

CENTOS 7.X 系统安装及优化

变

启动流程sysvinit

串行启动：一次一个，一个一个启动

并行启动：全部的一起启动

init优点

运行非常良好。主要依赖于shell脚本

init缺点

- 1、启动慢
- 2、容易夯住，fstab与nfs挂载问题

1. upstart 技术

使用的版本：

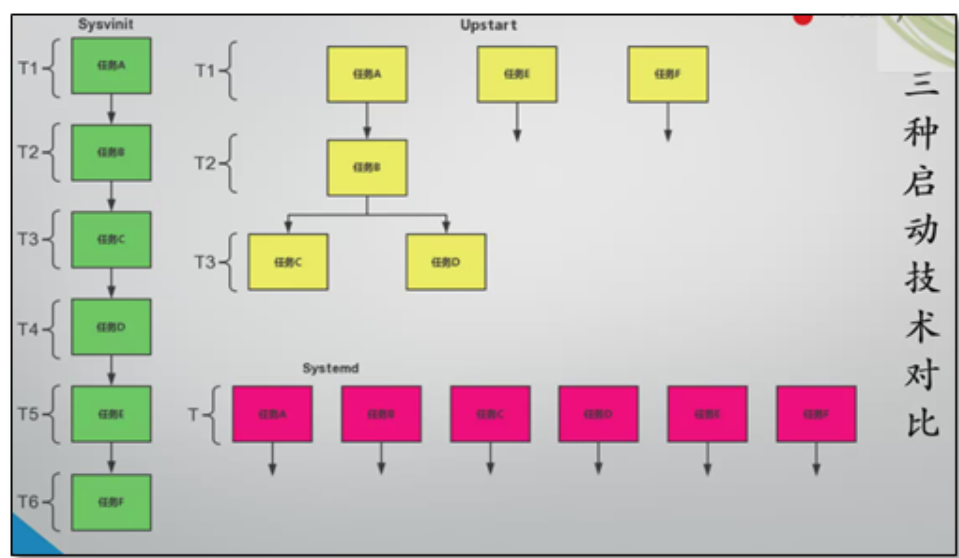
- centos6
- ubuntu14

1. systemd技术

克服init固有缺点，提高系统的启动速度

降低迁移成本

1. 三种启动技术对比



1. 并行启动三大原理：

- 1、解决socket 依赖/端口依赖

socket 网络套接字文件

2、解决D-Bus

依赖：采用了D-Bus 为程序之间的通讯工具，类似消息队列，可以缓存信息

3、解决文件系统依赖、类似autofs机制

1. 安装centos7系统

1. 虚拟机选择

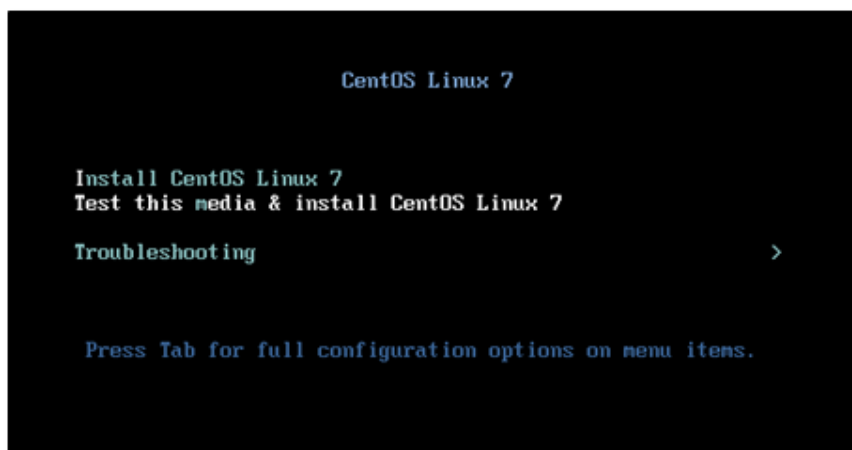


磁盘空间尽量大一些

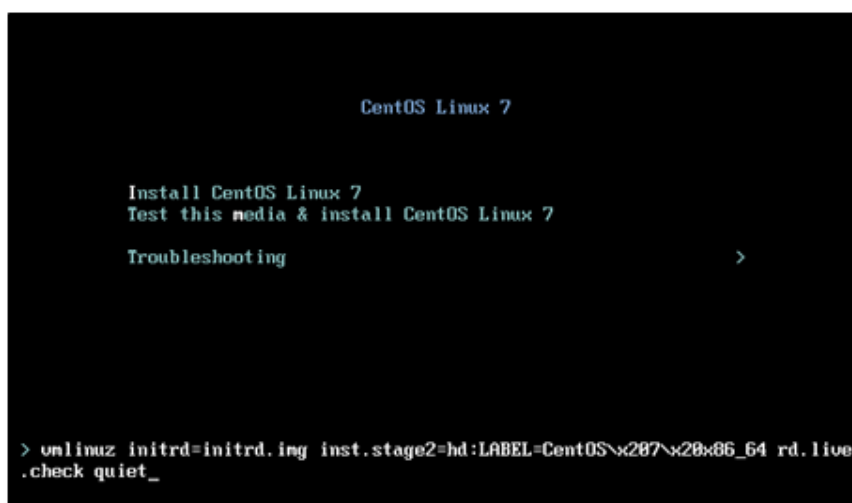


## 1. 系统安装

### 1) 开机进入安装



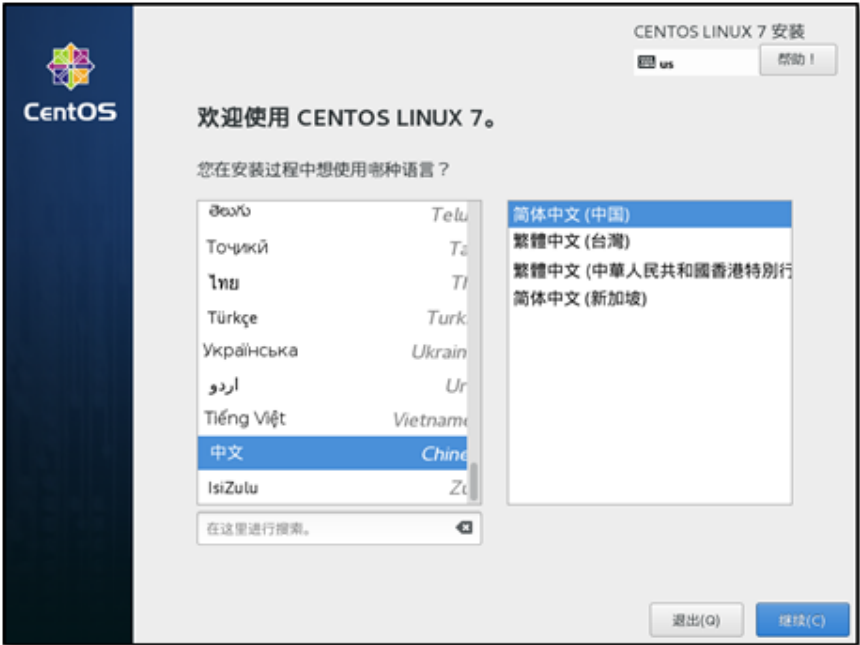
### 2) 按下TAB键



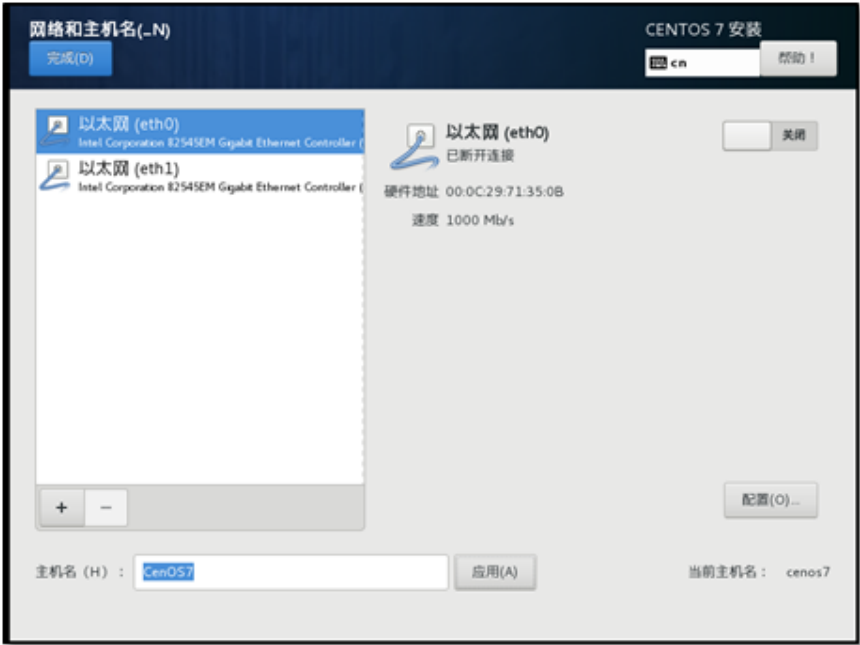
添加内核参数 让网卡名称变为eth

空格 输入 net.ifnames=0 biosdevname=0 然后回车

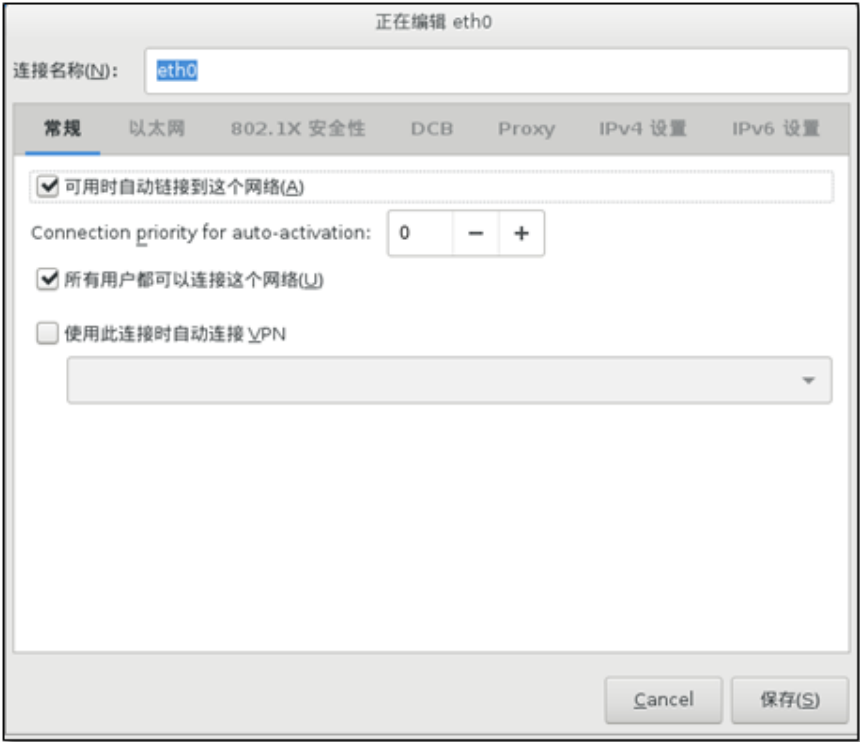
### 3) 选择语言



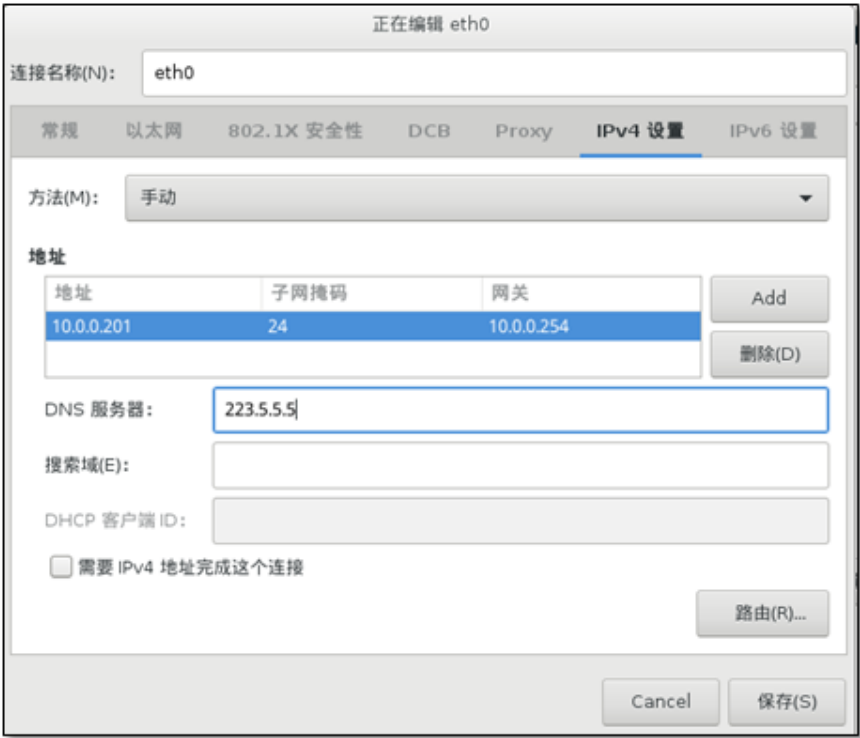
4) 设置网络连接和主机名



5)注意勾选开机自启动

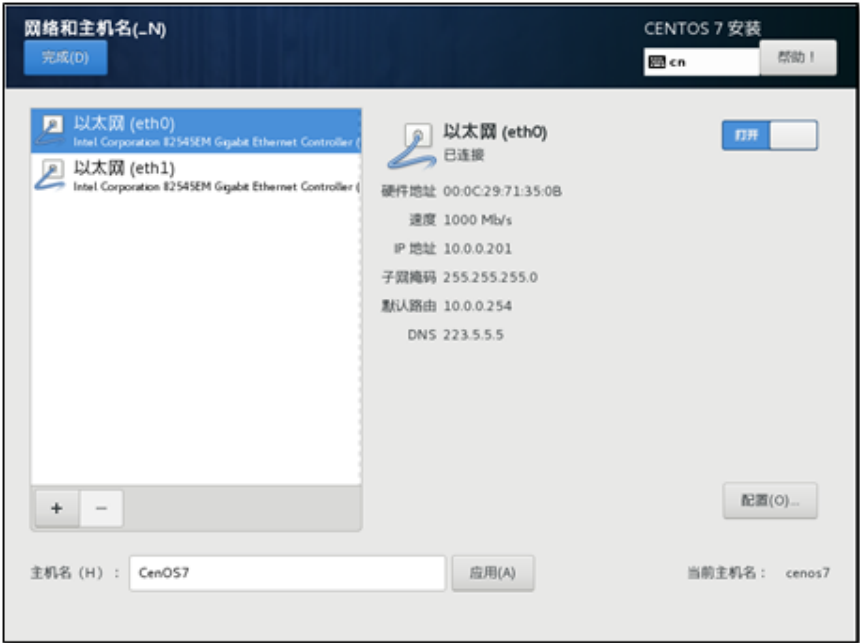


6) 设置ip地址



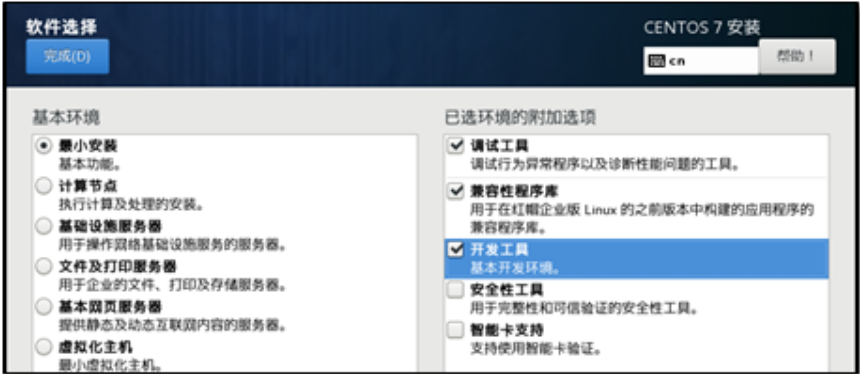
同样的方式将另外 一块网卡配置好

7) 修改主机名



8)选择安装的环境，选择最小化安装。

注意勾选依赖包



9) 磁盘分区为自定义分区。



选择/boot挂载点的大小。这里设置为1g



添加swap挂载点



注意选择都为标准分区类型，直接写到磁盘



添加根分区后选择完成

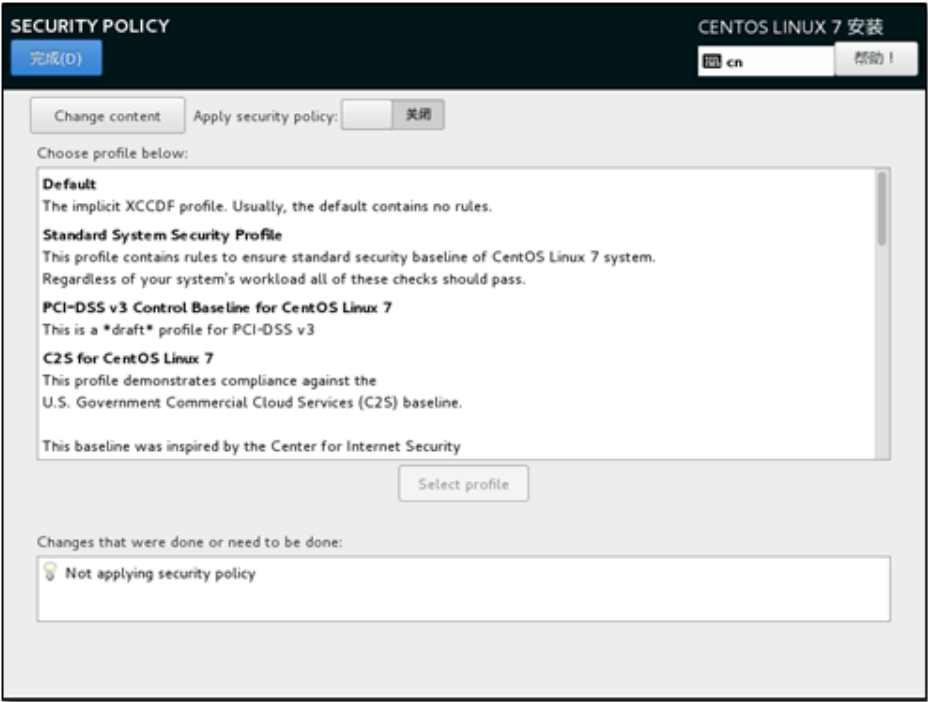


关闭密码保护



选择关闭





关闭KDUMP功能



开始安装，注意为469个包



安装的过程中设置root的密码。

## 1. 安装完成后的优化

查看系统版本

```
[root@CentOS7 ~]# cat /etc/redhat-release
CentOS Linux release 7.3.1611 (Core)
```

## 修改yum源, base源

```
[root@CentOS7 ~]# curl -o /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100  2573  100  2573    0     0   2940      0 --:--:-- --:--:-- --:--:--  2940
```

## epel源

```
[root@CentOS7 ~]# curl -o /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100  1084  100  1084    0     0    915      0 0:00:01 0:00:01 --:--:--  916
```

## 安装软件

```
yum install vim wget bash-completion lrzsz nmap nc tree htop iftop net-tools -y
```

## bash-completion 补全命令参数的包

### 1. centos7 与centos 6的区别

#### 1. 解决更换网卡后名称变更的问题

## 更改名称的方法

修改CentOS7网卡名称为传统名称eth0格式 <http://oldboy.blog.51cto.com/2561410/1722101>

#### 1. nmtui

## centos7 上管理网路的图形化工具

#### 1. 主机名配置文件的区别

## 临时 hostname

## 永久 /etc/hostname

```
[root@CentOS7 ~]# cat /etc/hostname
CentOS7
```

## 一步到位

```
[root@CentOS7 ~]# hostnamectl
Static hostname: CentOS7
    Icon name: computer-vm
    Chassis: vm
    Machine ID: 4ab1c20b25f64a38a7197b8453b04b2c
    Boot ID: abd0a393540b4788bc1e571b413b33c4
    Virtualization: vmware
    Operating System: CentOS Linux 7 (Core)
    CPE OS Name: cpe:/o:centos:centos:7
```

```
Kernel: Linux 3.10.0-514.el7.x86_64  
Architecture: x86-64
```

## 修改主机名

```
[root@CentOS7 ~]# hostnamectl set-hostname clsn
```

## 1. 字符集

### 临时

```
[root@clsn ~]# echo $LANG  
zh_CN.UTF-
```

### 永久

```
[root@clsn ~]# cat /etc/locale.conf  
LANG="zh_CN.UTF-8"
```

### 一步到位

```
[root@clsn ~]# localectl  
System Locale: LANG=zh_CN.UTF-8  
VC Keymap: cn  
X11 Layout: cn
```

## 1. 查看系统版本号

```
[root@clsn bin]# cat /etc/os-release  
NAME="CentOS Linux"  
VERSION="7 (Core)"  
ID="centos"  
ID_LIKE="rhel fedora"  
VERSION_ID="7"  
PRETTY_NAME="CentOS Linux 7 (Core)"  
ANSI_COLOR="0;31"  
CPE_NAME="cpe:/o:centos:centos:7"  
HOME_URL="https://www.centos.org/"  
BUG_REPORT_URL="https://bugs.centos.org/"  
  
CENTOS_MANTISBT_PROJECT="CentOS-7"  
CENTOS_MANTISBT_PROJECT_VERSION="7"  
REDHAT_SUPPORT_PRODUCT="centos"  
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

## 1. 兼容的rc.local

```
cat /etc/rc.local  
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure  
# that this script will be executed during boot.
```

想要使用rc.local要给他可执行权限

```
[root@clsn bin]# chmod +x /etc/rc.d/rc.local
[root@clsn bin]# ll /etc/rc.d/rc.local
-rwxr-xr-x. 1 root root 473 11月 7 2016 /etc/rc.d/rc.local
```

## 1. 运行级别Runlevel

之前的查看运行级别的方法依旧使用

```
[root@clsn init.d]# runlevel
N 3
```

原运行级别文件

```
[root@clsn ~]# ls -lh /usr/lib/systemd/system/runlevel*.target
lrwxrwxrwx. 1 root root 15 11月 13 11:16 /usr/lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx. 1 root root 13 11月 13 11:16 /usr/lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx. 1 root root 17 11月 13 11:16 /usr/lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 11月 13 11:16 /usr/lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 11月 13 11:16 /usr/lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx. 1 root root 16 11月 13 11:16 /usr/lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx. 1 root root 13 11月 13 11:16 /usr/lib/systemd/system/runlevel6.target -> reboot.target
```

7中设置运行级别

```
root@clsn init.d]# systemctl get-default multi-user.target
systemctl get-default #查看运行级别
```

示例

```
[root@clsn ~]# systemctl get-default
multi-user.target
[root@clsn ~]# systemctl set-default multi-user.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/multi-user.target.
```

运行级别的变更

```
[root@clsn ~]# ls -lh /usr/lib/systemd/system/runlevel*.target
lrwxrwxrwx. 1 root root 15 11月 13 11:16 /usr/lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx. 1 root root 13 11月 13 11:16 /usr/lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx. 1 root root 17 11月 13 11:16 /usr/lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 11月 13 11:16 /usr/lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 11月 13 11:16 /usr/lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx. 1 root root 16 11月 13 11:16 /usr/lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx. 1 root root 13 11月 13 11:16 /usr/lib/systemd/system/runlevel6.target -> reboot.target
```

## 1. systemd一统天下

## 系统启动文件的目录

```
[root@clsn init.d]# ls /usr/lib/systemd/system
```

## 启动命令示例

```
[root@clsn init.d]# cat /usr/lib/systemd/system/crond.service
[Unit]
Description=Command Scheduler
After=auditd.service systemd-user-sessions.service time-sync.target

[Service]
EnvironmentFile=/etc/sysconfig/crond
ExecStart=/usr/sbin/crond -n $CRONDARGS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process

[Install]
WantedBy=multi-user.target
```

语法格式 <http://www.jinbuguo.com/systemd/systemd.service.html>

## 1. 管理服务

### 关闭防火墙服务及开机自启动

```
systemctl stop postfix.service
systemctl disable postfix.service
```

### 关闭防火墙详细过程

```
[root@clsn ~]# systemctl disable firewalld.service
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@clsn ~]# systemctl stop firewalld.service
[root@clsn ~]# systemctl status firewalld.service
```

systemctl:融合service和chkconfig的功能于一体,兼容SysV和LSB的启动脚本, 而且能够在进程启动过程中更有效地引导加载服务。。

## 1. 命令对比

Sysvinit 命令	Systemd 命令	备注
<code>service crond start</code>	<code>systemctl start crond.service</code>	启动服务。
<code>service crond stop</code>	<code>systemctl stop crond.service</code>	停止服务。
<code>service crond restart</code>	<code>systemctl restart crond.service</code>	重启服务。
<code>service crond reload</code>	<code>systemctl reload crond.service</code>	当支持时，重新装载配置文件而不中断等待操作。
<code>service crond condrestart</code>	<code>systemctl condrestart crond.service</code>	如果服务正在运行那么重启它。
<code>service crond status</code>	<code>systemctl status crond.service</code>	查看服务运行状态。
<code>chkconfig --list</code>	<code>systemctl list-unit-files</code>	输出在各个运行级别下服务的启用和禁用情况。
<code>chkconfig crond on</code>	<code>systemctl enable crond.service</code>	在下次启动时或满足其他触发条件时设置服务为启用。
<code>chkconfig crond off</code>	<code>systemctl disable crond.service</code>	在下次启动时或满足其他触发条件时设置服务为禁用。
<code>chkconfig --list crond</code>	<code>systemctl is-enabled crond.service</code>	用来检查一个服务在当前环境下被配置为启用还是禁用。
<code>chkconfig --add crond</code>	<code>systemctl daemon-reload</code>	当您创建新服务文件或者变更设置时使用。

`systemctl start crond.service`和`systemctl start crond`效果一样。

1. 开机时间优化

```
systemd-analyze time
systemd-analyze blame
systemd-analyze plot >boottime.avg
```

开机启动时间

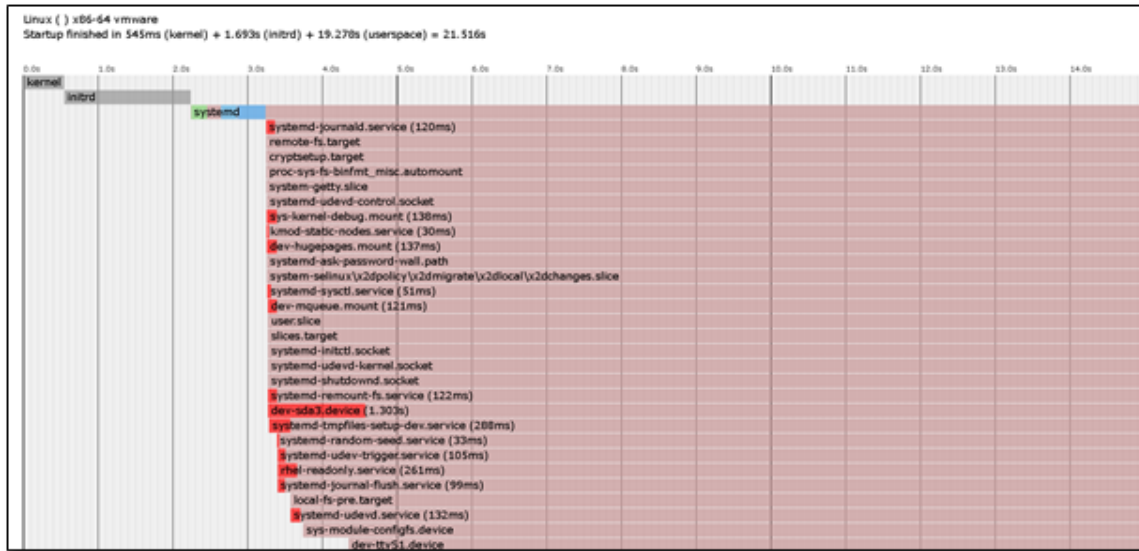
```
[root@clsn system]# systemd-analyze time
Startup finished in 387ms (kernel) + 1.406s (initrd) + 31.831s (userspace) = 33.626s
```

查看具体的使用时间

```
[root@clsn system]# systemd-analyze blame
10.812s sshd-keygen.service
10.758s abrt-ccpp.service
6.395s chronyd.service
.....
```

生产图表

```
[root@clsn tmp]# systemd-analyze plot >boottime.svg
[root@clsn tmp]# sz boottime.svg
```



## 1. 开机启动项优化

```
[root@kickstart ~]# systemctl list-unit-files | egrep "^ab|^aud|^kdump|vm|^md|^mic|^post|lvm"
Removed symlink /etc/systemd/system/multi-user.target.wants/abrt-ccpp.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/abrt-oops.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/abrt-vmcore.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/abrt-xorg.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/abrttd.service.
Removed symlink /etc/systemd/system/basic.target.wants/auditd.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/vmtoolsd.service.
Removed symlink /etc/systemd/system/vmtoolsd.service.requires/vgauthd.service.
```

## 1. OpenStack部署优化环境

### 1. 修改网卡配置文件

# 第一个里程碑：重启完成之后，登录系统 #

```
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eth0
UUID=4bf2af89-5f6d-4979-83e9-df6ce1c3cc41
DEVICE=eth0
ONBOOT=no
```

# 第二个里程碑：删掉多余的行，只留以下内容 #

```
TYPE=Ethernet
BOOTPROTO=dhcp
NAME=eth0
DEVICE=eth0
ONBOOT=no
```

# 第三个里程碑：在以上基础上，继续配置 #

# 红色为修改和增加部分

# 以下代码为优化后的网卡全部配置 #

```
TYPE=Ethernet
BOOTPROTO=none          #取消dhcp
NAME=eth0
DEVICE=eth0
ONBOOT=yes              #设置开机启动
IPADDR=10.0.0.11         #IP地址
NETMASK=255.255.255.0    #子网掩码
GATEWAY=10.0.0.254       #网关
DNS1=223.5.5.5           #DNS
```

# 说明：与/etc/init.d/network restart 效果一致

# 第五个里程碑：检查 ping 一下网关10.0.0.254测试网络连通性 #

```
ping 10.0.0.254 -c2
```

```
[root@localhost ~]# ping 10.0.0.254 -c4
PING 10.0.0.254 (10.0.0.254) 56(84) bytes of data.
64 bytes from 10.0.0.254: icmp_seq=1 ttl=128 time=0.095 ms
64 bytes from 10.0.0.254: icmp_seq=2 ttl=128 time=0.103 ms
64 bytes from 10.0.0.254: icmp_seq=3 ttl=128 time=0.102 ms
64 bytes from 10.0.0.254: icmp_seq=4 ttl=128 time=0.110 ms

--- 10.0.0.254 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.095/0.102/0.110/0.011 ms
[root@localhost ~]#
```

到这里，ok，可以愉快的使用xshell了

## 1. 系统的优化

# 登录xshell开始优化 #

### 1. 防火墙的优化

```
systemctl disable firewalld.service
systemctl stop firewalld
```

### 1. Selinux的优化

```
sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/selinux/config
```



## 1. ssh的优化

# 修改完成后为红色部分 #

```
vi /etc/ssh/sshd_config
```

93行: **GSSAPIAuthentication no**

129行: **UseDNS no**

```
sed -i
```

```
' 93s@GSSAPIAuthentication yes@GSSAPIAuthentication no@;129s@#UseDNS yes@UseDNS no@' /etc/ssh/sshd_config
```

# 重启ssh #

```
systemctl restart sshd
```

## 1. hosts的优化

```
vi /etc/hosts
# 增加2行
10.0.0.11    controller
10.0.0.31    compute1
10.0.0.32    compute2
```

## 1. 修改主机名

```
hostnamectl set-hostname controller
```

## 1. yum源优化

# 使用光盘搭建本地yum源

```
umount /mnt
cd /etc/yum.repos.d/
mkdir test -p
\mv *.repo test
echo '[local]'
name=local
baseurl=file:///mnt
gpgcheck=0' >local.repo
mount /dev/cdrom /mnt
yum makecache
```

## 1. 其他优化

```
# 关闭网卡图形化设置模式 #
systemctl stop NetworkManager.service
systemctl disable NetworkManager.service
# 关闭邮件服务
systemctl stop postfix.service
systemctl disable postfix.service
# 下载tab补全命令 #
yum install -y bash-completion.noarch
# 下载 常用命令 #
yum install -y net-tools vim lrzsz wget tree screen lsof tcpdump
```

```
# 至此：模板机优化完成；关机开始克隆 #  
shutdown -h now
```

## 1. 参考文献

[https://access.redhat.com/documentation/zh-CN/Red\\_Hat\\_Enterprise\\_Linux/7/index.html](https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/7/index.html)

赞0

如无特殊说明，文章均为本站原创，转载请注明出处

- 转载请注明来源：CentOS 7.X 系统安装及优化
- 本文永久链接地址：<https://www.nmtui.com/clsn/lx321.html>

该文章由 惨绿少年 发布



惨绿少年Linux [www.nmtui.com](http://www.nmtui.com)