
Building Installation Media

Table of Contents

1. Determining The Requirements	1
2. Constructing The Root Filesystem	1
2.1. Make A List Of Necessary Commands	1
2.2. Build The Required Packages	2
2.3. Create A Staging Area	2
2.4. Extract The Required Files	2
2.5. Create Configuration Files	3
2.6. Test The Files In The Staging Area	3
2.7. Create A Filesystem Image	3
3. Building A Boot/Root Diskset	3
3.1. Create A Boot Disk	3
3.2. Write The Root Filesystem Image To Diskette	4
3.3. Test The Configuration	4
4. Building A Bootable CD-ROM	4

1. Determining The Requirements

The GNU/Linux System Architect Toolkit is designed to help people create their own distributions built from source code. Users of the Architect Toolkit decide which packages they need for their distribution. After downloading the source code onto a development system Architect takes care of turning the source into binary packages. Once all of the binary packages are created they need to be installed onto the target system. And that is the task of the installation media.

The installation media is a bootable system that contains just enough functionality to get packages installed onto the target media. Such a limited system can be made small enough to fit on a couple of floppy disks or a CD-ROM. Floppy-based installation disks have the advantage of being easy to build and make changes to, but they are severely limited on storage capacity. CD-ROM's have plenty of storage space, but are more tedious to build and, in the case of non-rewritable disks, less forgiving of mistakes. We will explore both types of installation media starting with floppy disks and then moving into bootable CD-ROMs.

2. Constructing The Root Filesystem

Once the installation media has booted it really does not matter if it is a floppy diskset or a CD-ROM, it will have the same functionality.

2.1. Make A List Of Necessary Commands

To install a new system we need to:

1. Partition disks - fdisk, sfdisk
2. Mirror partitions - mdadm
3. Create filesystems - mke2fs

4. Mount filesystems - mount
5. Fetch packages - ifconfig, tftp
6. Install packages - tar, gz
7. Install a boot loader - grub

In addition we need a shell and any required library files.

2.2. Build The Required Packages

We will need to build the following packages:

- bash-mini
- coreutils
- e2fsprogs
- glibc
- grub (or grub-serial0)
- gzip
- mdadm (optional)
- net-tools (optional)
- netkit-tftp (optional)
- tar
- util-linux

2.3. Create A Staging Area

2.4. Extract The Required Files

- bash-mini - (extract all)
- coreutils - bin/cat, bin/chmod, bin/chown, bin/cp, bin/ln, bin/lis, bin/mkdir, bin/mknod, bin/mv, bin/rm, bin/stty, bin/sync
- e2fsprogs - sbin/mke2fs
- glibc - lib/ld-linux.so.2, lib/libc.so.6, lib/libdl.so.2, lib/librt.so.1, lib/libpthread.so.0
- grub (or grub-serial0) - boot, sbin/grub
- gzip - bin/gzip, bin/gunzip

- mdadm (optional) - sbin/mdadm
- net-tools (optional) - sbin/ifconfig sbin/route
- netkit-tftp (optional) - bin/tftp
- tar - bin/tar
- util-linux - bin/mount, bin/umount, sbin/fdisk, sbin/sfdisk

2.5. Create Configuration Files

We need an a blank `mtab` and an `fstab` that looks like this:

```
/dev/ram0    /          ext2        defaults    0          0
proc         /proc      proc        defaults    0          0
```

We need an `/sbin/init.sh` that looks like this:

```
#!/bin/sh
PATH=/sbin:/bin
export PATH
mount -o remount,rw /
mount -a
```

2.6. Test The Files In The Staging Area

Using the `chroot` command

2.7. Create A Filesystem Image

Using a loopback device

3. Building A Boot/Root Diskset

3.1. Create A Boot Disk

1. Decide if the boot disk will be used on a machine that is set up with a serial console or if it will have a monitor and keyboard.
2. Choose either the `grub-serial0` package or the `grub` package depending on the decision made in the previous step.
3. Extract the chosen package to a temporary location.

```
bash# cd /var/tmp
bash# tar -zxf grub-serial0-0.95.i386.tar.gz
```

4. Create a second extended filesystem on a floppy disk and mount it on `/media/floppy`.

```
bash# mke2fs -m0 /dev/fd0
bash# mount /dev/fd0 /media/floppy
```

5. Copy the entire /boot directory from the grub package onto the floppy.

```
bash# cp -R /var/tmp/grub-serial0-0.95/boot /media/floppy
```

6. Copy a Linux kernel to the diskette.

```
bash# cp /usr/src/linux/arch/i386/boot/zImage /media/floppy/boot/vmlinuz
```

7. Create an appropriate menu.lst file if desired.

```
bash# cat <<"END-OF-FILE" >/media/floppy/boot/grub/menu.lst
> # GRUB configuration file for serial console operation
> serial --unit=0 --speed=9600
> terminal serial
> default 0
> timeout 3
> title Serial Console Boot Disk
> kernel (fd0)/boot/vmlinuz root=/dev/fd0 load_ramdisk=1 prompt_ramdisk=1 console=ttyS0
> END-OF-FILE
```

8. Unmount the floppy.

```
bash# umount /dev/fd0
```

9. Run the grub binary from the recently extracted grub package.

```
bash# /var/tmp/grub-serial0-0.95/sbin/grub
```

10. Execute the grub commands to set up the bootloader code on the floppy.

```
grub> root (fd0)
grub> setup (fd0)
```

3.2. Write The Root Filesystem Image To Diskette

3.3. Test The Configuration

4. Building A Bootable CD-ROM