

1. 问题重述

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。

2. 实验报告

2.1 实验过程

1. 写入test文件

在使用echo #!/bin/sh >>test或者echo "#!/bin/sh" 的时候，都会出现这样的错误：

```
└─(kali㉿kali)-[/tmp/test]
└─$ echo "#!/bin/sh"
zsh: event not found: /bin/sh
```

Fence 1

经过查询，发现“!”是一个特殊的符号，用于shell的histroy expansion。可以和其他的字符相结合，展开得到之前的commands。

比如说：

```
└─(kali㉿kali)-[/tmp/test]
└─$ echo "HelloWorld"
HelloWorld
```

```
└─(kali㉿kali)-[/tmp/test]
└─$ !!
```

```
└─(kali㉿kali)-[/tmp/test]
└─$ echo "HelloWorld"
```

Fence 2

而“!/bin/sh”是在查找之前以“/bin/sh”结尾的名字，结果是no found。

Solution: 使用单引号 (Single Quotes)

2. 执行test

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

(kali㉿kali)-[/tmp/test]
└─$ ls -l
total 4
-rw-rw-r-- 1 kali kali 54 Sep  9 07:09 test
```

Fence 3

发现报错，检查文件权限发现，test文件没有执行权限。

3. 查询手册 (touch、chmod)

1. touch指令用于修改文件的时间戳，不会对文件的内容产生影响，因此可以用于创建空文件
2. chmod指令的作用是Change file mode bits, 改变文件的权限。比如chmod +x test，就可以让test文件可以执行

4. 再次运行test

```
(kali㉿kali)-[/tmp/test]
└─$ chmod +x test

(kali㉿kali)-[/tmp/test]
└─$ ./test
HTTP/1.1 200 OK
Date: Tue, 09 Sep 2025 11:21:45 GMT
Content-Type: text/html
Content-Length: 236052
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: Tue, 09 Sep 2025 09:56:44 GMT
Accept-Ranges: bytes
Vary: User-Agent,Accept-Encoding
Cache-Control: private, max-age=600
Expires: Tue, 09 Sep 2025 11:31:45 GMT
ETag: "39a14-63e5b524d594d-gzip"
Content-Language: zh-CN
```

Fence 4

5. 查询shebang

- o gemini: “The string `#!/bin/sh` itself is known as a **shebang** (a combination of "hash" and "bang"). When placed as the very first line of a script file, it tells the operating system which interpreter to use to execute the script. In this case, it specifies that the script should be run using the `/bin/sh` shell. This is a crucial element for making a script directly executable.”
- o google: When a text file with a shebang is used as if it were an executable in a [Unix-like](#) operating system, the [program loader](#) mechanism parses the rest of the file's initial line as an interpreter directive. The loader executes the specified [interpreter](#) program, passing to it as an argument the path that was initially used when attempting to run the script, so that the program may use the file as input data.[8] For example, if a script is named with the path *path/to/script*, and it starts with the line `#!/bin/sh`, then the program loader is instructed to run the program `/bin/sh`, passing *path/to/script* as the first argument.

简单来说，shebang的意思就是“#!/”，放在脚本文件的前面，用于指定执行该文件的interpreter、shell等。

6. 保存输出

```
└─(kali㉿kali)-[/tmp/test]
└─$ ./test
HTTP/1.1 200 OK
Date: Tue, 09 Sep 2025 11:21:45 GMT
Content-Type: text/html
Content-Length: 236052
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: Tue, 09 Sep 2025 09:56:44 GMT
Accept-Ranges: bytes
Vary: User-Agent,Accept-Encoding
Cache-Control: private, max-age=600
Expires: Tue, 09 Sep 2025 11:31:45 GMT
ETag: "39a14-63e5b524d594d-gzip"
Content-Language: zh-CN
```

```
└─(kali㉿kali)-[/tmp/test]
└─$ ls
test
```

```
└─(kali㉿kali)-[/tmp/test]
└─$ ./test | tail -n 5 > ~/last-5-lines.txt
```

```
└─(kali㉿kali)-[/tmp/test]
└─$ cat ~/last*
Cache-Control: private, max-age=600
Expires: Tue, 09 Sep 2025 11:43:41 GMT
ETag: "39a14-63e5b524d594d-gzip"
Content-Language: zh-CN
```

Fence 5