行为。

- 我的验证：

1. `ls`:

```
root@Youer:~# ls
Desktop    GmSSL     Templates  client    main    snap    testmainso
Downloads  Pictures  cheat      homework  server  socket
root@Youer:~# ls -l
total 92
lrwxrwxrwx  1 root root    13 Sep  2  2022 Desktop -> /root/Desktop
lrwxrwxrwx  1 root root    15 Sep  2  2022 Downloads -> /root/Downloads
drwxr-xr-x 10 root root  4096 Sep  8 14:45 GmSSL
lrwxrwxrwx  1 root root    14 Sep  2  2022 Pictures -> /root/Pictures
lrwxrwxrwx  1 root root    15 Sep  2  2022 Templates -> /root/Templates
drwxr-xr-x  3 root root  4096 Sep 29  2022 cheat
-rwxr-xr-x  1 root root 16472 Dec 14  2022 client
drwxr-xr-x  2 root root  4096 Sep  3  2022 homework
-rwxr-xr-x  1 root root 16048 Dec  4  2022 main
-rwxr-xr-x  1 root root 16464 Dec 14  2022 server
drwxr-xr-x  4 root root  4096 Sep  2  2022 snap
drwxr-xr-x  2 root root  4096 Sep  2  2022 socket
-rwxr-xr-x  1 root root 16160 Dec  4  2022 testmainso
root@Youer:~# ls -a
```

```
.              .bashrc  .dotnet      .motd_shown      .vscode-server
Pictures   homework  socket
..             .cache   .head.h.swo  .profile         Desktop
Templates  main      testmainso
.Xauthority    .cgdb    .head.h.swp  .python_history  Downloads
cheat      server
.bash_history  .ddd     .lesshst     .viminfo         GmSSL
client     snap
root@Youer:~# ls /path/to/dir
ls: cannot access '/path/to/dir': No such file or directory
```

2. `who`:

```
root@Youer:~# who
root@Youer:~# who -a
root@Youer:~# who -m
```

3. `pwd`:

```
root@Youer:~# cd GmSSL
root@Youer:~/GmSSL# cd build
root@Youer:~/GmSSL/build# pwd
/root/GmSSL/build
```

4. `cd`:

```
root@Youer:~/GmSSL/build# cd ..
root@Youer:~/GmSSL# cd
root@Youer:~# cd /root/GmSSL/build
root@Youer:~/GmSSL/build#
```

5. `man`:
    1. man ls:（省略了部分内容）

    ```
    LS(1)                                        User Commands
    LS(1)

    NAME
          ls - list directory contents

    SYNOPSIS
          ls [OPTION]... [FILE]...

    DESCRIPTION
          List  information  about  the FILEs (the current directory
    ```

```
by default).  Sort entries alphabeti-
      cally if none of -cftuvSUX nor --sort is specified.

      Mandatory arguments to long options are mandatory for short
options too.

      -a, --all
            do not ignore entries starting with .

      -A, --almost-all
            do not list implied . and ..

      --author
            with -l, print the author of each file

      -b, --escape
            print C-style escapes for nongraphic characters

      --block-size=SIZE
Manual page ls(1) line 1 (press h for help or q to quit)
```

2. man 5 passwd:（省略了部分内容）

```
PASSWD(5)                             File Formats and Conversions
PASSWD(5)

NAME
      passwd - the password file

DESCRIPTION
      /etc/passwd contains one line for each user account, with
seven fields delimited by colons
      (“:”). These fields are:

      •   login name

      •   optional encrypted password

      •   numerical user ID

      •   numerical group ID

      •   user name or comment field

      •   user home directory

      •   optional user command interpreter

      If the password field is a lower-case “x”, then the
encrypted password is actually stored in the
      shadow(5) file instead; there must be a corresponding line
in the /etc/shadow file, or else the
```

```
      user account is invalid.

Manual page passwd(5) line 1 (press h for help or q to quit)
```

6. `whereis`:

```
root@Youer:~# whereis java
java: /usr/share/java
root@Youer:~# whereis ssh
ssh: /usr/bin/ssh /etc/ssh /usr/share/man/man1/ssh.1.gz
```

7. `which`：

```
root@Youer:~# which python
root@Youer:~# which vim
/usr/bin/vim
root@Youer:~# which python3
/usr/bin/python3
```

8. `find`:

```
root@Youer:~# find / -name "testmainso"
/root/testmainso
find: File system loop detected; ‘/mnt/wslg/distro’ is part of the same file system loop as ‘/’.// 表示 /mnt/wslg/distro 指向了一个循环挂载点，这通常发生在挂载配置错误时。
find: ‘/mnt/d/$RECYCLE.BIN/S-1-5-18’: Permission denied // 表明没有足够的权限来访问某些目录
find: ‘/mnt/d/$RECYCLE.BIN/S-1-5-21-2139585154-949242277-2800085354-500’: Permission denied
root@Youer:~# find /home -type d -name "*logs*"
root@Youer:~#
```

9. `locate`:

```
root@Youer:~# locate testmainso
/root/testmainso
root@Youer:~# locate ".c"
/etc/adduser.conf
/etc/apg.conf
/etc/appstream.conf
/etc/brltty.conf
/etc/ca-certificates.conf
/etc/ca-certificates.conf.dpkg-old
```

```
/etc/debconf.conf
/etc/deluser.conf
```

10. grep:

~~~
```
root@Youer:~# grep "argc" /root/GmSSL/tools/sm4_ctr_sm3_hmac.c
int sm4_ctr_sm3_hmac_main(int argc, char **argv)
      argc--;
      if (argc < 1) {
      while (argc > 0) {
                        if (--argc < 1) goto bad;
                        if (--argc < 1) goto bad;
                        if (--argc < 1) goto bad;
                        if (--argc < 1) goto bad;
                        if (--argc < 1) goto bad;
                        if (--argc < 1) goto bad;
               argc--;
root@Youer:~# grep -r "argc" /root/GmSSL/tools
/root/GmSSL/tools/reqparse.c:int reqparse_main(int argc, char **argv)
/root/GmSSL/tools/reqparse.c:   argc--;
/root/GmSSL/tools/reqparse.c:   while (argc > 0) {
/root/GmSSL/tools/reqparse.c:                   if (--argc < 1) goto bad;
/root/GmSSL/tools/reqparse.c:                   if (--argc < 1) goto bad;
/root/GmSSL/tools/reqparse.c:           argc--;
/root/GmSSL/tools/sm4_cbc.c:int sm4_cbc_main(int argc, char **argv)
/root/GmSSL/tools/sm4_cbc.c:    argc--;
/root/GmSSL/tools/sm4_cbc.c:    if (argc < 1) {
/root/GmSSL/tools/sm4_cbc.c:    while (argc > 0) {
/root/GmSSL/tools/sm4_cbc.c:                    if (--argc < 1) goto bad;
/root/GmSSL/tools/sm4_cbc.c:                    if (--argc < 1) goto bad;
/root/GmSSL/tools/sm4_cbc.c:                    if (--argc < 1) goto bad;
/root/GmSSL/tools/sm4_cbc.c:                    if (--argc < 1) goto bad;
/root/GmSSL/tools/sm4_cbc.c:            argc--;
/root/GmSSL/tools/sm2decrypt.c:int sm2decrypt_main(int argc, char **argv)
/root/GmSSL/tools/sm2decrypt.c: argc--;
/root/GmSSL/tools/sm2decrypt.c: if (argc < 1) {
/root/GmSSL/tools/sm2decrypt.c: while (argc > 0) {
/root/GmSSL/tools/sm2decrypt.c:                 if (--argc < 1) goto bad;
/root/GmSSL/tools/sm2decrypt.c:                 if (--argc < 1) goto bad;
/root/GmSSL/tools/sm2decrypt.c:                 if (--argc < 1) goto bad;
/root/GmSSL/tools/sm2decrypt.c:                 if (--argc < 1) goto bad;
/root/GmSSL/tools/sm2decrypt.c:         argc--;
```
~~~

- 总结：man命令是自学命令的神器，可以查看命令的文档。who命令显示当前登录用户信息。ls、pwd、cd命令都与目录有关，ls用于列出目录内容，pwd用于打印当前工作目录路径，cd则用于改变当前目录。whereis、which、find、locate 和 grep 都是 Linux 系统中用于查找的工具，其中whereis用于搜索二进

制、源和手册页文件的位置，比如whereis java；which用于搜索可执行文件的路径，比如which python；find在文件系统中递归搜索文件和目录，而locate使用数据库快速搜索文件路径，而grep用于搜索文件中的文本内容。

让Kimi推荐一些重要命令，然后选择会高频应用的3个命令echo、sort、sed进行自主深入学习，每个命令不少于 5 个例子，提交验证，实践过程和总结

- AI的推荐
    - 对话链接
    - 对话内容 User: 请为我推荐Linux的高频应用的一些重要命令并给出至少五个示例

      Kimi: Linux 系统以其强大的命令行界面而闻名，许多高频应用都可以通过命令行高效地完成。以下是一些重要的命令及其示例：

      1. `grep`：搜索文件内容
          - 查找包含特定文本的行：

            ```
            grep "text" filename
            ```

          - 在所有 `.txt` 文件中递归搜索包含 "text" 的行：

            ```
            grep -r "text" /path/to/directory/*.txt
            ```

      2. `cp`：复制文件或目录
          - 复制单个文件：

            ```
            cp source.txt destination.txt
            ```

          - 复制整个目录（递归复制）：

            ```
            cp -r source_directory destination_directory
            ```

      3. `mv`：移动或重命名文件/目录
          - 重命名文件：

            ```
            mv oldname.txt newname.txt
            ```

          - 移动文件到另一个目录：

```
mv filename.txt /path/to/directory/
```

4. `chmod`：更改文件或目录的权限

    - 给文件所有者添加写权限：

```
chmod u+w filename
```

    - 将目录设置为只有所有者可读写执行：

```
chmod 700 directoryname
```

5. `tar`：打包和压缩文件

    - 创建一个压缩包：

```
tar -czvf archive.tar.gz /path/to/directory
```

    - 解压一个压缩包：

```
tar -xzvf archive.tar.gz
```

6. `wget`：从网络下载文件

    - 下载一个文件：

```
wget http://example.com/file.txt
```

    - 从命令行下载并保存为特定文件名：

```
wget -O newfilename http://example.com/file.txt
```

7. `ssh`：安全地登录到远程服务器

    - 登录到远程服务器：

```
ssh username@remotehost
```

  - 使用特定端口登录：

    ```
    ssh -p port_number username@remotehost
    ```

8. `scp`：安全地复制文件到远程服务器

   - 复制文件到远程服务器：

     ```
     scp localfile.txt username@remotehost:/path/to/destination/
     ```

   - 从远程服务器复制文件到本地：

     ```
     scp username@remotehost:/path/to/remotefile.txt /local/directory
     ```

9. `df`：报告文件系统的磁盘空间使用情况

   - 显示所有文件系统的磁盘空间使用情况：

     ```
     df -h
     ```

   - 只显示特定文件系统的磁盘空间使用情况：

     ```
     df -h /dev/sda1
     ```

10. `top`：实时显示系统进程

    - 启动 `top` 命令查看系统进程：

      ```
      top
      ```

    - 显示特定用户的进程：

      ```
      top -u username
      ```

这些命令覆盖了文件操作、系统管理、网络操作等多个方面，是Linux用户日常工作中经常使用的命令。

User: 请在上述命令之外给我推荐Linux的高频应用的一些重要命令，每个命令不少于 5个例子

Kimi: 当然，除了你提到的命令外，Linux 还提供了许多其他高频使用的命令。以下是一些额外的重要命令及其使用示例：

1. `ls`：列出目录内容
    - 列出当前目录下的文件和目录：

    ```
    ls
    ```

    - 以长格式列出详细信息：

    ```
    ls -l
    ```

    - 显示隐藏文件：

    ```
    ls -a
    ```

    - 递归列出所有子目录的内容：

    ```
    ls -R
    ```

    - 按修改时间排序：

    ```
    ls -lt
    ```

2. `cd`：改变当前目录
    - 切换到用户的主目录：

    ```
    cd ~
    ```

    - 切换到父目录：

    ```
    cd ..
    ```

    - 切换到特定目录：

    ```
    cd /path/to/directory
    ```

- 切换到上一个工作目录：

```
cd -
```

3. `echo`：显示文本或变量内容

    - 显示文本：

    ```
    echo "Hello, World!"
    ```

    - 显示变量的值：

    ```
    echo $PATH
    ```

    - 创建新文件并写入内容：

    ```
    echo "Some text" > filename.txt
    ```

    - 追加文本到现有文件：

    ```
    echo "More text" >> filename.txt
    ```

4. `sed`：流编辑器，用于文本替换

    - 替换文件中所有出现的 "old" 为 "new"：

    ```
    sed 's/old/new/g' filename.txt
    ```

    - 删除文件中的空行：

    ```
    sed '/^$/d' filename.txt
    ```

    - 仅对第3行进行替换：

    ```
    sed '3s/old/new/' filename.txt
    ```

- 打印文件中第5行：

```
sed -n '5p' filename.txt
```

5. `awk`：模式扫描和处理语言

- 打印文件的每一行：

```
awk '{print}' filename.txt
```

- 打印文件的第一列：

```
awk '{print $1}' filename.txt
```

- 打印第2列到最后一列：

```
awk '{print $2,$3,$4,$5,$6,$7,$8,$9}' filename.txt
```

- 计算文件中行数：

```
awk 'END{print NR}' filename.txt
```

6. `sort`：对文本文件的行进行排序

- 默认按字典顺序排序：

```
sort filename.txt
```

- 按数字排序：

```
sort -n filename.txt
```

- 逆序排序：

```
sort -r filename.txt
```

  - 按第2列排序：

```
sort -k2 filename.txt
```

7. `uniq`：报告或忽略重复的行
  - 删除文件中的重复行：

```
sort filename.txt | uniq
```

  - 只显示重复的行：

```
sort filename.txt | uniq -d
```

  - 显示重复行的次数：

```
sort filename.txt | uniq -cd
```

8. `head`：输出文件的开头部分
  - 显示文件的前10行：

```
head filename.txt
```

  - 显示文件的前20行：

```
head -n 20 filename.txt
```

  - 显示文件的前一个字符：

```
head -c 1 filename.txt
```

9. `tail`：输出文件的末尾部分
  - 显示文件的最后10行：

```
tail filename.txt
```

- 显示文件的最后20行：

```
tail -n 20 filename.txt
```

- 显示文件的最后100个字符：

```
tail -c 100 filename.txt
```

10. `chmod`：更改文件或目录的权限
    - 为所有用户添加读权限：

      ```
      chmod a+r filename.txt
      ```

    - 设置文件的权限为只读：

      ```
      chmod 444 filename.txt
      ```

    - 为文件所有者添加执行权限：

      ```
      chmod u+x filename.txt
      ```

这些命令在处理文件、文本和系统管理任务时非常有用。掌握这些命令可以帮助你更高效地使用 Linux 系统。

- 我重点学习的命令：`sort`、`echo`、`sed`：
  - `echo`：

```
root@Youer:~# touch example.txt
root@Youer:~# echo "Hello, World!" > example.txt
root@Youer:~# echo "This is another line." >> example.txt
root@Youer:~# cat example.txt
Hello, World!
This is another line.
root@Youer:~# echo "Hello, World!"
Hello, World!
root@Youer:~# echo "New line \n and tab \t characters."
New line \n and tab \t characters.
root@Youer:~# echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
:/usr/local/games:/usr/lib/wsl/lib:/mnt/c/ProgramFiles/CommonFiles/Orac
le/Java/javapath:/mnt/c/Windows/system32:/mnt/c/Windows:/mnt/c/Windows/
System32/Wbem:/mnt/c/Windows/System32/WindowsPowerShell/v1.0/:/mnt/c/Wi
ndows/System32/OpenSSH/:/mnt/c/WINDOWS/system32:/mnt/c/WINDOWS:/mnt/c/W
INDOWS/System32/Wbem:/mnt/c/WINDOWS/System32/WindowsPowerShell/v1.0/:/m
nt/c/WINDOWS/System32/OpenSSH/:/mnt/d/git/Git/cmd:/mnt/c/Program
Files/Java/jdk-17/bin:/mnt/c/Program Files/NVIDIA Corporation/NVIDIA
NvDLISR:/mnt/c/Program Files (x86)/NVIDIA
Corporation/PhysX/Common:/mnt/c/Program Files/dotnet/:/mnt/c/Program
Files (x86)/Windows Kits/10/Windows Performance Toolkit/:/mnt/c/Program
Files (x86)/Microsoft SQL Server/160/Tools/Binn/:/mnt/c/Program
Files/Microsoft SQL Server/160/Tools/Binn/:/mnt/c/Program
Files/Microsoft SQL Server/Client
SDK/ODBC/170/Tools/Binn/:/mnt/c/Program Files/Microsoft SQL
Server/160/DTS/Binn/:/mnt/c/Program Files (x86)/Microsoft SQL
Server/160/DTS/Binn/:/mnt/d/tomcat/apache-tomcat-
9.0.93/bin:/mnt/c/Users/xlm20/AppData/Local/Microsoft/WindowsApps:/mnt/
c/Users/xlm20/AppData/Local/Programs/Microsoft VS
Code/bin:/mnt/d/xampp/php:/mnt/d/ÊýʌçÈíᴇ
þ/modelsim_ase/win32aloem:/mnt/d/ÊýʌçÈíᴇ
þ/shudian/modelsim_ase/win32aloem:/mnt/c/Users/xlm20/AppData/Local/GitH
ubDesktop/bin:/mnt/d/nmap:/mnt/c/Users/xlm20/.dotnet/tools:/mnt/d/JDK22
/bin:/snap/bin
```

- `sort`:

```
root@Youer:~# sort --help
Usage: sort [OPTION]... [FILE]...
or:  sort [OPTION]... --files0-from=F
Write sorted concatenation of all FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options
too.
Ordering options:

-b, --ignore-leading-blanks  ignore leading blanks
-d, --dictionary-order      consider only blanks and alphanumeric
characters
-f, --ignore-case           fold lower case to upper case characters
-g, --general-numeric-sort  compare according to general numerical
value
-i, --ignore-nonprinting    consider only printable characters
-M, --month-sort            compare (unknown) < 'JAN' < ... < 'DEC'
-h, --human-numeric-sort    compare human readable numbers (e.g., 2K
1G)
-n, --numeric-sort          compare according to string numerical value
-R, --random-sort           shuffle, but group identical keys.  See
shuf(1)
```

```
      --random-source=FILE    get random bytes from FILE
-r, --reverse               reverse the result of comparisons
      --sort=WORD             sort according to WORD:
                              general-numeric -g, human-numeric -h,
month -M,
                              numeric -n, random -R, version -V
-V, --version-sort          natural sort of (version) numbers within
text

Other options:

      --batch-size=NMERGE   merge at most NMERGE inputs at once;
                           for more use temp files
-c, --check, --check=diagnose-first  check for sorted input; do not
sort
-C, --check=quiet, --check=silent  like -c, but do not report first bad
line
      --compress-program=PROG  compress temporaries with PROG;
                              decompress them with PROG -d
      --debug               annotate the part of the line used to sort,
                              and warn about questionable usage to
stderr
      --files0-from=F       read input from the files specified by
                           NUL-terminated names in file F;
                           If F is - then read names from standard
input
-k, --key=KEYDEF          sort via a key; KEYDEF gives location and
type
-m, --merge               merge already sorted files; do not sort
-o, --output=FILE         write result to FILE instead of standard
output
-s, --stable              stabilize sort by disabling last-resort
comparison
-S, --buffer-size=SIZE    use SIZE for main memory buffer
-t, --field-separator=SEP  use SEP instead of non-blank to blank
transition
-T, --temporary-directory=DIR  use DIR for temporaries, not $TMPDIR or
/tmp;
                              multiple options specify multiple
directories
      --parallel=N          change the number of sorts run concurrently
to N
-u, --unique              with -c, check for strict ordering;
                              without -c, output only the first of an
equal run
-z, --zero-terminated     line delimiter is NUL, not newline
      --help     display this help and exit
      --version  output version information and exit

KEYDEF is F[.C][OPTS][,F[.C][OPTS]] for start and stop position, where
F is a
field number and C a character position in the field; both are origin
1, and
the stop position defaults to the line's end.  If neither -t nor -b is
```

```
in
effect, characters in a field are counted from the beginning of the
preceding
whitespace.  OPTS is one or more single-letter ordering options
[bdfgiMhnRrV],
which override global ordering options for that key.  If no key is
given, use
the entire line as the key.  Use --debug to diagnose incorrect key
usage.

SIZE may be followed by the following multiplicative suffixes:
% 1% of memory, b 1, K 1024 (default), and so on for M, G, T, P, E, Z,
Y.

*** WARNING ***
The locale specified by the environment affects sort order.
Set LC_ALL=C to get the traditional sort order that uses
native byte values.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/sort>
or available locally via: info '(coreutils) sort invocation'
root@Youer:~# cat example.txt
Hello, World!
This is another line.
root@Youer:~# sort example.txt
Hello, World!
This is another line.
root@Youer:~# sort -n example.txt
Hello, World!
This is another line.
root@Youer:~# sort -r example.txt
This is another line.
Hello, World!
root@Youer:~# sort -k2 example.txt
Hello, World!
This is another line.
```

- `sed`:

```
root@Youer:~# cat example.txt
Hello, World!
This is another line.
root@Youer:~# sed 's/line/new_line/g' example.txt
Hello, World!
This is another new_line.
root@Youer:~# echo "" >> example.txt
root@Youer:~# cat example.txt
Hello, World!
This is another line.
```

```
root@Youer:~# sed '/^$/d' example.txt
Hello, World!
This is another line.
root@Youer:~# sed -n '2p' example.txt
This is another line.
root@Youer:~# sed '1s/o/xlm/' example.txt
Hellxlm, World!
This is another line.

root@Youer:~# cat example.txt
Hello, World!
This is another line.

root@Youer:~# sed -i '/^$/d' example.txt
root@Youer:~# cat example.txt
Hello, World!
This is another line.
root@Youer:~# sed -i 's/line/new_line/g' example.txt
root@Youer:~# cat example.txt
Hello, World!
This is another new_line.
root@Youer:~#
```

- 总结：echo、sort和sed都与文本处理有关，其中echo主要用于输出和插入数据，sort功能专一，用于排序，sed可以进行替换、删除特殊字符等复杂的文本处理。