

晴川のLinux笔记-Linux基础

系统安装

Centos6/7.x

安装准备

1. 下载安装镜像文件
- 2 `http://www.centos.org` ->downloads->mirrors
- 3 `http://mirrors.aliyun.com/centos/6.8/isos/x86_64/`
- 4 `http://mirrors.aliyun.com/centos/6.8/isos/i386/`
- 5 主要下载 `Centos-6.8-x86_64-bin-DVD1.iso` 和 `Centos-6.8-x86_64-bin-DVD2.iso`

安装 CentOS6.8

选择系统引导方式

- 1 选择 `install or upgrade an existing system`

检查安装光盘介质

- 1 选择: `skip`

选择安装过程语言

- 1 选择: `english`

选择键盘布局

- 1 选择: `U.S.English`

选择合适的物理设备

- 1 选择: `basic storage devices`

初始化硬盘提示

- 1 | 选择: yes ,discard and data

初始化主机名以及网络配置

- 1 | (1) . 为系统设置主机名 主机名为: ctenet
- 2 | (2) . 配置网卡及连接网络 (可选)

系统时钟及时区

- 1 | 选择: Asia/Shanghai
- 2 | 取消: system clock uses UTC
- 3 | 然后: next

设置 root 口令

磁盘分区类型选择与磁盘分区配置过程

(1) 选择系统安装磁盘空间类型

- 1 | 选择: create custom layout

(2) 进入 'create custom layout'分区界面

- 1 | 可以 create (创建) ,update (修改) ,delete (删除) 等操作。

(3) 按企业生产标准定制磁盘分区

- 1 | 选择: standard partition
- 2 | 1) . 创建引导分区, /boot 分区
- 3 | mount point:/boot
- 4 | file system type:ext4
- 5 | size:200
- 6 |
- 7 | 2) . 创建 swap 交换分区
- 8 | mount point :<not applicable>
- 9 | file system type:swap
- 10 | size:1024 (物理内存的 1-2 倍)
- 11 | addtion size options : fixed size

```
12 force to be a primary partition
13
14 3). 创建 ( / ) 根分区
15 mount point :/
16 file system type : ext4
17 size : 剩余
18 addition size options : fill to maximum allowable size
   (根分区是最后一个分区, 所以把剩余的空间都分配给根分区)
19 force to be a primary partition
20
21 4). 格式化警告
22 选择: format
```

系统安装包的选择与配置

(1) 启动引导设备的配置 系统默认使用 GRUB 作为启动加载器, 引导程序默认在 MBR 下:

```
1  ``
2  install boot loader on /dev/sda ->change device
3  选择 master boot record -/dev/sda
4  [选择的是操作系统所在的那个设备, 如 /dev/sda]
5
6  Boot Loader operation system list
7  列表中选择的是操作系统根目录 / 所在的分区, 如 CentOS /dev/sda4
8
9  ``
```

(2) 系统安装类型选择及自定义额外包组 系统默认是 desktop , 但是这里选择 minimal。 自定义安装包选择: customsize now base system : base 然后: next

开始安装 ->安装完成 ->reboot

系统安装后的配置

更新系统, 打补丁到最新

```
1 修改更新 yum 源:
2 cp /etc /yum.repos.d/CentOS-Base.repo
  /etc/yum.repos.d/CentOS-Base.repo.ori
3 wget -O /etc/yum.repos.d/CentOS-Base.repo
  http://mirrors.163.com/.help/CentOS-6-Base-163.repo
4 ll /etc/pki/rpm-gpg/
5 rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY*
6 yum update -y
7
8 ps: 一般在首次安装时执行 yum update -y , 如果是在实际生产环境
    中, 切记使用, 以免导致异常。
```

安装额外的软件包

```
1 yum install tree telnet dos2unix sysstat lrzsz nc nmap -
  y
2 yum grouplist      #查看包组列表
3 yum frouplist "development Tools"
```

Bash基础特性

命令历史

(1) 使用命令: history

(2) 环境变量:

```
1      a) HISTSIZE: 命令历史缓冲区中记录的条数, 默认为 1000;
2
3      b) HISTFILE: 记录当前登录用户在 logout 时历史命令存放文件;
4
5      c) HISTFILESIZE: 命令历史文件记录历史的条数, 默认为 1000;
```

(3) 操作命令历史:

- 1 a) `history -d OFFSET` 删除指定行的命令历史;
- 2
- 3 b) `history -c` 清空命令历史缓冲区中的命令;
- 4
- 5 c) `history #` 显示历史中最近的#条命令;
- 6
- 7 d) `history -a` 手动追加当前会话缓冲区中的命令至历史文件中;

(4) 调用历史中的命令:

- 1 a) `!#` : 重复执行第#条命令;
- 2
- 3 b) `!!` : 重复执行上一条 (最近一条命令;)
- 4
- 5 c) `!string` : 重复执行最近一次以指定字符串开头的命令;
- 6
- 7 d) 调用上一条命令的最后一个参数:
- 8
- 9 i. `!$`
- 10
- 11 ii. `ESC, .`

(5) 控制命令历史的记录方式:

- 1 环境变量: `HISTCONTROL`
- 2
- 3 三个值:
- 4
- 5 **ignoredups**: 忽略重复的命令; 所谓重复, 一定是连续且完全相同, 包括选项和参数;
- 6
- 7 **ignorespace**: 忽略所有以空白开头的命令, 不记录;
- 8
- 9 **ignoreboth**: 忽略上述两项, 既忽略重复的命令, 也忽略空白开头的命令;

修改环境变量的方式:

```
1 export 变量名 =“VALUE”  
2  
3 或:  VARNAME=“VALUE” export VARNAME
```

命令补全

内部命令：直接通过 shell 补全； 外部命令：bash 根据 PATH 环境变量定义的路径，自左而右地在每个路径搜寻以给定命令命名的文件，第一次找到即为要执行的命令；

- Note: 在第一次通过 PATH 搜寻到命令后，会将其存入 hash 缓存中，下次使用不再搜寻 PATH，从 hash 中查找；

```
1 [root@sslinux ~]# hash  
2 hits command  
3 1 /usr/sbin/ifconfig  
4 1 /usr/bin/vim  
5 1 /usr/bin/ls
```

Tab 键补全：若用户给出的字符在命令搜索路径中有且仅有一条命令与之相匹配，则 Tab 键直接补全；

若用户输入的字符在命令搜索路径中有多条命令与之相匹配，则再次 Tab 键可以将这些命令列出；

路径补全

以用户输入的字符串作为路径开头，并在其指定路径的上级目录下搜索以指定字符串开头的文件名；

```
1 如果唯一，则直接补全；  
2  
3 否则，再次 Tab，列出所有符合条件的路径及文件；
```

命令行展开

- 1) ~：展开为用户的主目录；

```
1 [root@sslunix log]# pwd
2 /var/log
3 [root@sslunix log]# cd ~
4 [root@sslunix ~]# pwd
5 /root
```

2) ~USERNAME : 展开为指定用户的主目录;

```
1 [root@sslunix ~]# pwd
2 /root
3 [root@sslunix ~]# cd ~sslunix
4 [root@sslunix sslunix]# pwd
5 /home/sslunix
```

3) {} : 可承载一个以逗号分隔的列表, 并将其展开为多个路径;

```
1 [root@localhost test]# ls
2 [root@localhost test]# mkdir -pv ./tmp/{a,b}/shell
3 mkdir: created directory './tmp'
4 mkdir: created directory './tmp/a'
5 mkdir: created directory './tmp/a/shell'
6 mkdir: created directory './tmp/b'
7 mkdir: created directory './tmp/b/shell'
8 [root@localhost test]# mkdir -pv ./tmp/{tom,johnson}/hi
9 [root@localhost test]# tree .
10
11 └─ tmp
12   └─ a
13     └─ shell
14   └─ b
15     └─ shell
16   └─ johnson
17     └─ hi
18   └─ tom
19     └─ hi
20 9 directories, 0 files
```

命令的执行状态结果

表示命令是否成功执行；

bash 使用特殊变量 `$?` 保存最近一条命令的执行状态结果；

- 环境变量 `$?` 的取值：

0：成功；

1-255：失败，1,127,255 为系统保留；

- 程序执行有两类结果：

程序的返回值；程序自身执行的输出结果；

程序的执行状态结果；`$?`

```
1 [root@localhost test]# ls /etc/sysconfig/
2
3 [root@localhost test]# echo $?
4
5 0      #程序的执行状态结果；执行成功；
6
7 [root@localhost test]# ls /etc/sysconfig/NNNN
8
9 ls: cannot access /etc/sysconfig/NNNN: No such file or
   directory      #程序自身的执行结果；
10
11 [root@localhost test]# echo $?
12
13 2      #执行失败；
```

命令别名

- 通过 alias 命令实现：

a、alias：显示当前 shell 进程所有可用的命令别名；

b、定义别名，格式为：alias NAME='VALUE'

```
1 | 定义别名 NAME，其执行相当于执行命令 VALUE，VALUE 中可包含命令、
   选项以及参数；仅当前会话有效，不建议使用；
```


c、通过修改配置文件定义命令别名：

- 1 当前用户： ~/.bashrc
- 2 全局用户： /etc/bashrc

- Bash 进程重新读取配置文件：

```
1 source /path/to/config_file
2
3 . /path/to/config_file
```

- 撤销别名： unalias

```
1 unalias [-a] name [name...]
```

- Note:

对于定义了别名的命令，要使用原命令，可通过、COMMAND 的方式使用；

- Example:

```
1 [root@sslinux sslinux]# alias
2 alias cp='cp -i'
3 alias egrep='egrep --color=auto'
4 alias fgrep='fgrep --color=auto'
5 alias grep='grep --color=auto'
6 alias ll='ls -d .* --color=auto'
7 alias ll='ls -l --color=auto'
8 alias ls='ls --color=auto'
9 alias mv='mv -i'
10 alias rm='rm -i'
11 alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
12 [root@sslinux sslinux]# grep alias /root/.bashrc
13 ### User specific aliases and functions
14 alias rm='rm -i'
15 alias cp='cp -i'
16 alias mv='mv -i'
```

通配符 glob

Bash 中用于文件名"通配"

- 通配符: *,?,[],

\1) * 任意长度的任意字符;

```
1 | a * b
```

```
[root@sslinux sslinux]# ls -ld /etc/au*  
drwxr-x---. 3 root root 41 Sep 3 22:05 /etc/audisp  
drwxr-x---. 3 root root 79 Sep 3 22:09 /etc/audit
```

```
1 |
```

\2) ? 任意单个字符;

```
1 | a?b  
2 [root@sslinux sslinux]# ls -ld /etc/*d?t  
3 drwxr-x---. 3 root root 79 Sep 3 22:09 /etc/audit
```

\3) [] 匹配指定范围内的任意单个字符;

```
1 | [0-9] [a-z] 不区分大小写;  
2 [admin] 可以是区间形式的,也可以是离散形式的;  
3 [root@sslinux sslinux]# ls -ld /etc/[ab]*  
4 drwxr-xr-x. 2 root root 4096 Sep 3 22:05  
  /etc/alternatives  
5 drwxr-xr-x. 2 root root 33 Sep 3 22:04 /etc/avahi  
6 drwxr-xr-x. 2 root root 33 Sep 3 22:04  
  /etc/bash_completion.d  
7 -rw-r--r--. 1 root root 2835 Oct 29 2014 /etc/bashrc  
8 drwxr-xr-x. 2 root root 6 Mar 6 2015 /etc/binfmt.d
```

\4) 匹配指定范围以外的任意单个字符;

```
1 | [^0-9] : 单个非数字的任意字符;
```

- 专用字符结合: (表示一类字符中的单个)

[digit:] 任意单个数字，相当于 [0-9];

[lower:] 任意单个小写字母;

[upper:] 任意单个大写字母;

[alpha:] 任意单个大小写字母;

[alnum:] 任意单个数字或字母;

[space:] 任意空白字符;

[punct:] 任意单个特殊字符;

- Note:

在使用 [] 应用专用字符集合时，外层也需要嵌套 []。

Example:

```
1 # ls -d /etc/*[[digit:]]*[[lower:]]
```

bash快捷键

编辑命令

- Ctrl + a : 移到命令行首
- Ctrl + e : 移到命令行尾
- Ctrl + f : 按字符前移（右向）
- Ctrl + b : 按字符后移（左向）
- Alt + f : 按单词前移（右向）
- Alt + b : 按单词后移（左向）
- Ctrl + xx: 在命令行首和光标之间移动
- Ctrl + u : 从光标处删除至命令行首
- Ctrl + k : 从光标处删除至命令行尾
- Ctrl + w : 从光标处删除至字首
- Alt + d : 从光标处删除至字尾
- Ctrl + d : 删除光标处的字符
- Ctrl + h : 删除光标前的字符
- Ctrl + y : 粘贴至光标后
- Alt + c : 从光标处更改为首字母大写的单词

- Alt + u : 从光标处更改为全部大写的单词
- Alt + l : 从光标处更改为全部小写的单词
- Ctrl + t : 交换光标处和之前的字符
- Alt + t : 交换光标处和之前的单词
- Alt + Backspace: 与 Ctrl + w 相同类似, 分隔符有些差别

重新执行命令

- Ctrl + r: 逆向搜索命令历史
- Ctrl + g: 从历史搜索模式退出
- Ctrl + p: 历史中的上一条命令
- Ctrl + n: 历史中的下一条命令
- Alt + .: 使用上一条命令的最后一个参数

控制命令

- Ctrl + l: 清屏
- Ctrl + o: 执行当前命令, 并选择上一条命令
- Ctrl + s: 阻止屏幕输出
- Ctrl + q: 允许屏幕输出
- Ctrl + c: 终止命令
- Ctrl + z: 挂起命令

Bang (!) 命令

- !!: 执行上一条命令
- !blah: 执行最近的以 blah 开头的命令, 如 !!s
- !blah:p: 仅打印输出, 而不执行
- !\$: 上一条命令的最后一个参数, 与 Alt + . 相同
- !: *p*: 打印输出! 的内容
- !*: 上一条命令的所有参数
- !: *p*: 打印输出! 的内容
- ^blah: 删除上一条命令中的 blah
- blahfoo: 将上一条命令中的 blah 替换为 foo
- blahfoo^: 将上一条命令中所有的 blah 都替换为 foo

