Modifying initrd image

From OpenVZ Linux Containers Wiki

The frequent reason why your kernel can't boot is not properly created initrd image. Here is a small description of what you can do if you encounter a similar problem.

Contents

What is initrd image

Your boot loader usually supports `initrd` instruction. For example, in GRUB:

```
OpenVZ (2.6.8-022stab077)
        root (hd0,0)
        kernel /vmlinuz-2.6.8-022stab077 ro root=LABEL=/ console=tty0
        initrd /initrd-2.6.8-022stab077.img
```

GRUB loads initrd-2.6.8-022stab077.img file at a certain address in memory. When kernel boots, it checks for initrd image, and if it exists starts `init` script that resides on this image. `init` script is usually written in nash (a sort of bash-like shell, just smaller). When `init` script on initrd image is finished, kernel usually calls standard System V `init` process (/sbin/init, etc.)

Why initrd image is necessary

Suppose your root partion resides on some SCSI device and driver for this SCSI devices is compiled as a kernel module. Of course this module is required at boot time to have access to the root partion — but it is not in the kernel. Thus the need for an initrd image.

Additionally after udev subsystem become common, somebody has to start udev to create device nodes. This is initrd's duty too.

Typical problem

Consider a real problem. After booting the kernel we get the following:

```
...
Creating root device
mkrootdev: label / not found
Mounting root filesystem
mount: error 2 mounting ext3
mount: error 2 mounting none
Switching to new root
switchroot: mount failed: 22
umount /initrd/dev failed: 2
Kernel panic - not syncing: Attempted to kill init!
```

This can appear if there is no module loaded for device, where root partion resides. To solve the problem, extract the initrd image.

Extracting initrd image

Initrd image is just cpio-gzip archive. So to extract it:

```
$ mkdir initrd
$ cd initrd
$ gzip -dc /boot/initrd-2.6.16-026test014.4-smp.cpio | cpio -id
$ ls -1
```

```
bin
dev
etc
init
initrd-2.6.16-026test014.4-smp.cpio
lib
loopfs
proc
sbin
sys
sysroot
```

Analyzing init script

```
$ cat init
#!/bin/nash

mount -t proc /proc /proc
setquiet
echo Mounted /proc filesystem
echo Mounting sysfs
mount -t sysfs none /sys
echo Creating /dev
mount -o mode=0755 -t tmpfs none /dev
mknod /dev/console c 5 1
mknod /dev/null c 1 3
mknod /dev/zero c 1 5
mkdir /dev/pts
mkdir /dev/shm
echo Starting udev
/sbin/udevstart
echo -n "/sbin/hotplug" > /proc/sys/kernel/hotplug
echo "Loading mptbase.ko module"
insmod /lib/mptbase.ko
echo "Loading mptscsih.ko module"
insmod /lib/mptscsih.ko
/sbin/udevstart
echo Creating root device
mkrootdev /dev/root
umount /sys
echo Mounting root filesystem
mount -o defaults --ro -t ext3 /dev/root /sysroot
mount -t tmpfs --bind /dev /sysroot/dev
echo Switching to new root
switchroot /sysroot
umount /initrd/dev
```

We can see that init tries to load modules `mptbase.ko` and `mptscsih.ko`. Check for presense of these modules on initrd image:

```
$ ls -1 ./lib/
mptbase.ko
mptscsih.ko
```

So they are here... But on the node in question there is a device supported by driver in another module: `mptspi.ko`! After adding it to the image and into init script everything should work.

Creating initrd

We just have to cpio and gzip directory cpio:

```
$ find ./ | cpio -H newc -o > /boot/new-initrd.cpio
1354 blocks
$ cd /boot
$ gzip new-initrd.cpio
$ mv new-initrd.cpio.gz new-initrd.img
```

Next, try to boot your kernel with newly created initrd image.

Who create initrd by default?

Usually there is an `mkdinitrd` package installed, that allows to create initrd image. You can use this program, it has a lot of options. OpenVZ kernel RPM-package (and “make install” target too) uses this program to create an initial (default)

of options. OpenVZ kernel RPM package (and "make install" target too) uses this program to create an initial (default) initrd image.

Retrieved from "http://wiki.openvz.org/Modifying_initrd_image"
Categories: HOWTO | Troubleshooting

HOWTO
Troubleshooting