链滴

shell 字符串处理 - 从批量替换文件名说起

作者：jianhongli
原文链接：https://ld246.com/article/1554038232423
来源网站：链滴

背景

有一天服务器上的一个服务.发布系统每次发布完成后的jar 包文件名都是带版本号的.由于里面有特殊码,需要读取一个不带版本号的特殊文件名.比如:

demo-1.3.1.jar -> demo.jar

demo-common-2019-02-03-SNAPSHOT.jar ------> demo-common.jar

demo-utils-1.3.2.jar ----> demo-utils.jar

万能的百度与google告诉我,我可以这样实现文件名批量替换:

```
for i in demo-*2017*.jar;do
   sudo mv "$i" "${i%%-2017*.jar}.jar";
done
```

这里用到了${变量%%关键词}功能.此功能的意思是:**若变量内容从尾向前的数据符合"关键词",则将符的最长的数据删除**; 这里的关键词部分使用到了通配符 (*);

字符串字符处理方法

这里介绍的字符串处理都没有使用管道进行多次处理.效率会高非常多.

假设

```
a="demo-2017-02-03-SNASPSHOT.jar"
```

1. 子串截取

表达式	说明	示例	输出
${#string}	$string 的长度	${#a}	9
${string:pos}	从$pos开始提取子串	${a:5}	017-02-03-SNASPSHOT.jar
${string:pos:len}	从$pos开始提取长为 len 的子串	{a:5:4}	2017
${string#substr}	从string开头删除最**短**匹配substr子串	{a#*2}	017-02-03-SNASPSHOT.jar
${string##substr}	从string开头删除最**长**匹配substr子串	{a##*2}	-03-SNASPSHOT.jar
${string%substr}	从string尾删除最**短**匹配substr子串	{a%2*}	demo-2017-0
${string%%substr}	从string尾删除最**长**匹配substr子串	{a%%2*}	demo-

2. 子串替换

表达式	说明	示例	输出

${str/substr/replace}	使用replace 来代替**第一个**匹配的`substr`
${a/2/9}	demo-9017-02-03-SNASPSHOT.jar
${str//substr/replace}	使用replace 来代替**所有**匹配的`substr`
${a//2/9}	demo-9017-09-03-NASPSHOT.jar
${string/#substr/replace}	如果str的**前缀**匹配substr, 那么就用replace来代替匹配到的substr
${a/#*2/xxxx}	xxx-03-SNASPSHOT.jar
${string/%substr/replace}	如果str的**后缀**匹配substr, 那么就用replace来代替匹配到的substr
${a/%2*/xxxx}	demo-xxxx

3. 其它说明

- 以上对字符串进行了简单的替换,截取操作.这些操作都可以在当前进程完成,不需要调用外部命令.也会生成新的进程.
- 上面所有的示例中的相关的偏移以及子串的描述都是直接使用了字面值进行示例的.其实也可以使用量.而且直接写变量名即可.

比如:

```
a="demo-2017-02-03-SNASPSHOT.jar"
bbb=3
ccc=4
echo "${a:bbb:ccc}"
## 输出
o-20
```

变量引用默认值

表达式	说明
${var-DEFAULT}	如果var没有被声明, 那么就以DEFAULT作为其值*
${var:-DEFAULT}	如果var没有被声明, 或者其值空, 那么就以DEFAULT作为其值 *
${var=DEFAULT}	如果var没有被声明, 那么就以DEFAULT作为其值 *
${var:=DEFAULT}	如果var没有被声明, 或者其值空, 那么就以DEFAULT作为其值 *
${var+OTHER}	如果var声明了, 那么其值就是OTHER, 否则就为null字符串
${var:+OTHER}	如果var被设置了, 那么其值就是OTHER, 否则就为null字符串
${var?ERR_MSG}	如果var没被声明, 那么就打印ERR_MSG *
${var:?ERR_MSG}	如果var没被设置, 那么就打印E

R_MSG *

${!varprefix*} 匹配之前所有以varprefix开头进声明的变量

${!varprefix@} 匹配之前所有以varprefix开头进声明的变量

变量设定方式	str 没有设定	str 为空串	st
已设定为非空串	说明		
var=${str-expr}	var=expr	var=	var=$str
就 取 expr			
var=${str:-expr}	var=expr	var=expr	va
=$str	无 或 空取 expr		
var=${str+expr}	var=	var=expr	var=e
pr	有 就 取 expr		
var=${str:+expr}	var=	var=	var=expr
且 不空 取 expr			
var=${str=expr}	str=expr var=expr		var= str 不变
ar=str str 不变	str 无时两者都赋值		
var=${str:=expr}			
var=${str?expr}			
var=${str:?expr}			

参考:

1. https://blog.csdn.net/dongwuming/article/details/50605911
2. https://www.cnblogs.com/fengbohello/p/5954895.html
3. https://blog.csdn.net/x1269778817/article/details/46535729