

A Guided Tour of a Linux Boot

Craig Van Degrift

The firmware and software programs output various messages as the computer and Linux come to life. These notes attempt to help clarify what routines and devices are doing.

1. The Motherboard BIOS Triggers the Video Display Card BIOS Initialization

The Matrox Millennium adapter BIOS sends the following to the screen:
MATROX POWER GRAPHICS ACCELERATOR

MGA Series

VGA/VBE BIOS, Version V2.2
Copyright (C) 1995, Matrox Graphic, Inc.
Copyright (C), LSI Logic Corporation, Inc. 1990-1991

The screen is cleared.

2. Motherboard BIOS Initializes Itself

The Motherboard BIOS sends the following to the screen:

*Award Modular BIOS v4.60PGA, An Energy Star Ally
Copyright (C) 1984-98, Award Software, Inc.

Version 1.15JE33

Cyrix M II/IBM 6x86MX-233 CPU Found

Memory Test: 131072K OK

Award Plug and Play BIOS Extension v1.0A
Copyright (C) 1998, Award Software, Inc.

Press DEL to enter SETUP
11/19/1998 - VP3 - 536B - W877 - 2A5LEF09C - 00

The screen is cleared.

3. SCSI Controller BIOS Initializes

The SCSI controller BIOS sends the following to the screen:

Adaptec AHA-2940 Ultra/Ultra W BIOS v 1.23
(c) 1996 Adaptec, Inc. All Rights Reserved.
<<< Press for SCSISelect(TM)

SCSI ID:LUNNUMBER ## 0:0 - MICROP 3243-19 1128RQAV - Drive C: (80h)
SCSI ID:LUNNUMBER ## 1:0 - SEAGATE ST51080N - Drive D: (81h)
SCSI ID:LUNNUMBER ## 2:0 - HP C2520A

A Guided Tour of a Linux Boot

Craig Van Degrift

SCSI ID:LUNNUMBER ## 5:0 - Sony CD-R-625

Bios installed successfully!

The screen is cleared.

4. Hardware Summary

The motherboard BIOS then displays the following summary of its hardware inventory:

Award Software, Inc.

CPU Type	: Cyrix M II/IBM 6x86MX	Base Memory	: 640K
Co-Processor	: Installed	Extended Memory:	130048K
CPU Clock	: 233	Cache Memory	: 1024K

Diskette Drive A: 1.44M, 3.5 in.	Display Type	: EGA/VGA
Diskette Drive B: 1.44M, 3.5 in.	Serial Port(s)	: 3F8 2F8
Pri. Master Disk: None	Parallel Port(s)	: 278
Pri. Slave Disk : None	Bank 0/1 DRAM Type	: None
Sec. Master Disk: None	Bank 2/3 DRAM Type	: EDO
Sec. Slave Disk : None	Bank 4/5 DRAM Type	: EDO

And Runs its Virus checking code that looks for changed boot sectors.

!!!! Trend ChipAwayVirus On Guard !!!! Now Detecting Boot Sector Type Virus...
ChipAwayVirus BIOS Version 1.62 Verifying DMI Pool Data.....

The screen is cleared.

5. OS/2 BootManager Menu

The Master Boot Record (MBR) on the first hard disk is read, by DOS tradition, into address 0x00007c00, and the processor starts executing instructions there.

This MBR boot code loads the first sector of code on the active DOS partition. In my case this is the OS/2 BootManager which displays the following menu of OS choices:

Boot Manager				
DOS 7.1	Disk 1	C:Primary	408M	FAT
OS2Warp4	Disk 1	E:Logical	2573M	HPFS
TinyWarp	Disk 1	F:Logical	142M	HPFS
BigLinux	Disk 2	Logical	1020M	Type 83

A Guided Tour of a Linux Boot

Craig Van Degrift

```
|
| Timer Disabled, a timeout boot will not occur
|
| Use ^ or V to select. Press Enter to boot.
|
```

6. Lilo is started

If the BigLinux selection is chosen and if Linux has been installed with Lilo, Lilo is loaded into address 0x00007c00.

Lilo prints

LILO

with its progress revealed by individually printing the letters. The first "L" is printed after Lilo moves itself to a better location at 0x0009A000. The "I" is printed just before it starts its secondary boot loader code.

Lilo's secondary boot loader prints the next "L", loads descriptors pointing to parts of the kernel, and then prints the final "O". The descriptors are placed at 0x0009d200.

The boot message and a prompt line, if specified, are printed. The pressing "Tab" at the prompt, allows the user to specify a system and to provide command-line specifications to the Linux Kernel, its drivers, and the "init" program. Also, environment variables may be defined at this point.

The following line is from /boot/message:

```
>>> Press to list available boot image labels.
```

The following line is the prompt from /sbin/lilo:

boot:

Note: If Lilo is not used, then the boot code built into the head of the Linux kernel, linux/arch/i386/boot/bootsect.S prints "Loading" and continues.

Lilo displays the following as it loads the kernel code. It gets the text "Linux-2.2.12" from the "label=..." specification in lilo.conf.

```
Loading linux-2.2.12.....
```

7. The Linux Kernel Initializes

A Guided Tour of a Linux Boot

Craig Van Degrift

The kernel code in `/linux/arch/i386/boot/setup.S` arranges the transition from the processor running in real mode (DOS mode) to protected mode (full 32-bit mode). Blocks of code named `Trampoline.S` and `Trampoline32.S` help with the transition. Small kernel images (`zImage`) are decompressed and loaded at `0x00010000`. Large kernel images (`bzImage`) are loaded instead at `0x00100000`. This code sets up the registers, decompresses the compressed kernel (which has `linux/arch/i386/head.S` at its start), printing the following 2 lines from `linux/arch/i386/boot/compressed/misc.c`

Uncompressing Linux... Ok. Booting the kernel.

The i386-specific `setup.S` code has now completed its job and it jumps to `0x00010000` (or `0x00100000`) to start the generic Linux kernel code.

- o **Processor, Console, and Memory Initialization**

This runs `linux/arch/i386/head.S` which in turn jumps to `start_kernel(void)` in `linux/init/main.c` where the interrupts are redefined.

`linux/kernel/module.c` then loads the drivers for the console and pci bus.

From this point on the kernel messages are also saved in memory and available using `/bin/dmesg`. They are then usually transferred to `/var/log/message` for a permanent record. The following line is from `linux/init/version.c`:

```
Linux version 2.2.12 (root@Baldy) (gcc version 2.8.1) #1 Thu Nov 4 19:44:40 PST
1999
Detected 187547890 Hz processor.
```

`console_init(..)` in `linux/init/main.c` causes the following line from `linux/drivers/console.c` to be printed:

```
Console: colour VGA+ 132x60
```

`calibrate_delay()` in `linux/init/main.c` prints the following line:

```
Calibrating delay loop... 187.19 BogoMIPS
```

`mem_init()` in `linux/init/main.c` causes the following line from `linux/arch/i386/mm/init.c` to be printed:

```
Memory: 127952k/131008k available (952k kernel code, 412k reserved, 1652k data,
40k init)
```

`buffer_init()` in `linux/init/main.c` causes the following line from `linux/arch/i386/mm/fault.c` to be printed:

A Guided Tour of a Linux Boot

Craig Van Degrift

Checking if this processor honours the WP bit even in supervisor mode... Ok.

check_bugs() in /linux/init/main.c causes the following lines from linux/include/asm-i386/bugs.h to be printed:

CPU: Cyrix M II 2.5x Core/Bus Clock stepping 08
Checking 386/387 coupling... OK, FPU using exception 16 error reporting.
Checking 'hlt' instruction... OK.
POSIX conformance testing by UNIFIX
mtrr: v1.35a (19990819) Richard Gooch (rgooch@atnf.csiro.au)

o **PCI Bus Initialization**

mpci_init() in linux/init/main.c causes the following lines from linux/arch/i386/kernel/bios32.c to be printed:

PCI: PCI BIOS revision 2.10 entry at 0xfb2d0
PCI: Using configuration type 1

pci_init() in linux/init/main.c causes the following line from linux/drivers/pci/pci.c to be printed:

PCI: Probing PCI hardware

linux/drivers/quirks.c prints the following:

PCI: 00:38 [1106/0586]: Work around ISA DMA hangs (00)
Activating ISA DMA hang workarounds.

o **Network Initialization**

socket_init() in linux/init/main.c causes the following network initializations:

linux/net/socket.c prints:

Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039

linux/net/unix/af_unix.c prints:

NET4: Unix domain sockets 1.0 for Linux NET4.0.

linux/net/ipv4/af_inet.c prints:

NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP

linux/net/ipv4/ip_gre.c prints:

A Guided Tour of a Linux Boot

Craig Van Degrift

GRE over IPv4 tunneling driver

linux/net/core/dev.c prints:

early initialization of device gre0 is deferred

linux/net/core/rtnetlink.c prints:

Initializing RT netlink socket

- **The Kernel Idle Thread (Process 0) is Started**

At this point a kernel thread is started running `init()` which is one of the routines defined in `linux/init/main.c`. This `init()` must not be confused with the program `/sbin/init` that will be run after the Linux kernel is up and running. `mkswapd_setup()` in `linux/init/main.c` causes the following line from `linux/mm/vmscan.c` to be printed:

Starting kswapd v 1.5

- **Device Driver Initialization**

The kernel routine `linux/arch/i386/kernel/setup.c` then initializes devices and file systems (built into the kernel??). It produces the following lines and then forks to run `/sbin/init`:

- **Generic Parallel Port Initialization**

The parallel port initialization routine `linux/drivers/misc/parport_pc.c` prints the following:

parport0: PC-style at 0x378 [SPP,PS2,EPP]

- **Character Device Initializations**

The following 3 lines are from `linux/drivers/char/serial.c`:

Serial driver version 4.27 with no serial options enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A

The following line is from `linux/drivers/char/lp.c`:

lp0: using parport0 (polling).

linux/drivers/char/rtc.c prints:

Real Time Clock Driver v1.09

- **Block Device Initializations**

A Guided Tour of a Linux Boot

Craig Van Degrift

linux/drivers/block/rd.c prints:

RAM disk driver initialized: 16 RAM disks of 8192K size

linux/drivers/block/loop.c prints:

loop: registered device at major 7

linux/drivers/block/floppy.c prints:

Floppy drive(s): fd0 is 1.44M, fd1 is 1.44M
FDC 0 is a post-1991 82077

▪ SCSI Bus Initialization

The following lines are from aic7xxx.c, scsi.c, sg.c, sd.c or sr.c in the subdirectory linux/drivers/scsi:

```
(scsi0) found at PCI 9/0
(scsi0) Wide Channel, SCSI ID=7, 16/255 SCBs
(scsi0) Warning - detected auto-termination
(scsi0) Please verify driver detected settings are correct.
(scsi0) If not, then please properly set the device termination
(scsi0) in the Adaptec SCSI BIOS by hitting CTRL-A when prompted
(scsi0) during machine bootup.
(scsi0) Cables present (Int-50 YES, Int-68 NO, Ext-68 YES)
(scsi0) Downloading sequencer code... 413 instructions downloaded
scsi0 : Adaptec AHA274x/284x/294x (EISA/VLB/PCI-Fast SCSI) 5.1.19/3.2.4
```

```
scsi : 1 host.
```

```
(scsi0:0:0:0) Synchronous at 20.0 Mbyte/sec, offset 8.
Vendor: MICROP Model: 3243-19 1128RQAV Rev: RQAV
Type: Direct-Access ANSI SCSI revision: 02
```

```
Detected scsi disk sda at scsi0, channel 0, id 0, lun 0
```

```
(scsi0:0:1:0) Synchronous at 10.0 Mbyte/sec, offset 15.
```

```
Vendor: SEAGATE Model: ST51080N Rev: 0958
```

```
Type: Direct-Access ANSI SCSI revision: 02
```

```
Detected scsi disk sdb at scsi0, channel 0, id 1, lun 0
```

```
Vendor: HP Model: C2520A Rev: 3503
```

```
Type: Processor ANSI SCSI revision: 02
```

```
Detected scsi generic sgc at scsi0, channel 0, id 2, lun 0
```

```
(scsi0:0:6:0) Synchronous at 5.0 Mbyte/sec, offset 15.
```

```
Vendor: SONY Model: CD-R CDU926S Rev: 1.1f
```

```
Type: CD-ROM ANSI SCSI revision: 02
```

```
Detected scsi CD-ROM sr0 at scsi0, channel 0, id 6, lun 0
```

```
scsi : detected 1 SCSI cdrom 2 SCSI disks total.
```

```
sr0: scsi3-mmc drive: 0x/0x caddy
```

```
Uniform CDROM driver Revision: 2.55
```

```
SCSI device sda: hdwr sector= 512 bytes. Sectors= 8388315 [4095 MB] [4.1 GB]
```


A Guided Tour of a Linux Boot

Craig Van Degrift

```
#
# Runlevel 0 is halt.
# Runlevel 1 is single-user.
# Runlevels 2-5 are multi-user.
# Runlevel 6 is reboot.

l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6

# The following run when you do 'telinit A' or 'telinit B' and are disconnected from
# getting Ctrl-C, Ctrl-\, and Ctrl-Z from the keyboard. Ctrl-D will logout from the
# current version of bash that has been temporarily replaced by the "demand" program.
# The demand program's parent is init, not bash, and bash can still be used even
# though the output from the demand program is going to the same terminal.
#
la:A:wait:/home/c/sigplayposix
lb:B:wait:/home/c/sigplayold
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin

# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

# shutdown -t1 => 1 second pause before shutdown
# shutdown -a => use /etc/shutdown.allow (root always allowed)
# shutdown -r => reboot after shutdown
# shutdown -h => halt after shutdown

# What to do when the power fails/returns.
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
# Format:
# :::
1:12345:respawn:/sbin/agetty 9600 tty1
2:2345:respawn:/sbin/agetty 9600 tty2
3:2345:respawn:/sbin/agetty 9600 tty3
4:2345:respawn:/sbin/agetty 9600 tty4
5:2345:respawn:/sbin/agetty 9600 tty5
6:2345:respawn:/sbin/agetty 9600 tty6
```

A Guided Tour of a Linux Boot

Craig Van Degrift

```
x:5:respawn:/usr/X11R6/bin/xdm -nodaemon

#n1:234:wait:/etc/init.d/rc.inet1
#n2:234:wait:/etc/init.d/rc.inet2

# Example how to put a getty on a serial line (for a terminal)
#
#T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100
#T1:23:respawn:/sbin/getty -L ttyS1 9600 vt100

# Example how to put a getty on a modem line.
#
#T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS3
```

When this is processed at the very end of the booting process, the following lines are printed:

```
Activating swap...
Checking root filesystem...
Parallelizing fsck version 1.04 (16-May-96)
Checking all file systems.
[/sbin/fsck.ext2]fsck.ext2 -a /dev/sdb5
/dev/sdb5: clean, 26562/261120 files, 545241/1044193 blocks
Checking all filesystems...
Parallelizing fsck version 1.04 (16-May-96)
/dev/sda2: clean, 28715/376832 files, 1174190/1502077 blocks
/dev/sda7: clean, 450/26104 files, 15544/104391 blocks
Mounting local file systems...
/proc on /proc type proc (rw)
/dev/sda2 on /mnt/extra type ext2 (rw)
/dev/sdia7 on /mnt/backup type ext2 (rw)
Telling the kernel that the machine time is local, not GMT
Mon Nov 29 19:47:59 PST 1999
Cleaning up: /tmp /var/lock /var/run.
Starting System logger: syslogd.
Starting Kernel logger: klogd.

Starting BSD Printer daemon lpd: lpd.
Initializing random number generator... done.
INIT: Bootwait action - sleeping 5 seconds...
INIT: Boot (no wait) - sleeping 15 seconds...
INIT: Bootwait - Done napping.
```

Upon entering run level 2 it prints:

```
INIT: Entering runlevel: 2
```

Notice how /sbin/init waited for the bootwait action to finish before going to run level 2, the default runlevel specified by the id:2:default: entry in /etc/inittab. At this point all actions which are associated with runlevel 2 are started. Specifically, for this simplified /etc/inittab these are:

```
rc 2
```

A Guided Tour of a Linux Boot

Craig Van Degrift

/etc/init.d/rc takes the parameter 2 and walks through each link in the directory /etc/rc2.d, killing ("K") or starting ("S") scripts in the order specified by their numbering ("00" through "99"). An alternate arrangement for these script files has them placed one directory lower in (/etc/rc.d/rc2.d, etc.).

/etc/inittab also specifies that all 6 virtual consoles shall be set up in run level 2:

```
/sbin/agetty 9600 tty1
/sbin/agetty 9600 tty2
/sbin/agetty 9600 tty3
/sbin/agetty 9600 tty4
/sbin/agetty 9600 tty5
/sbin/agetty 9600 tty6
```

Each of the agetty's starts a login shell, /sbin/login, and after a log-in, starts /bin/sh which is typically a symbolic link to /bin/bash.

At this point most distributions clear the screen and set the cursor set to the top left character.

For illustrative purposes, I have made my /etc/init.d/rcS script indicate when it is started and we get the following:

```
Starting initscript...
Starting initscript...
Starting initscript...
Starting initscript...
    Starting initscript...
        Starting initscript...
            Starting initscript...
```

/sbin/login copies the file /etc/issue to the screen before prompting for the user name and password:

Kanji-Flash Softworks

Baldy login: INIT: Boot (no wait) - Done napping.root

Password:

Last login: Mon Nov 29 17:54:57 1999 on tty2

Linux Baldy 2.2.12 #1 Thu Nov 4 19:44:40 PST 1999 i686 unknown

9. The Bash Shell is Started

The bash shell, /bin/bash is then started up. Bash initialization begins by executing script in /etc/profile which set the system-wide environment variables:

```
# /etc/profile
# System wide environment and startup programs
```

A Guided Tour of a Linux Boot

Craig Van Degrift

```
# Functions and aliases go in $HOME/.bashrc

export PATH="/bin:/usr/bin:/usr/X11R6/bin:/opt/bin:/usr/local/bin:/usr/local/java/bin:./"

umask 022

if [ `id -gn` = `id -un` ] && [ `id -u` != 0 ]; then
    umask 002
fi

if [ -z "$UID" ]; then
    UID=`id -u`
fi

if [ "$UID" = 0 ]; then
    PATH=/sbin:/usr/sbin:$PATH
else
    PATH=$PATH:
fi

export USER=`id -un`
export LOGNAME=$USER
export HOSTNAME=`/bin/hostname`
export LOCATE_PATH=/var/state/misc/locatedb
```

Next, Bash executes the script in `/root/.profile` for user-specific customizations:

```
# ~/.profile --
# The personal initialization file, executed for login shells

if [ -n "$BASH_VERSION" ]; then
    if [ -r "$HOME/.bashrc" ]; then
        # login shells are always interactive, are they?
        . $HOME/.bashrc
    fi
fi
```

In this case, it merely has the shell execute the script `.bashrc` in the user's home directory. Both `.profile` and `.bashrc` set user-specific customizations, but only `.bashrc` is run at each shell invocation. `.profile` is only run at log-in.

In my case for root, `/root/.bashrc` contains:

```
#Make backspace work as expected by PC users but otherwise leave as expected by remote
terminal users.
if [ -z "$TERM" ]; then
    echo ".bashrc: TERM empty: this shouldn't happen!" 1>&2
    echo " Please contact 'support@lst.de'" 1>&2
else
```

A Guided Tour of a Linux Boot

Craig Van Degrift

```
case $TERM in
linux*)
  stty erase '^?'
  ;;
*)
  stty erase '^H'
  ;;
esac
fi

#Exports some environmental variables:

#export GROFF_TTYPESETTER=latin1
#export LC_CTYPE=iso-8859-1
#export METAMAIL_PAGER=less

export LESSCHARSET=latin1
export PS1="\[\033[0m\033[1;32m\][\d - \@: \w]\[\033[1;37m\] "
export TERMINFO=/usr/lib/terminfo
export HISTSIZE=200
export LD_LIBRARY_PATH=/usr/local/lib
export PRINTER=magic
export
LS_COLORS='no=01;37:fi=01:di=01;34:ln=01;36:pi=40;33:so=01;35:bd=40;33;01:cd=40;33;0
1:or=40;31;01:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.zip=01;3
1:*.z=01;31:*.Z=01;31:*.gz=01;31:*.deb=01;31:*.jpg=01;35:*.gif=01;35:*.bmp=01;35:*.ppm=01;
35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.mpg=01;37:*.avi=01;37:*.gl=01;37:*.dl=
01;37:'

#Set some aliases:

alias which='type -path'
alias ll="ls --almost-all -l --classify --full-time --color"
alias ls="ls -A --color"
alias showhex="/usr/bin/hexdump -f /etc/hexdump.fmt"
alias cc="gcc -DEOF=-1 -Wall -lint"
alias whatfor="whatfor --color --all"
alias last="last -n 10 -a -x"
alias less="less --raw-control-chars --LONG-PROMPT --force"
alias pssig="ps -sx --forest"
alias psx="ps -o user,ppid,pid,pgid,TTY,stat,command --forest x"
alias psw="ps -aux --forest"

#Set some key bindings for bash's command line input:

# Home
bind ""\e[1~": beginning-of-line'
# Insert
bind ""\e[2~": abort'
# Delete
bind ""\e[3~": delete-char'
# End
```

A Guided Tour of a Linux Boot

Craig Van Degrift

```
bind ""\e[4~": end-of-line'  
# Page Up  
bind ""\e[5~": abort'  
# Page Down  
bind ""\e[6~": abort'
```

#Sets tab stops to every 4th character:

```
setterm -clrtabs  
setterm -regtabs 4
```

Finally, bash initialization is complete and we get the bash prompt!!!!