

定制 CentOS Live OS

Beck@2017.2.4

发行版的 Linux 一般会推出 Live CD 供用户体验。Live CD 类似于 WinPE, 无需安装, 直接从光盘或者 U 盘启动即可使用, 比较适合于完成一些比较简单的任务。

但是 Live CD 的文件系统一般是 squashfs 的只读格式, 大多数文件修改, 如在系统运行时安装的软件和设定在重新启动后无法保存。因此需要定制 LiveOS, 以满足实际的使用需求。

下面以 CentOS6.8 Live CD 官方发行版为基础, 介绍一下如何修改 Live CD.

1. Live CD 文件结构

____EFI	EFI 启动选项
__boot	
__isolinux	Legacy 启动选项
__boot.cat	启动 catalog
__initrd0.img	启动时加载的根文件系统
__isolinux.bin	启动引导文件
__isolinux.cfg	启动菜单配置文件
__memtest	memtest86 测试软件
__splash.jpg	启动界面背景图片
__vesamenu.c32	启动菜单程序
__vmlinuz0	启动时加载的 kernel
__LiveOS	
__livecd-iso-to-disk	安装 livecd 到硬盘
__osmin.img	最小 OS
__squashfs.img	启动后加载的文件系统

需要修改的文件主要有 squashfs.img 和 isolinux.cfg

2. 操作环境

操作系统：完整安装的 CentOS6.8 (与 Live CD 版本一致，为了保证 kernel 版本相同)

必要软件：squashfs-tools

用 yum 安装 yum install squashfs-tools

3. 解包 LiveCD

3.1 将下载好的 CentOS6.8 liveCD iso 镜像文件拷贝到完整安装的 CentOS6.8 任意路径下，这里放在根目录下。

```
[root@localhost ~]# cd /
[root@localhost /]# ls
bin    CentOS-6.8-x86_64-LiveCD.iso  dev  home  lib64  media  mnt  opt  root  selinux  sys  usr
boot  cgroup                        etc  lib   lost+found  misc  net  proc  sbin  srv    tmp  var
[root@localhost /]# _
```

3.2 将 iso 镜像挂载到/media

```
[root@localhost /]# mount -o loop -t iso9660 CentOS-6.8-x86_64-LiveCD.iso /media
```

3.3 新建一个路径 /livecd，并将 iso 镜像内的文件拷贝到这个路径下

```
[root@localhost /]# mkdir /livecd
[root@localhost /]# cp -rf /media/* /livecd/
[root@localhost /]# ls /livecd
EFI  GPL  isolinux  LiveOS
```

3.4 进入 /livecd/LiveOS,使用 unsquashfs 解压缩 squashfs.img,会生成一个 squashfs-root 的文件夹,

```
[root@localhost LiveOS]# ls
livecd-iso-to-disk osmin.img squashfs.img
[root@localhost LiveOS]# unsquashfs squashfs.img
Parallel unsquashfs: Using 1 processor
1 inodes (32768 blocks) to write

=====
created 1 files
created 2 directories
created 0 symlinks
created 0 devices
created 0 fifos
[root@localhost LiveOS]# ls
livecd-iso-to-disk osmin.img squashfs.img squashfs-root
```

3.5 squashfs-root 文件夹内含有一个 LiveOS 文件夹, 其下包含一个 ext3fs.img 的镜像文件, 该镜像即为 LiveOS 启动后真正要挂载的 ext3 根文件系统, 我们对 LiveOS 的修改也主要是修改这个 img 内的内容。

```
[root@localhost ~]# cd /livecd/LiveOS/squashfs-root/LiveOS
[root@localhost LiveOS]# ls
ext3fs.img
```

3.6 创建一个/ext3fs 文件夹, 并以 rw 可写模式将 ext3fs.img 挂载到这个文件夹

```
[root@localhost /]# mkdir /ext3fs
[root@localhost /]# mount -o loop,rw /livecd/LiveOS/squashfs-root/LiveOS/ext3fs.img /ext3fs
```

3.7 用 chroot 将根目录切换到/ext3fs, 此后的 shell 命令都将以/ext3fs 这个位置作为根目录

```
[root@localhost /]# chroot /ext3fs/
[root@localhost /]# ls
bin  dev  home  lib64  media  opt  root  selinux  sys  usr
boot  etc  lib  lost+found  mnt  proc  sbin  srv  tmp  var
```

4 在 /ext3fs 内做修改

此处假设修改的目的为使 LiveCD 以命令行模式启动，启动后能以 root 自动登录 shell，无需输入密码。进入 shell 后能自动执行脚本。且需要安装某些必要的系统监控软件，如 ipmitool 和 mcelog 等。接下来将逐步介绍怎样进行修改。

4.1 修改启动级别，使 LiveCD 以命令行启动

将/etc/inittab 中最后一行 id:5:initdefault: 中的 5 改为 3，使用 init 3 级别启动。

```
# inittab is only used by upstart for the default runlevel.
#
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# System initialization is started by /etc/init/rcS.conf
#
# Individual runlevels are started by /etc/init/rc.conf
#
# Ctrl-Alt-Delete is handled by /etc/init/control-alt-delete.conf
#
# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/serial.conf,
# with configuration in /etc/sysconfig/init.
#
# For information on how to write upstart event handlers, or how
# upstart works, see init(5), init(8), and initctl(8).
#
# Default runlevel. The runlevels used are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
~
~
~
~
~
~
~
~
~
~
~
~
```

4.2 修改以 root 免密码自动登录

修改 /etc/init/tty.conf 中 exec /sbin/mingetty \$TTY 这一行为

```
exec /sbin/mingetty -autologin=root $TTY
```

修改前

```
# tty - getty
#
# This service maintains a getty on the specified device.
#
# Do not edit this file directly. If you want to change the behaviour,
# please create a file tty.override and put your changes there.

stop on runlevel [S016]

respawn
instance $TTY
exec /sbin/mingetty $TTY
usage 'tty TTY=/dev/ttyX - where X is console id'
~
```

修改后

```
# tty - getty
#
# This service maintains a getty on the specified device.
#
# Do not edit this file directly. If you want to change the behaviour,
# please create a file tty.override and put your changes there.

stop on runlevel [S016]

respawn
instance $TTY
exec /sbin/mingetty --autologin=root_$TTY
usage 'tty TTY=/dev/ttyX - where X is console id'
~
~
~
~
```

4.3 设定进入 shell 后自动执行某个脚本

修改 /root/.bash_profile,增加一行 `bash /mnt/live/scripts/MainShell`, 其中 /mnt/live 对应的位置为 LiveCD 烧录到 U 盘后, 在 Windows 下看到的 U 盘根目录。所以添加这条语句后, 需要在 Windows 下在 U 盘根目录下创建 scripts 文件夹, 并将编辑好的脚本文件 MainShell 放到这个文件夹内。

修改前

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH

~
```

修改后

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH

bash /mnt/live/scripts/MainShell_
~
~
```

4.4 安装和配置 ipmitool

利用指令 `yum -y install ipmitool` 来联网安装 ipmitool, 安装完成后需要将加载 ipmi 驱动模块的指令添加到开机设定档/etc/rc.d/rc.local 中, 实现开机后自动加载 ipmi 驱动

```
[root@localhost ~]# echo "insmod /lib/modules/2.6.32-642.el6.x86_64/kernel/drivers/char/ipmi/ipmi_msghandler.ko" >>/etc/rc.d/rc.local
[root@localhost ~]# echo "insmod /lib/modules/2.6.32-642.el6.x86_64/kernel/drivers/char/ipmi/ipmi_devintf.ko" >>/etc/rc.d/rc.local
[root@localhost ~]# echo "insmod /lib/modules/2.6.32-642.el6.x86_64/kernel/drivers/char/ipmi/ipmi_si.ko" >>/etc/rc.d/rc.local
```

4.5 安装和配置 mcelog

利用指令 `yum -y install mcelog` 来联网安装 mcelog 工具。安装完成后需要将 mcelog 以 daemon 方式运行的设定写入/etc/rc.d/rc.local 中

```
# echo "mcelog --daemon" >>/etc/rc.d/rc.local
```

5 重新打包 squashfs.img

5.1 完成在/ext3fs 内的修改后，按 ctrl+d 退出 chroot，重新回到系统的根目录，并 umount /ext3fs

5.2 重新打包生成 squashfs.img

进入 /livecd/LiveOS, 将原始的 squashfs.img 移动到其他位置，这里以/tmp 为例。使用 mksquashfs 命令将修改过的包含 ext3fs.img 的文件夹 squashfs-root 重新打包为新的 squashfs.img。

打包完成后同样使用 mv 将 squashfs-root 移动到/tmp. 此时/livecd/LiveOS 下应该只包含 livecd-iso-to-disk, osmin.img 和修改过的 squashfs.img 三个文件。

```
[root@localhost livecd]# cd /livecd/LiveOS/
[root@localhost LiveOS]# ls
livecd-iso-to-disk  osmin.img  squashfs.img  squashfs-root
[root@localhost LiveOS]# mv squashfs.img /tmp
[root@localhost LiveOS]# mksquashfs squashfs-root squashfs.img
```

5.3 修改启动菜单文件 isolinux.cfg

进入/livecd/isolinux 文件夹，用 chmod +w isolinux.cfg 为 isolinux.cfg 增加写入权限。

编辑 isolinux.cfg 文件，建议将 timeout 修改为 30（3 秒）以减少启动画面等待时间。并记下 CDLABEL=CentOS-6.8-x86_64-LiveCD 这个 cd 标签，如果修改这个标签，后续的打包 ISO 指令里的标签必须与修改后的保持一致（区分大小写），如果这个 label 不一致会导致无法正常启动。

```
[root@localhost isolinux]# cd /
[root@localhost /]# cd livecd/
[root@localhost livecd]# ls
EFI  GPL  isolinux  LiveOS
[root@localhost livecd]# cd isolinux/
[root@localhost isolinux]# ls
boot.cat  initrd0.img  isolinux.bin  isolinux.cfg  memtest  splash.jpg  vesamenu.c32  vmlinuz0
[root@localhost isolinux]# chmod +w isolinux.cfg
[root@localhost isolinux]# vi isolinux.cfg
```

```

default vesamenu.c32
timeout 30

menu background splash.jpg
menu title Welcome to CentOS-6.8-x86_64-LiveCD!
menu color border 0 #ffffff #00000000
menu color sel 7 #ffffff #ff000000
menu color title 0 #ffffff #00000000
menu color tabmsg 0 #ffffff #00000000
menu color unsel 0 #ffffff #00000000
menu color hotsel 0 #ff000000 #ffffff
menu color hotkey 7 #ffffff #ff000000
menu color timeout_msg 0 #ffffff #00000000
menu color timeout 0 #ffffff #00000000
menu color cmdline 0 #ffffff #00000000
menu hidden
menu hiddenrow 5
label rd.live.check0
    menu label Verify and Boot
    kernel vmlinuz0
    append initrd=initrd0.img root=live:CDLABEL=CentOS-6.8-x86_64-LiveCD rootfstype=auto ro liveimg qu
    iet nodiskmount nolvmount rhgb rd.luks=0 rd.md=0 rd.dm=0 rd.live.check
label linux0
    menu label Boot
    kernel vmlinuz0
    append initrd=initrd0.img root=live:CDLABEL=CentOS-6.8-x86_64-LiveCD rootfstype=auto ro liveimg qu
    iet nodiskmount nolvmount rhgb rd.luks=0 rd.md=0 rd.dm=0
menu default
label linuxtext0
    menu label Boot (Text Mode)
    kernel vmlinuz0
    append initrd=initrd0.img root=live:CDLABEL=CentOS-6.8-x86_64-LiveCD rootfstype=auto ro liveimg 3
    quiet nodiskmount nolvmount rhgb rd.luks=0 rd.md=0 rd.dm=0
label linux0
    menu label Boot (Basic Video)
-- INSERT --

```

6 打包 iso 镜像

进入到/livecd 路径，在该路径下使用 mkisofs 指令打包制作新的 LiveCD iso 镜像文件。
注意-V 参数后的 CD label 必须与 isolinux.cfg 中的 label 一致。

```
mkisofs -R -b isolinux/isolinux.bin -c isolinux/boot.cat -o /centos68_new.iso -V
"CentOS-6.8-x86_64-LiveCD" -no-emul-boot -boot-load-size 4 -boot-info-
table . ( 注意最后一个. 代表当前目录，不可省略 )
```

执行成功后会在根目录下生成一个名为 centos68_new.iso 的 LiveCD 新镜像文件。将该
镜像烧录到 U 盘即可作为 Linux LiveCD 启动盘使用

```

[root@localhost ~]# cd /livecd/
[root@localhost livecd]# ls
EFI GPL isolinux LiveOS
[root@localhost livecd]# mkisofs -R -b isolinux/isolinux.bin -c isolinux/boot.cat -o /centos68_new.i
so -V "CentOS-6.8-x86_64-LiveCD" -no-emul-boot -boot-load-size 4 -boot-info-table .

```


7 创建开机自动执行脚本

需要根据自己的需求撰写 MainShell 脚本文件，并在 Windows 下将该脚本放在 U 盘根目录下的 scripts 文件夹内。

scripts/MainShell 为 4.3 步骤自定义的文件路径，可在 4.3 步骤时自行定义路径名或脚本文件名放在/mnt/live 路径下。

后续其他脚本的调用可以通过修改 MainShell 方便的调整。