

1) 在/tmp 下新建一个名为 test 的目录。

命令：

cd /tmp

mkdir test

```
(kali@kali)-[/tmp]
$ ls
config-err-qi2NOT
ssh-zaGm00GUseic
systemd-private-67ff084aff7c45fda967385e9682ac93-color.service-AMkZSi
systemd-private-67ff084aff7c45fda967385e9682ac93-haveged.service-YRRkJB
systemd-private-67ff084aff7c45fda967385e9682ac93-ModemManager.service-5WT70G
systemd-private-67ff084aff7c45fda967385e9682ac93-polkit.service-rD8Ldg
systemd-private-67ff084aff7c45fda967385e9682ac93-systemd-logind.service-3760N
C
systemd-private-67ff084aff7c45fda967385e9682ac93-upower.service-huJv2W
test
```

2) 用命令 man 查看命令 touch 的使用手册

man touch

```
kali@kali: /tmp
File Actions Edit View Help
TOUCH(1) User Commands TOUCH(1)

NAME
    touch - change file timestamps

SYNOPSIS
    touch [OPTION] ... FILE ...

DESCRIPTION
    Update the access and modification times of each FILE to the current
    time.

    A FILE argument that does not exist is created empty, unless -c or -h
    is supplied.

    A FILE argument string of - is handled specially and causes touch to
    change the times of the file associated with standard output.

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

    -a      change only the access time

    -c, --no-create
            do not create any files

    -d, --date=STRING

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

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

cd test

touch test

```
(kali㉿kali)-[/tmp/test]
$ ls
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
```

```
echo '#!/bin/sh' > test
```

```
echo 'curl --head --silent https://www.nju.edu.cn' >> test
```

```
(kali㉿kali)-[/tmp/test]
$ echo '#!/bin/sh' > test

(kali㉿kali)-[/tmp/test]
$ cat test
#!/bin/sh

(kali㉿kali)-[/tmp/test]
$ echo 'curl --head --silent https://www.nju.edu.cn' >> test

(kali㉿kali)-[/tmp/test]
$ cat test
#!/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。

```
(kali㉿kali)-[/tmp/test]
$ ./tmp/test/test
zsh: permission denied: /tmp/test/test
```

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。

```
kali@kali: /tmp/test
File Actions Edit View Help
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 num-
  ber 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 rwXst, or a single
  letter from the set ugo. Multiple symbolic modes can be given, sepa-
  rated 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.

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

```
(kali@kali)-[/tmp/test]
$ chmod u+x test

(kali@kali)-[/tmp/test]
$ ls -l test
-rwxrw-r-- 1 kali kali 54 Sep 14 16:18 test

(kali@kali)-[/tmp/test]
$ ./test
HTTP/1.1 200 OK
Date: Sun, 14 Sep 2025 08:29:12 GMT
Content-Type: text/html
Content-Length: 300524
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: Sun, 14 Sep 2025 08:22:01 GMT
Accept-Ranges: bytes
Vary: User-Agent,Accept-Encoding
Cache-Control: private, max-age=600
Expires: Sun, 14 Sep 2025 08:39:12 GMT
```

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。

Sheban 是放在脚本文件第一行的一个特殊字符序列, 它由两个字符组成: 一个井号 (#) 和一个叹号 (!), 看起来像这样: #!。它的作用是告诉系统应该使用哪个解释器来执行这个脚本。

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。

```
(kali@kali)-[/tmp/test]
$ ./test | sed -n "14,18p" > /home/kali/last_5_lines.txt

(kali@kali)-[/tmp/test]
$ cat /home/kali/last_5_lines.txt
Vary: User-Agent,Accept-Encoding
Cache-Control: private, max-age=600
Expires: Sun, 14 Sep 2025 09:06:13 GMT
ETag: "495ec-63ebe94cdf605-gzip"
Content-Language: zh-CN
```