文本处理的一些命令

* tr [option]…set1 [set2]

-d:删除和set1匹配的字符 -s：去除set1指定的重复字符

举例：删除 "hello shiyanlou" 中所有的 'o’’l’’h'

$ echo 'hello shiyanlou' | tr -d 'olh'

# 将"hello" 中的ll,去重为一个l

$ echo 'hello' | tr -s 'l'

# 将输入文本，全部转换为大写或小写输出

$ echo 'input some text here' | tr '[:lower:]' '[:upper:]'

# 上面的'[:lower:]' '[:upper:]'你也可以简单的写作'[a-z]' '[A-Z]',当然反过来将大写变小写也是可以的

* col[option]

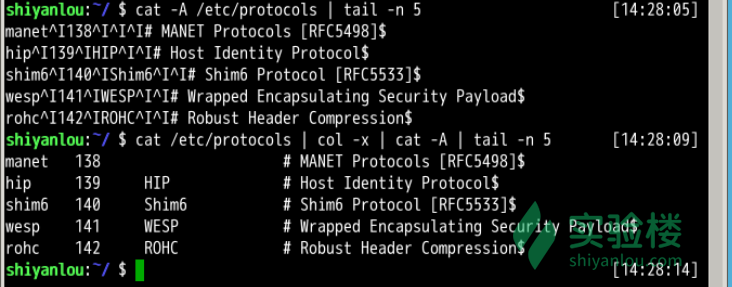
-x：将tab转换为空格 -h:将空格转换为tab

举例：/etc/protocaols中有很多不可见字符，^I就是tab转义成可见字符的符号。

cat –A /etc/protocols

使用col –x 将/etc/protocols中的tab转换为空格，然后再使用cat查看，你发现^I不见了。

cat /etc/protocols |col –x |cat –A



* join命令

用于将两个文件中包含相同内容的那一行合并在一起

Join [option]…file1 file2

| **选项** | **说明** |
| --- | --- |
| -t | 指定分隔符，默认为空格 |
| -i | 忽略大小写的差异 |
| -1 | 指明第一个文件要用哪个字段来对比，默认对比第一个字段 |
| -2 | 指明第二个文件要用哪个字段来对比，默认对比第一个字段 |

举例：$ cd /home/shiyanlou

# 创建两个文件

$ echo '1 hello' > file1

$ echo '1 shiyanlou' > file2

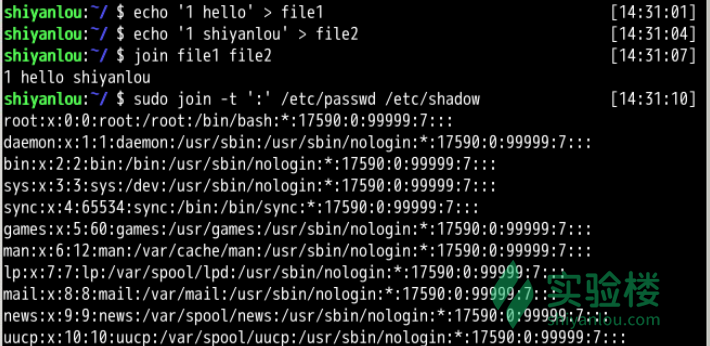
$ join file1 file2

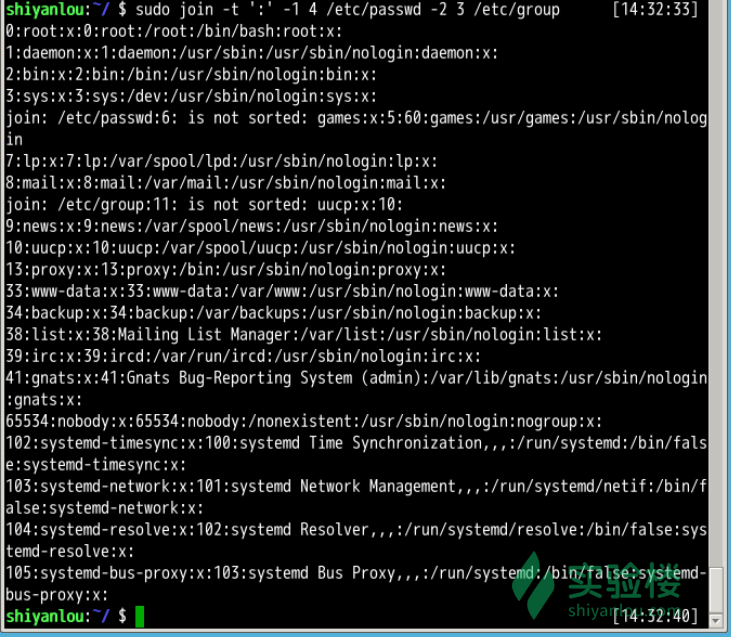
# 将/etc/passwd与/etc/shadow两个文件合并，指定以':'作为分隔符

$ sudo join -t':' /etc/passwd /etc/shadow

# 将/etc/passwd与/etc/group两个文件合并，指定以':'作为分隔符, 分别比对第4和第3个字段

$ sudo join -t':' -1 4 /etc/passwd -2 3 /etc/group





(问题：看不懂这个图！！！)

* paste[option] file… 这个命令与join命令类似，在不对比数据的情况下简单的将多个文件合并在一起，以tab隔开

| **选项** | **说明** |
| --- | --- |
| -d | 指定合并的分隔符，默认为Tab |
| -s | 不合并到一行，每个文件为一行 |

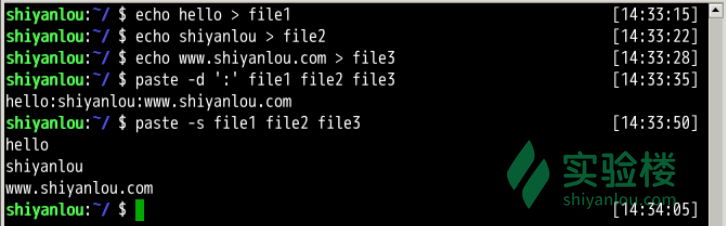
echo hello > file1

$ echo shiyanlou > file2

$ echo www.shiyanlou.com > file3

$ paste -d ':' file1 file2 file3

$ paste -s file1 file2 file3



| **文件描述符** | **设备文件** | **说明** |
| --- | --- | --- |
| 0 | /dev/stdin | 标准输入 |
| 1 | /dev/stdout | 标准输出 |
| 2 | /dev/stderr | 标准错误 |

* **管道默认是连接前一个命令的输出到下一个命令的输入**，而重定向通常是需要一个文件来建立两个命令的连接

$ mkdir Documents

$ cat > Documents/test.c <<EOF

Hello World

EOF(按EOF退出)

这里的>就是重定向

$ echo 'hello shiyanlou' > redirect

$ cat redirect 将echo命令的输出从默认的标准输出重定向到一个普通文件

* 要隐藏某些错误或者警告，需要添加修饰符

将标准错误重定向到标准输出，再将标准输出重定向到文件，注意要将重定向到文件写到前面

$ cat Documents/test.c hello.c >somefile 2>&1

# 或者只用bash提供的特殊的重定向符号"&"将标准错误和标准输出同时重定向到文件

$ cat Documents/test.c hello.c &>somefilehell

**注意你应该在输出重定向文件描述符前加上&,否则shell会当做重定向到一个文件名为1的文件中**

* 将输出重定向到文件并且打印到终端 可以使用tee

echo ‘hello shiyanlou’ | tee hello