shell 笔记

作者：lingfei0312
原文链接：https://ld246.com/article/1492675587297
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

系统设定
 默认输出设备：标准输出 stdout， 1
 默认输入设备：标准输入 stdin，
 标准错误输出： stderr， 2
I/O重定向
输出重定向： > 覆盖输出
 >> 追加输出
 2> 错误输
 2>> 错误追加输出
 &> 将标准输出和错误输出到同意文件
输入定向： <
 << here documents ex: cat <<EOF > test.txt
set -C 禁止已存在的文件使用覆盖重定向 >| 强制覆盖输出

grep正则表达式元字符
 .:任意单个字符匹配
 []:指定范围内的单个字符匹配
 [^]:制定范围外的单个字符匹配
 *:匹配其前0次或任意次
 \?:匹配其前字符0次或1
 \{m,n\}:匹配其前的字符至少m次至多n次
 .*:匹配任意长度任意字符
 ^:锚定行首
 $:锚定行尾
 \<,\b:锚定词首
 \>,\b:锚定词尾
\(\):用于分组做后项引用 1 \2 \3
grep 选项
 -i 忽略字符大小写
 -v 反向搜索
 -o 只显示匹配结果
 --color 显示颜色
 -E 使用扩展正则表达式
 -A 显示匹配结果的后n行
 -B 显示配结果的前n行
 -C 显示匹配结果的上下n行
grep扩展正则表达式
 +：匹配其前符至少1次
 分组：
 (): 分组 \1 \2 \3
 a|b: 匹配a或者b(匹配左边字符串或者右边符串)

逻辑运算：与 &、或 | 、非！ 、异或
1：真
0：假
与
1 & 1 = 1
1 & 0 = 0
0 & 1 = 0
0 & 0 = 0
或
1 | 1 = 1
1 | 0 = 1
0 | 1 = 1
0 | 0 = 0
非
脚本在执行时会启动一个子shell进程：
 命令行中启动的脚本会集成当前shell环境变量
 系统自动执行的脚本（非命令行启动）就需要自我定义需要的各环境变量；

bash变量类型：
 环境变量：作用域为当前shell进程及其子进程
 export varname=value
 本地变量（局部变量）
 varname=value : 作用域为整个bash进程
 local varname=value ： 作用域为当前代码段
 位置变量
 $1, $2, ...
 shift
 殊变量
 $?: 上一个命令的执行状态返回值
 $#: 参数的个数
 $*: 参数列表
 @: 参数列表
程序执行，可能有 两类返回值：
 程序执行结果
 程序状态返回代码(0-255)
 0 正确执行
 1-255：错误执行 1，2，127 系统预留
变量名称：
 1、只能包含字母，数字和下划线，并且不能以数字开头
 2、不应该和统中已有的环境变量重名
 3、最好做到见名知义
撤销变量：
unset varname
export varname
查看当前shell中的变量
set
查看当前shell中的环境变量
printenv， env， export
脚本：命令的堆砌，按实际需要，结合命令流程控制机制实现源程序
shebang：魔数
#!/bin/bash

“”强引用(做变量引用)
‘’弱引用(不做变量替换)
引用变量： ${varname} 括号有时可以省略
条件判断
bash中如何实现条件判断
条件测试类型
 整数测试
 字符测
 文件测试
条件测试的表达式
 [expression]
 [[expression]]
 test expression
整数比较：
 -eq：测试两个整数是否相等； 比如 $a -eq $b
 -ne: 测试两个整数是不等；
 -gt：测试一个数是否大于另一个数
 -ge：大于或等于
 -lt：测试一个数否小于另一个数
 -le：小于或等于
命令间的逻辑关系：
 逻辑与： &&
 第一个条件为假时，第二条件不用判断
 第一个条件为真时，第二条件必须再判断
 逻辑或： ||

条件判断，控制结构：
if判断条件； then
 statement1
 statement2
 .。。
fi

shell中的算术运算
a=3
b=4
1、let 算术运算表达式
 let c=$a $b
2、$[算术运算表达式]
 c=$[$a+$b]
3、$((算术运算表达式))
 c=$(($a $b))
4、expr 算术运算表达式，表达式中个操作数及运算符之间要有空，而且要使用命令引用
 c=`expr $a + $b`

文件测试
 -e filename 测试文件是否存在
 -f filename 测试文件是否为普文件
 -d filename 测试是否为目录
 -r filename 测试当前用户对指定文件是否有读取权
 -w
 -x
bash测试脚本是否有语法错误
bash -n scripts
exit: 退出脚本
exit #
如何脚本没有明确定义推出状态码，那么最后执行的一条命的退出码即为脚本的退出码

awk打印前3列
cat video.txt | awk -F '/' '{for(i=1;i<=3;i++)printf $i"/"; printf "\n"}' | uniq
grep 匹配IPV4地址
grep -E -o "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(250-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)" * -R
阿里云centos的yum地址
<pre class="brush: bash; gutter: false">[base]
name=CentOS-$releasever
enabled=1
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-7

[updates]
name=CentOS-$releasever
enabled=1
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/releasever/updates/basearch/
gpgcheck=1
gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-7

[extras]
name=CentOS-$releasever
enabled=1
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/releasever/extras/basearch/
gpgcheck=1
gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-7</pre>

解决selinux审计不通过的问题
yum install setroubleshoot
sealert -a /var/log/audit/audit.log