

---

# FASTBOOT

Brett Nash

nash at fluffyspider's domain

March 2004

---

## WHAT WERE WE DOING

If sharp can do it in 7, we can do it 6!<sup>a</sup>:

Sharp's fastboot laptop runs a DVD player, TV tuner app or a CD player in 7 seconds. Our job is to beat it on Toshiba's hardware.

---

<sup>a</sup>No matter how much carsten preferred Sharp's time

---

## THE BOOT PROCESS

There are quite a few steps involved in getting a machine into a useful state. All take time, and all are slow...

### BIOS:

First of the mark, embedded in hardware - ROM normally. Gets the system of the ground, does anything really hardware specific, and loads and runs the boot loader.

Toshiba is doing this...

---

## Boot Loader:

On most H/W part of the BIOS. On x86 hardware, a separate application. Loaded from reserved block on the first hard disk.

Loads the OS - often involves a selection.

LILO is fairly fast and fairly small.

However takes 1 second, and prints out some stuff...

---

## Linux Kernel:

*Writing a new OS only for the 386 in 1991  
gets you your second 'F' for this term.  
- Andy Tanenbaum*

The real controller of the hardware. Provides an interface to hardware for user applications.

Fairly cautious in hardware probing - just in case the hardware is flaky. Hence boot speed is the worst of all hardware boot sequences.

Takes about 5-10s on a default debian kernel.

---

## Init:

First app spawned by OS. Responsible for getting the system to a useful state. Actually just fires off init scripts - one-by-one.

Also takes care of orphaned processes and some other unix process related things.

---

## The Init Scripts:

Seven directories filled with weirdly named shell scripts and symlinks - Who thought this was a good idea?

Slow, ugly, slow, and barely maintainable. An absolute mess.

Did I mention there are performance related issues?

---

X:

*X-Windows: ...A mistake carried out to perfection.*

*X-Windows: ....It could be worse, but it'll take time.*

*X-Windows: ...Even your dog won't like it.*

Like the Kernel, cautious about probing hardware.

Spends a lot of time waiting for CRT to sync...

Many improvements here...

---

Toshiba's Applications:

Never seen them.

---

## INITIAL TIMES

Culling the init scripts, improved kernel.

About 30 seconds total for our stuff.

---

## WHAT ARE WE AFTER

- << 1s LILO
- < 1s Kernel
- < 1s Init
- < 1s X and Application

---

## OUR IMPROVEMENTS

### LILO:

- Code inspect for horrible inefficiencies
- Remove printing code
- Configure will 0s wait, no prompt
- Force to use a single BIOS call to load disk image all at once.
- Remove the other copies of the printing code...

---

## LILO (cont)...

Also make sure disk image and kernel image are not fragmented in partition.

LILO whines and screams about loading image with one BIOS call - works fine on modern hardware.

Final time << 1s! Success!

```
drift:~# lilo
```

```
Warning: Video adaptor (CGA) is incompatible with the boot loader  
installation ('install = menu')
```

---

## Disk Image:

We use the initrd system for our main filesystem.

Compressed image loaded by LILO, mounted by kernel as a ram disk.

Loading this is slowest part of Boot loader...

---

## Linux Kernel:

*You are in a maze of twisting  
configuration options all alike*

First change: Remove support for hardware which is NOT on the system. Be aggressive.

Second change: Put back the stuff we actually need...  
Promise not to be so aggressive next time.

Third change: Remove support for hardware we won't use.  
eg ethernet.

This takes time down to around 2 seconds...

...we can do better.

---

## Linux (cont)...

Biggest remaining for loading is detection of IDE devices...

They don't change on a laptop. Hardcode the drive information.

While we are there...

- HLT instruction works - don't test it - 4 times.
- Jiffies don't change - hardcode.
- We know what CPU we have - it's still not a 68k, so don't test it.

Kernel now loads in < 0.4 seconds! Slowest part is uncompressing and mounting the initrd image!

---

Init:

```
drift:~# rm -f /sbin/init
```

Init Scripts:

```
drift:~# rm -rf /etc/init.d /etc/rc?.d
```

---

## Mini Init:

A new init - smaller, faster...

...more likely to crash into a planet

- Hard coded startup sequence.
- Waits on a job only if necessary.
- Essentially fires off everything at once.

Total time: Too small to measure.

---

## X Changes:

Minimal configuration, minimal fonts, minimal modules.

Cuts time to a few seconds - need better.

Main cause of delay is waiting for things. Also probe delays...

Remove delays for an LCD.

Time  $< 1s$ , some of that is waiting for the backlight to come on...

---

## WRAP UP

Times  $< 6s$  for booting - way in front of sharps fastboot system, or other systems in the media.

Kernel boot time comparable to best time on a embedded devices with flash.

---

## QUESTIONS?

And no Asok, I don't like wood.