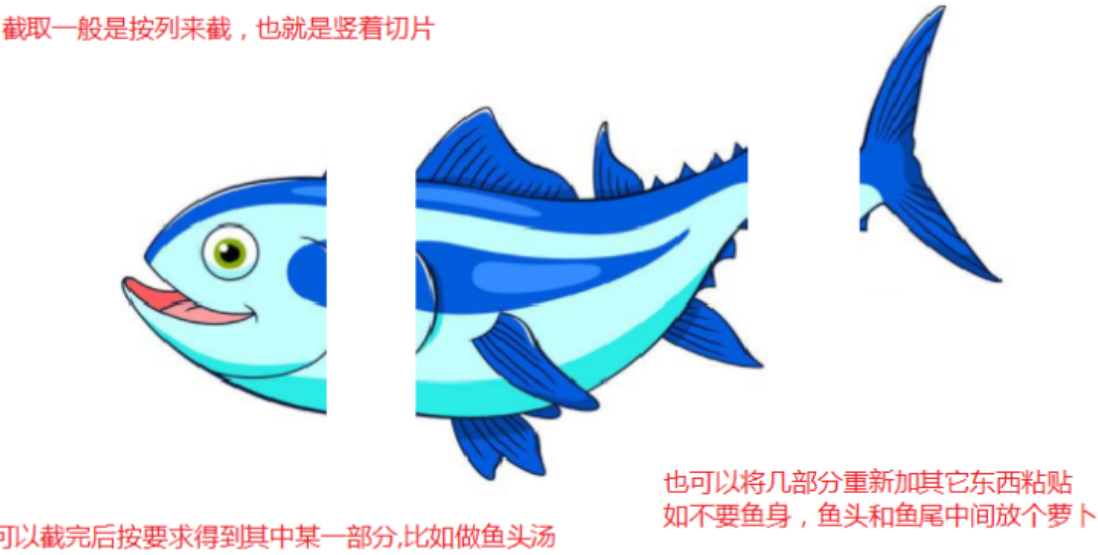


前言

运维工程师经常需要做文本处理，文本处理里有一个操作叫截取，也有些地方叫切片。在学习过程中很多地方都会用到截取(切片)，容易混淆。在这里，我来做一个分析与比较。

什么是截取(切片)?

就是对文本按列处理,比如截取其中的一部分或几部分等等。请看下图。



能做截取(切片)的命令或场景有哪些?

答: 主要有以下几种情况可以实现截取(切片):

- cut
- awk
- sed
- python中的序列(字符串,列表,元组)

cut截取

cut命令做截取主要的优点是简单方便，缺点是只能以单个字符为分隔符，不方便处理多个空格，分隔符不相同的场景。

常见参数	说明
-d	指定分隔符
-f	指定第几列
-c	截取第几个字符

示例:

以":"为分隔符，截取第7列

```
# head -1 /etc/passwd | cut -d: -f7
/bin/bash
```

截取字符串的第2到4个字符

```
# echo 12345 | cut -c2-4
234
```

截取字符串的第2到最后的字符

```
# echo 12345 | cut -c2-
2345
```

awk截取

awk做截取比cut强大很多, 它可以以多个空格或正则表达式为分隔符，所以几乎可以应用到任何截取的场景。

常用变量	说明
\$0	当前处理行的所有记录(所有列数之和，包括分隔符)
\$1 到 \$n	文件中每行以间隔符号分割的不同字段(\$1代表第1列。。。。以此类推)
NF	当前记录的字段数（列数）
\$NF	最后一列
NR	行号
FS	定义间隔符,等同于-F参数

整行为\$0

root:X:0:0:root:/root:/bin/bash

\$1 \$2 分隔符 \$7或者\$NF

示例:

以":"为分隔符截取第1列

```
# head -3 /etc/passwd |awk -F: '{print $1}'  
root  
bin  
daemon
```

以":"为分隔符截取第3列

```
# head -3 /etc/passwd |awk -F: '{print $3}'  
0  
1  
2
```

以"ab"这两个字符为分隔符OK

```
# echo 123ab456ab789 |awk -F"ab" '{print $2}'  
456
```

以逗号或点号为分隔符,截取第2列

```
# echo hello,world.awk |awk -F[,.] '{print $2}'  
world
```

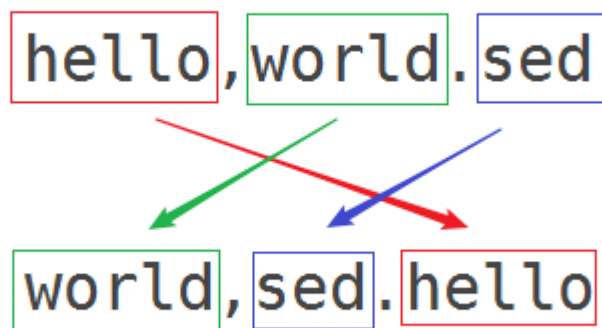
下面这样的字符串,以点为分隔符可以做,但是点号太多,不好数.所以可以用正则表达式,以连续的多个点来为分隔符

```
# echo "haha,hehe.....heihei....." |awk -F"[.]" '{print $2}'  
heihei
```

sed截取

将()之间的字符串(**一般为正则表达式**)定义为组,并且将匹配这个表达式的保存到一个区域(一个正则表达式最多可以保存9个),它们使用\1到\9来表示。然后进行替换操作

示例: 把 `hello,world.sed` 变成 `world,sed.hello` (注意: 1个逗号1个点号)



方法1:

```
# echo "hello,world.sed" | sed -r 's/(.*),(.*).\.(.*)/\2,\3.\1/'
world,sed.hello
```

解析:

将,号前的任意字符划为第1个域,以\1来表示

将,号和.号中间的任意字符划为第2个域,以\2表示

将.号后面的任意字符划分第3个域,以\3表示

方法2:

```
# echo "hello,world.sed" | sed -r 's/(.....)(.)(.....)(.)(...)/\3\2\5\4\1/'
world,sed.hello
```

解析:

几个点就代表几个字符

方法3:

```
# echo "hello,world.sed" | sed -r 's/(.{5})(.)(.{5})(.)(.{3})/\3\2\5\4\1/'
world,sed.hello
```

解析:

.{5}就代表5个字符

python切片

python中的序列(字符串,列表,元组)都可以进行切片操作(截取)

示例: python中字符串切片

```
a = "abcdefg"
print(a[0:3])      # 取第1个到第3个字符 (注意,不包含第4个)
print(a[2:5])      # 取第3个到第5个字符 (注意,不包含第6个)

print(a[0:-1])     # 取第1个到倒数第2个 (注意:不包含最后一个)
print(a[1:])       # 取第2个到最后一个
print(a[:])        # 全取
print(a[0:5:2])    # 取第1个到第5个,但步长为2 (结果为ace)
print(a[::-1])     # 字符串的倒序 (类似shell里的rev命令)
```

python中列表切片

```
os_list = ["rhel", "centos", "suse", "ubuntu"]

print(os_list[1:3])
print(os_list[1:-1])
print(os_list[:2])
print(os_list[::-1])
```

python中元组切片

```
os_list = tuple(["rhel", "centos", "suse", "ubuntu"])
```

```
print(os_list[1:3])
```

```
print(os_list[1:-1])
```

```
print(os_list[:2])
```

```
print(os_list[::-1])
```