

Rolling Your Own Building a REAL Internet Appliance

Triangle Linux Users Group

Alan Porter
porter@trilug.org
<http://www.AlanPorter.com>

September 2008

Agenda

- TMIO, company, background (video)
- Stealing with pride: O'Reilly
- Packages, building (example)
- Delivering, tmioppix (demo)
- Startup sequence
- Filesystems
- The finished product (demo)
- Q/A



Company timeline

- Tinkering ~2000
- TMIO (TN) started in 2003
- Software office (NC) opened in 2004
- Presented at CES (Las Vegas) in 2005
- Available for sale in 2005/2006
- Operations scaled back in late 2006



Video

- Shown at the Consumer Electronics Show (CES) in Las Vegas, January 2005



Trim a Distro - or - Roll Your Own

- Option 1 - trim an existing Distro
 - small distros: DSL, Slackware, Linux From Scratch
 - find what you DON'T need, trim (hard to know)
- Option 2 - roll your own
 - start with nothing, add what you need (easier)
 - understand the entire system
 - more fun!

Building Embedded Linux Systems

by Karim Yaghmour (O'Reilly)

Need 3 things:

- kernel
- basic filesystem
 - /etc/{passwd,group,hosts},
/etc/{ld.so.conf,modules.conf,nsswitch.conf}
 - busybox, /etc/inittab
- bootloader: LILO or GRUB



- One executable file, many functions:
 - 1.1 MB for all of this:

```
[, [[, addgroup, adduser, adjtimex, arping, ash, awk, basename, bunzip2, busybox,
bzip2, cal, cat, chgrp, chmod, chown, chroot, chvt, clear, cmp, cp, cpio, crond,
crontab, cut, date, dd, deallocvt, delgroup, deluser, df, dirname, dmesg,
dos2unix, du, dumpkmap, echo, egrep, env, expr, false, fbset, fdisk, fgrep, find,
fold, free, freeramdisk, fsck.minix, ftpget, ftpput, getty, grep, gunzip, gzip,
halt, head, hexdump, hostid, hostname, hwclock, id, ifconfig, ifdown, ifup, inetd,
init, install, ip, ipaddr, ipcalc, iplink, iproute, iptunnel, kill, killall,
klogd, last, length, less, ln, loadfont, loadkmap, logger, login, logname,
logread, losetup, ls, md5sum, mesg, mkdir, mkfifo, mkfs.minix, mknod, mkswap,
more, mount, mv, nameif, nc, netstat, nice, nohup, nslookup, od, openvt, passwd,
patch, pidof, ping, pivot_root, poweroff, printf, ps, pwd, rdate, readlink,
realpath, reboot, renice, reset, rm, rmdir, route, run-parts, rx, sed, seq,
setkeycodes, sh, shasum, sleep, sort, strings, stty, su, swapoff, swapon, sync,
syslogd, tail, tee, telnet, telnetd, test, tftp, time, top, touch, tr, traceroute,
true, tty, umount, uname, uncompress, uniq, unix2dos, unzip, uptime, usleep,
uudecode, uuencode, vi, vlock, watch, wc, which, who, whoami, xargs, yes, zcat
```

Early development

- Single Board Computer (SBC)
 - Arcom, EMJ Embedded (WDL Systems), TME
- Plug CF card into PC
- Copy files to CF card
- Plug CF card into prototype board
- Reboot, hope it works
- Lather, rinse, repeat



Later development

- Use SSH
- Copy rebuilt files from PC to SBC
- Copy library files directly from my Debian PC!
 - Until I could find the source for all

37 Packages

BASICS

- busybox
- devices
- kernel
- modutils

APPLICATION

- java
- xtext

BOOTING

- bootsplash
- etc
- grub

GRAPHICS

- freeglut
- freetype
- libxml2
- setterm
- touchscreen
- unclutter
- x11
- xpmroot

NETWORKING

- dhcp
- iptables
- minihttpd
- openssh
- ppp
- wget

PERIPHERALS

- aumix
- sbc5831
- sox
- wavtools

LIBRARIES

- expat
- gcc
- glibc
- libpng
- ncurses
- openssl
- zlib

SYSTEM UTILITIES

- e2fsprogs
- tar
- version



The "prep" script

- scrub - clean up build and staging areas
- populate - extract sources into our build area
- build - build sources into binaries
- stage - "make install" to staging area, tar up
- install - write image to compact flash card



The "packages"

xxxx.pkg contains 5 functions:

- `xxxx_depends` - lists dependencies
- `xxxx_populate` - extracts the source from tar
- `xxxx_build` - compiles
- `xxxx_stage` - copies deliverables to a staging area and tars it up
- `xxxx_install` - runs when CF card is created (grub and version)



Example - build

simple example

```
vi packages/minihttpd/minihttpd.pkg
```

```
./prep.pl scrub populate build stage minihttpd
```

complex example

```
vi packages/glibc/glibc.pkg
```



Delivering the goods

- The deliverable image
 - 256000000 bytes (CF's vary in size)
 - created as an image, dd-ed to the CF cards
 - part of oven boot-up is to register a serial number
- The tmioppix disk - DEMO

Oven Startup Sequence

- special graphical BIOS screen
- GRUB loads the kernel
- no initial ram disk (why?)
- bootsplash
- kernel calls init (provided by busybox)
- init reads inittab
- inittab says to load /etc/rc
- /etc/rc calls /etc/rc/rc.d/[0-9][0-9].*



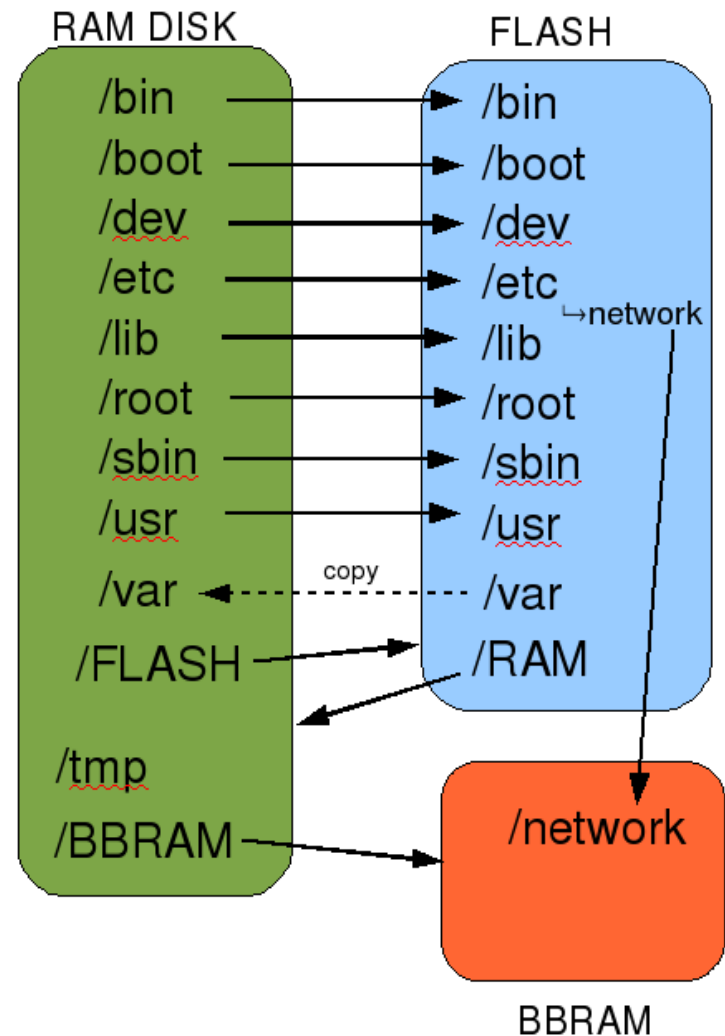
/etc/rc.d/[00-99].*

| STARTUP FILE | PACKAGE | DOES |
|----------------|-------------|--|
| 01.version | etc | boot message |
| 02.ramdisk | etc | set up filesystem |
| 03.libraries | etc | ldconfig |
| 04.modules | etc | depmod |
| 05.tmio_link | etc | link to application |
| 06.sbc5831 | sbc5831 | load module |
| 07.bbbram | etc | set up BBRAM filesystem |
| 11.hostname | etc | set hostname |
| 12.clock | etc | set clock |
| 14.aumix | aumix | set speaker volume |
| 21.firewall | iptables | set up firewall |
| 22.network | etc | start network/DHCP |
| 31.touchscreen | touchscreen | start touch screen driver |
| 32.x-windows | x11 | load X11 (starts oven GUI and daemons) |
| 99.tmio_app | etc | deprecated: start oven GUI and daemons |



Filesystems

- 1) mount flash on /
- 2) ram disk on /RAM
- 3) In -s /FLASH/xx /RAM/xx
- 4) cp -a /FLASH/var /RAM/var
- 5) mkdir /RAM/tmp
- 6) mkdir /RAM/FLASH
- 7) pivot_root /RAM /RAM/FLASH
- 8) make /proc, /dev, /BBRAM



DEMO

- Oven demo



Thank you!

- Questions?
- porter@trilug.org

