《Linux 系统基础》实验报告

第一次实验：

姓名：周帛岑

学号：221900309

2022 级智能软件与工程学院/系

邮箱：221900309@nju.edu.com

时间：2024 年 7 月 6 日

一、实验内容

本次实验通过在 shell 终端直接键入命令进行完成，涉及到的命令包括 cd，ls，touch，mkdir，chmod，cat，echo 等

主要任务：

1) 在/tmp 下新建一个名为 test 的目录。

2) 用命令 man 查看命令 touch 的使用手册。

3) 用命令 touch 在 test 目录中新建一个名为 test 的文件。

4) 用命令 echo 将以下内容一行一行地写入 test 文件。

```
#!/bin/sh
curl --head --silent https://www.nju.edu.cn
```

5) 尝试执行这个文件，即将该脚本的路径（./test）输入到您的 shell 中并回车。如果程序无法执行，请使用 ls 命令来获取信息并给出其不能执行的原因。

6) 查看命令 chmod 的手册，使用命令 chmod 改变 test 文件的权限，使 ./test 能够成功执行，不要使用 sh test 来执行该程序。

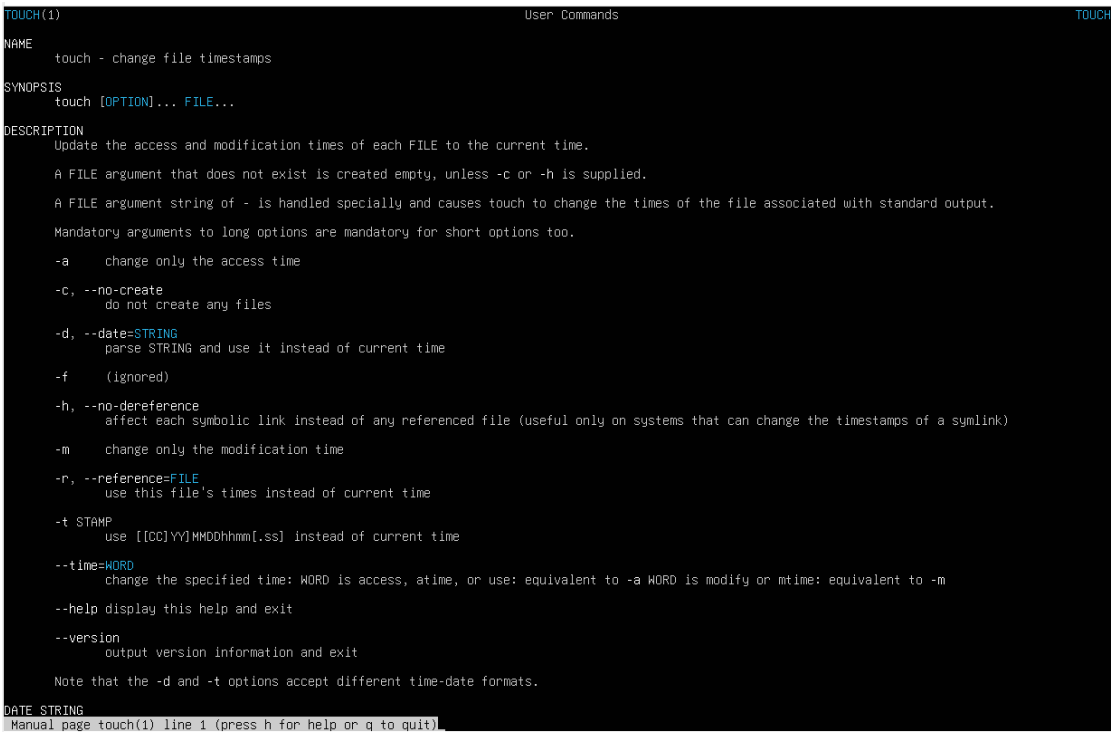
7) 请问你的 shell 是如何知道这个文件需要使用 sh 来解析的。请通过网络搜索“unix shebang”来了解更多信息。

8) 请使用 | 和 > ，将 test 文件输出的最后 5 行内容写入自己主目录下的 last-5-lines.txt 文件中。

二、实验结果

1) 在/tmp 下新建一个名为 test 的目录。

2) 用命令 man 查看命令 touch 的使用手册。



3) 用命令 touch 在 test 目录中新建一个名为 test 的文件。

```
pfdx@pfdx:/tmp/test$ touch test
pfdx@pfdx:/tmp/test$ ls
test
pfdx@pfdx:/tmp/test$
```

4) 用命令 echo 将以下内容一行一行地写入 test 文件。

#!/bin/sh

curl --head --silent https://www.nju.edu.cn

```
pfdx@pfdx:/tmp$ echo '#!/bin/sh' > test
pfdx@pfdx:/tmp$ echo 'curl --head --silent https://www.nju.edu.cn' >> test
pfdx@pfdx:/tmp$ cat test
#!/bin/sh
curl --head --silent https://www.nju.edu.cn
pfdx@pfdx:/tmp$
```

5) 尝试执行这个文件，即将该脚本的路径（./test）输入到您的 shell 中并回车。如果程序无法执行，请使用 ls 命令来获取信息并给出其不能执行的原因。

```
pfdx@pfdx:/tmp/test$ sudo ./test
sudo: ./test: command not found
pfdx@pfdx:/tmp/test$ ls
test
pfdx@pfdx:/tmp/test$ ls -l test
-rw-rw-r-- 1 pfdx pfdx 52 Jul  6 06:52 test
pfdx@pfdx:/tmp/test$ _
```

。

输出中-rw-rw-rr 中，可以得知，无论是谁均没有执行 test 文件的权限

6) 查看命令 chmod 的手册，使用命令 chmod 改变 test 文件的权限，使 ./test 能够成功执行，不要使用 sh test 来执行该程序。

```
CHMOD(1)                                  User Commands                                  CHMOD(1)

NAME
       chmod - change file mode bits

SYNOPSIS
       chmod [OPTION]... MODE[,MODE]... FILE...
       chmod [OPTION]... OCTAL-MODE FILE...
       chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION
       This  manual  page  documents the GNU version of chmod.  chmod changes the file mode bits of each given file according to mode, which can be either a
       symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.

       The format of a symbolic mode is [ugoa...][[-+=][perms...]...], where perms is either zero or more letters from the set rwxXst, or  a  single  letter
       from the set ugo.  Multiple symbolic modes can be given, separated by commas.

       A combination of the letters ugoa controls which users' access to the file will be changed: the user who owns it (u), other users in the file's group
       (g),  other  users  not in the file's group (o), or all users (a).  If none of these are given, the effect is as if (a) were given, but bits that are
       set in the umask are not affected.

       The operator + causes the selected file mode bits to be added to the existing file mode bits of each file; - causes them to be removed; and =  causes
       them to be added and causes unmentioned bits to be removed except that a directory's unmentioned set user and group ID bits are not affected.

       The letters rwxXst select file mode bits for the affected users: read (r), write (w), execute (or search for directories) (x), execute/search only if
       the  file  is  a  directory  or  already has execute permission for some user (X), set user or group ID on execution (s), restricted deletion flag or
       sticky bit (t).  Instead of one or more of these letters, you can specify exactly one of the letters ugo: the permissions granted  to  the  user  who
       owns  the  file (u), the permissions granted to other users who are members of the file's group (g), and the permissions granted to users that are in
       neither of the two preceding categories (o).

       A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1.  Omitted digits are assumed to be lead-
       ing zeros.  The first digit selects the set user ID (4) and set group ID (2) and restricted deletion or sticky (1) attributes.  The second digit  se-
       lects  permissions  for the user who owns the file: read (4), write (2), and execute (1); the third selects permissions for other users in the file's
       group, with the same values; and the fourth for other users not in the file's group, with the same values.

       chmod never changes the permissions of symbolic links; the chmod system call cannot change their permissions.  This is not a problem since  the  per-
       missions of symbolic links are never used.  However, for each symbolic link listed on the command line, chmod changes the permissions of the pointed-
       to file.  In contrast, chmod ignores symbolic links encountered during recursive directory traversals.

SETUID AND SETGID BITS
       chmod  clears the set-group-ID bit of a regular file if the file's group ID does not match the user's effective group ID or one of the user's supple-
       mentary group IDs, unless the user has appropriate privileges.  Additional restrictions may cause the set-user-ID and set-group-ID bits  of  MODE  or
       RFILE to be ignored.  This behavior depends on the policy and functionality of the underlying chmod system call.  When in doubt, check the underlying
       system behavior.

       For  directories  chmod preserves set-user-ID and set-group-ID bits unless you explicitly specify otherwise.  You can set or clear the bits with sym-
       bolic modes like u+s and g-s.  To clear these bits for directories with a numeric mode requires an additional leading zero like 00755, leading  minus
       like -6000, or leading equals like =755.

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

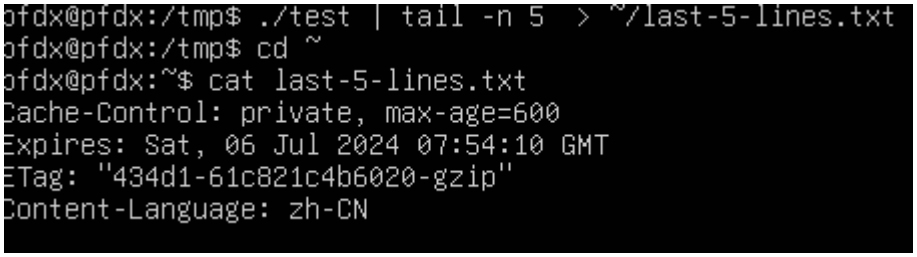
```
pfdx@pfdx:/tmp$ chmod +x test
pfdx@pfdx:/tmp$ ls -l test
-rwxrwxr-x 1 pfdx pfdx 54 Jul  6 07:18 test
pfdx@pfdx:/tmp$ ./test
HTTP/1.1 200 OK
Date: Sat, 06 Jul 2024 07:19:13 GMT
Content-Type: text/html
Content-Length: 275665
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Referer-Policy: no-referer-when-downgrade
X-Download-Options: noopen
X-Permitted-Cross-Domain-Policies: master-only
Last-Modified: Fri, 05 Jul 2024 15:56:46 GMT
Accept-Ranges: bytes
Vary: User-Agent,Accept-Encoding
Cache-Control: private, max-age=600
Expires: Sat, 06 Jul 2024 07:48:27 GMT
ETag: "434d1-61c821c4b6020-gzip"
Content-Language: zh-CN

pfdx@pfdx:/tmp$
```

7) 请问你的 shell 是如何知道这个文件需要使用 sh 来解析的。请通过网络搜索“unix shebang”来了解更多信息。

在 Unix 和 Linux 系统中，Shell 脚本的解析是由脚本文件的第一行决定的。这一行通常被称为“shebang”（#!），用于指示文件的解释器。而在 test 文件中，第一行指定了脚本解释器的位置。

8) 请使用 | 和 > ，将 test 文件输出的最后 5 行内容写入自己主目录下的 last-5-lines.txt 文件中。



三、实验中遇到的问题及解决方案

不知道如何只取用输出的最后五行，通过网上搜索的方式得到了使用 **tail -n x** 指令

四、实验的启示/意见和建议

约 20 分钟