

DOCUMENTATION

make_debian64.sh

Step 1.

Checking whether current iso creator using uid 0 or not:

```
if [ "$(id -u)" != "0" ]; then
    echo "This script must be run as root" 1>&2
    exit 1
fi
```

so other than root can not run this script

Step 2.

Installing debootstrap and requirements for host machine

```
apt-get install -y debootstrap syslinux isolinux squashfs-tools genisoimage rsync makepasswd
mkdir $HOME/live_boot
debootstrap --arch=amd64 --variant=minbase jessie $HOME/live_boot/chroot
http://ftp.us.debian.org/debian/
```

Step 3.

Preparing default password : debian for root user

```
default_password=debian
pass=`makepasswd --clearfrom=- --crypt-md5 <<< $default_password | cut -b 10-100`
sed -i -e "s,^root:[^:]\+:,root:$pass:," $HOME/live_boot/chroot/etc/shadow
```

Step 4.

Adding fzf, if fzf directory not exist at /root

```
#fzf addition
echo "fzf='fzf'" >> $HOME/live_boot/chroot/root/.bashrc
echo 'if [ -d "$fzf" ]; then' >> $HOME/live_boot/chroot/root/.bashrc
echo "echo 'fzf ready'" >> $HOME/live_boot/chroot/root/.bashrc
echo "else" >> $HOME/live_boot/chroot/root/.bashrc
echo "cd /root;git clone https://github.com/junegunn/fzf.git" >>
$HOME/live_boot/chroot/root/.bashrc
echo "fi" >> $HOME/live_boot/chroot/root/.bashrc
#eof fzf addition
```

Step 5.

Preparing autologin and auto startx at chroot environment

```
#auto login and startx
mkdir -p $HOME/live_boot/chroot/etc/systemd/system/getty@tty1.service.d
echo "[Service]" >
$HOME/live_boot/chroot/etc/systemd/system/getty@tty1.service.d/override.conf
echo "ExecStart=" >>
$HOME/live_boot/chroot/etc/systemd/system/getty@tty1.service.d/override.conf
echo "ExecStart=-/sbin/agetty --noissue --autologin myusername %I $TERM" >>
$HOME/live_boot/chroot/etc/systemd/system/getty@tty1.service.d/override.conf
echo "Type=idle" >>
$HOME/live_boot/chroot/etc/systemd/system/getty@tty1.service.d/override.conf
echo "[ -z $DISPLAY ] && exec startx" >> $HOME/live_boot/chroot/root/.bashrc
```

Step 6.

Reading file install.list in current directory which contains packages to install

```
#reading install.list
filename="install.list"
while read -r line
do
    packages+=" $line"
done < "$filename"

#linux-image-3.16.0-4-amd64
```

Step 7.

Creating new chroot environment to prepare guest host and installing packages which has been prepared at install.list and installing linux image

```
chroot $HOME/live_boot/chroot /bin/bash -c "uname -a; \
sleep 3; \
echo debian-live-amd64 > /etc/hostname; \
apt-get update; \
apt-get install --yes --force-yes $packages;apt-get install --yes --force-yes linux-image-3.16.0-4-amd64; apt-get clean"
```

Step 8.

Once we get out from chroot environment, preparing isolinux, vmlinuz and initrd for our guest host

```
mkdir -p $HOME/live_boot/image/{live,isolinux}
```

```
(cd $HOME/live_boot && mksquashfs chroot image/live/filesystem.squashfs -e boot)
```

```
(cd $HOME/live_boot && cp chroot/boot/vmlinuz-3.16.0-4-amd64 image/live/vmlinuz1 && cp  
chroot/boot/initrd.img-3.16.0-4-amd64 image/live/initrd1)
```

```
cat > $HOME/live_boot/image/isolinux/isolinux.cfg <<- EOM  
UI menu.c32  
prompt 0  
menu title Debian Live  
timeout 10  
label Debian Live x86_64  
menu label ^Debian Live _64  
menu default  
kernel /live/vmlinuz1  
append initrd=/live/initrd1 boot=live  
EOM
```

```
cd $HOME/live_boot/image/ && \  
cp /usr/lib/ISOLINUX/isolinux.bin isolinux/ && \  
cp /usr/lib/syslinux/modules/bios/menu.c32 isolinux/ && \  
cp /usr/lib/syslinux/modules/bios/hdt.c32 isolinux/ && \  
cp /usr/lib/syslinux/modules/bios/ldlinux.c32 isolinux/ && \  
cp /usr/lib/syslinux/modules/bios/libutil.c32 isolinux/ && \  
cp /usr/lib/syslinux/modules/bios/libmenu.c32 isolinux/ && \  
cp /usr/lib/syslinux/modules/bios/libcom32.c32 isolinux/ && \  
cp /usr/lib/syslinux/modules/bios/libgpl.c32 isolinux/ && \  
cp /usr/share/misc/pci.ids isolinux
```

Step 9. Final step will be creating the cdrom iso

```
genisoimage -rational-rock -volid "Debian Live" -cache-inodes -joliet -hfs -full-iso9660-  
filenames -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-  
table -output $HOME/live_boot/debian-live-amd64.iso $HOME/live_boot/image
```