

[unix.stackexchange.com](https://unix.stackexchange.com)

## boot - Creating bootable Debian image with debootstrap

5-6 minutes

---

I have been trying to create a bootable debian (jessie/8.4) image for the past 2 days, and as far as I can tell I have the procedure right, but I can not get the filesystem right. I am relatively sure that I am doing something wrong here, missing something with mounting or `/etc/fstab` (*there isn't one in my image*). I was hoping someone with some experience would be able to help me out/show me what I am missing.

Here are the errors I see as I'm booting into qemu-system-x86:

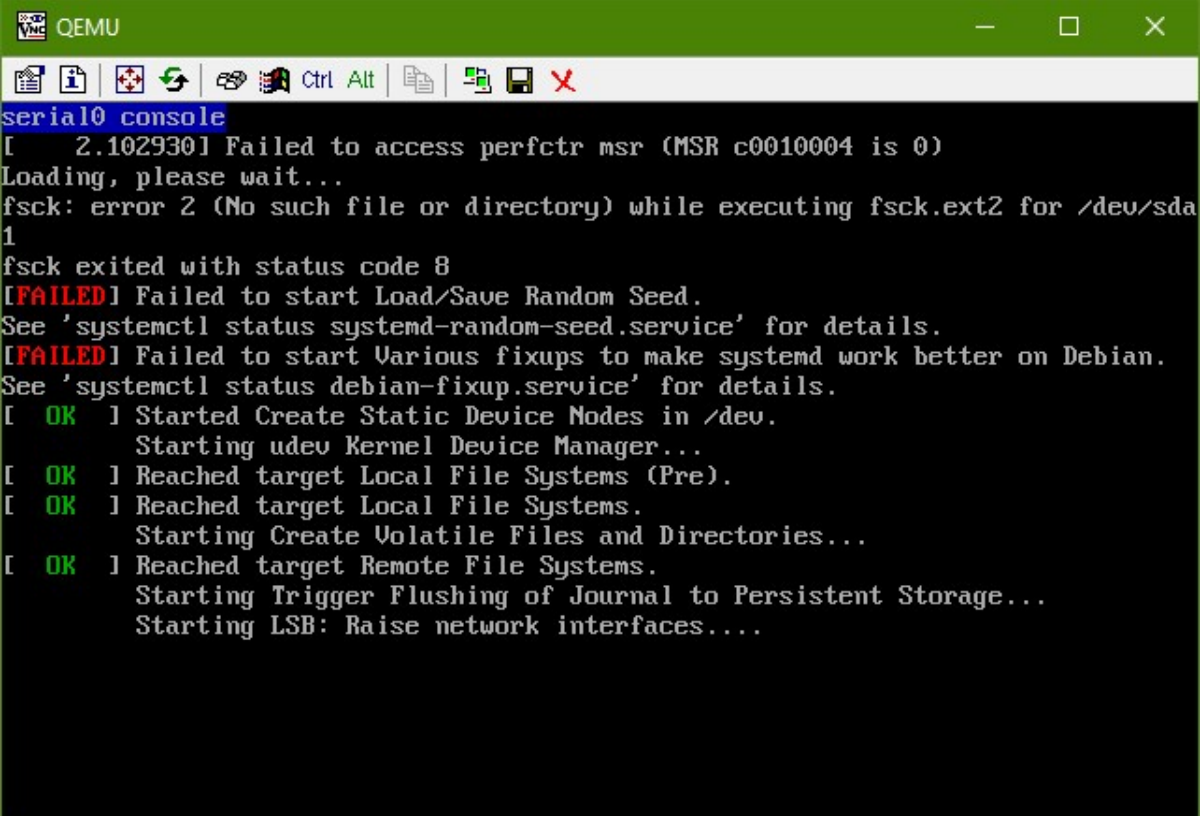
As text and then as the actual screenshots:

```
fsck: error 2 (No such file or directory) while
executing fsck.ext2 for /dev/sda1
fsck exited with status code 8
[FAILED] Failed to start Load/Save Random Seed
See `systemctl status systemd-random-seed.service`
for details.
[FAILED] Failed to start Various fixups to make
systemd work better on Debian.
See `systemctl status debian-fixup.service` for
details.
...
```

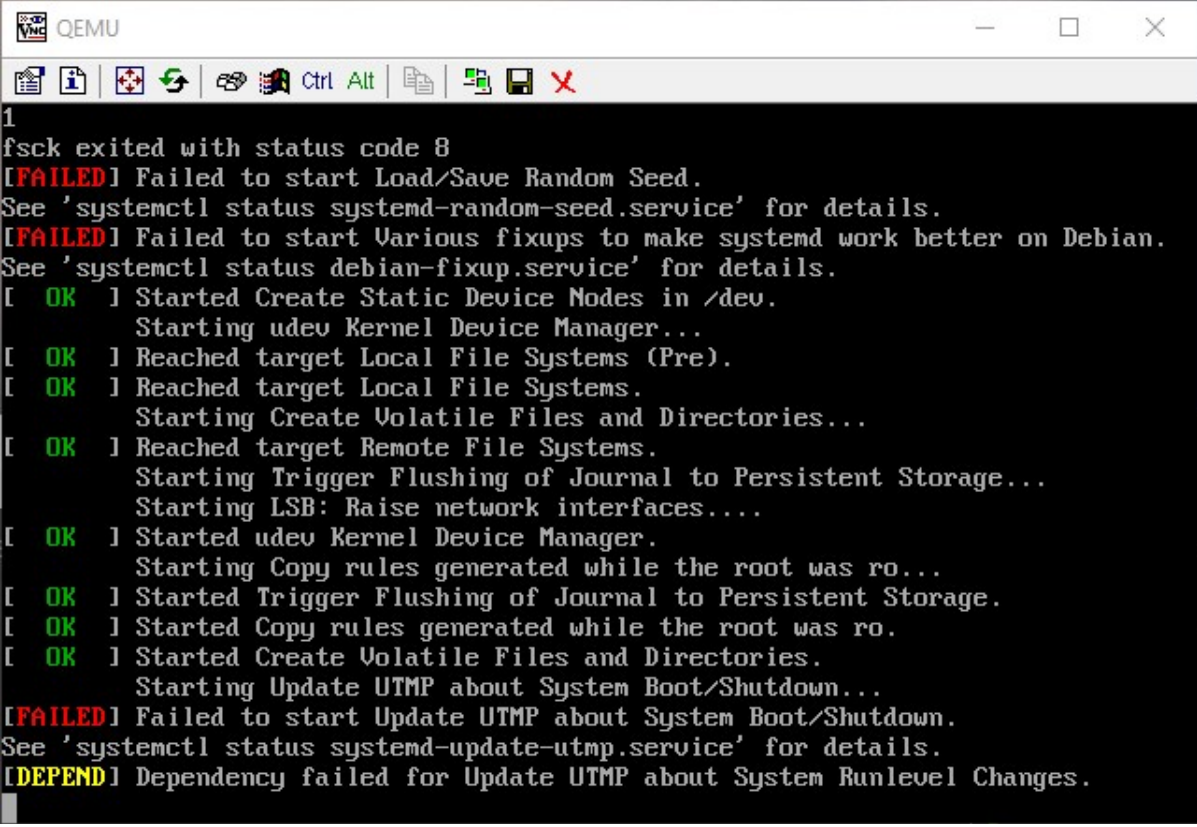
```
[FAILED] Failed to start Update UTMP about System
Boot/Shutdown.
```

```
See `systemctl status systemd-update-utmp.service`
for details.
```

```
[DEPEND] Dependency failed for Update UTMP about
System Runlevel Changes.
```



```
serial0 console
[ 2.102930] Failed to access perfctr msr (MSR c0010004 is 0)
Loading, please wait...
fsck: error 2 (No such file or directory) while executing fsck.ext2 for /dev/sda
1
fsck exited with status code 8
[FAILED] Failed to start Load/Save Random Seed.
See `systemctl status systemd-random-seed.service` for details.
[FAILED] Failed to start Various fixups to make systemd work better on Debian.
See `systemctl status debian-fixup.service` for details.
[ OK ] Started Create Static Device Nodes in /dev.
Starting udev Kernel Device Manager...
[ OK ] Reached target Local File Systems (Pre).
[ OK ] Reached target Local File Systems.
Starting Create Volatile Files and Directories...
[ OK ] Reached target Remote File Systems.
Starting Trigger Flushing of Journal to Persistent Storage...
Starting LSB: Raise network interfaces....
```



```
1
fsck exited with status code 8
[FAILED] Failed to start Load/Save Random Seed.
See 'systemctl status systemd-random-seed.service' for details.
[FAILED] Failed to start Various fixups to make systemd work better on Debian.
See 'systemctl status debian-fixup.service' for details.
[ OK ] Started Create Static Device Nodes in /dev.
Starting udev Kernel Device Manager...
[ OK ] Reached target Local File Systems (Pre).
[ OK ] Reached target Local File Systems.
Starting Create Volatile Files and Directories...
[ OK ] Reached target Remote File Systems.
Starting Trigger Flushing of Journal to Persistent Storage...
Starting LSB: Raise network interfaces...
[ OK ] Started udev Kernel Device Manager.
Starting Copy rules generated while the root was ro...
[ OK ] Started Trigger Flushing of Journal to Persistent Storage.
[ OK ] Started Copy rules generated while the root was ro.
[ OK ] Started Create Volatile Files and Directories.
Starting Update UTMP about System Boot/Shutdown...
[FAILED] Failed to start Update UTMP about System Boot/Shutdown.
See 'systemctl status systemd-update-utmp.service' for details.
[DEPEND] Dependency failed for Update UTMP about System Runlevel Changes.
```

Here are the instructions I've written up for myself / steps I've taken:

```
cd ~
mkdir debootstrap
cd debootstrap/
# get newest
wget http://ftp.debian.org/debian/pool/main
/d/debootstrap/debootstrap_1.0.80_all.deb
ar -x debootstrap_1.0.80_all.deb
zcat /root/debootstrap/data.tar.gz | tar xv

apt-get install parted

# 1.5Gbytes
dd if=/dev/zero of=1445.img bs=1024 count=1
seek=1536k
```

```
parted -s 1445.img -- mklabel msdos mkpart primary
1m 1.5g toggle 1 boot
losetup --show -f 1445.img
# prints out `/dev/loopX`, enter this on the next
lin
partprobe /dev/loop0
# only have to make the filesystem once --> if you
are troubleshooting steps, do not redo this line
mkfs -t ext2 /dev/loop0p1
mount /dev/loop0p1 /mnt

debootstrap --verbose
--components=main,contrib,non-free \
--include=firmware-realtek,linux-image-amd64,grub-
pc,ssh,vim \
--exclude=nano \
--arch amd64 jessie /mnt http://ftp.us.debian.org
/debian
```

### [source for information on using --components](#)

- ***Ensure that the kernel is installed, it should appear in /boot within the chroot, that is /mnt/boot with the following files:***
- `initrd.img-3.16.0-4-amd64`
- `vmlinuz-3.16.0-4-amd64`
- `config-3.16.0-4-amd64`
- `System.map-3.16.0-4-amd64`
- **install grub**

```
grub-install --boot-directory=/mnt/boot  
--modules=part_msdos /dev/loop0
```

- Set up APT
- copy over the apt sources

```
cp /etc/apt/sources.list /mnt/etc/apt/sources.list
```

- ensure the cdrom source is commented out
- add the line:

```
deb http://ftp.debian.org/debian stable-backports  
main contrib non-free
```

```
mount --bind /dev/pts /mnt/dev/pts
```

```
mount --bind /proc /mnt/proc
```

```
mount --bind /sys /mnt/sys
```

```
mount --bind /dev /mnt/dev
```

```
# if you want your pushprofilesettings
```

```
cp ~/.bashrc /mnt/root/
```

```
cp ~/.vimrc /mnt/root/
```

```
# chroot -- enter the system as if it were thy own
```

```
chroot /mnt /bin/bash
```

```
export HOME=/root
```

```
export LC_ALL=C
```

```
export LANG=C.UTF-8
```

```
export TERM=xterm-256color
```

```
mount from man mount:
```

--bind Remount a subtree somewhere else (its contents are available in both places).

-t <type> Mount of filesystem *type*, with this, mount will attempt to auto determine

edit /etc/default/grub:

1. Set GRUB\_CMDLINE\_LINUX="" to:

```
GRUB_CMDLINE_LINUX="console=tty0  
console=ttyS0,115200n8"
```

2. Uncomment GRUB\_TERMINAL=console

3. Beneath, add the line:

```
GRUB_SERIAL_COMMAND="serial --speed=115200  
--unit=0 --word=8 --parity=no --stop=1"
```

Make the grub config - **This MUST be done in a non-systemd-nspawn shell (that means chroot)**

```
grub-mkconfig -o /boot/grub/grub.cfg
```

```
exit
```

```
umount /mnt/sys
```

```
umount /mnt/dev
```

```
umount /mnt/dev/pts
```

```
umount /mnt/proc
```

**Can check for additional mounts with:** `mount | grep /mnt` **and then unmount them with** `umount`

```
systemd-nspawn -D /mnt
```

```
# not you are in a special container
```

**Set the password for root with passwd**

In /etc/ssh/sshd\_config comment out PermitRootLogin

```
without-password to read #PermitRootLogin without-  
password and insert PermitRootLogin yes beneath it
```

Now enable ssh on startup

```
systemctl enable ssh  
  
# this is needed to clean up both chroot and  
systemd-nspawn -D /mnt  
# once this is run you can not do systemd-nspawn  
either so wait until you are entirely done  
exit  
umount /mnt  
losetup -d /dev/loop0
```

*Check for additional mounts with:* `mount | grep /mnt` *If*  
**ANYTHING** is returned, *unmount them with* `umount`

## Recover (only necessary in ERROR)

**If you broke something, or need to retry, RE-MOUNT / SETUP  
CHROOT on existing .img:**

```
losetup --show -f 1445.img  
# prints out `/dev/loopX`, enter this on the next  
lin  
partprobe /dev/loop0  
mount /dev/loop0p1 /mnt
```

## testing img

```
qemu-system-x86_64 -hda 1445.img -m 1024 -vnc :0
```