

Kernel - Common Problems Booting Linux

From Texas Instruments Wiki

Jump to: [navigation](#), [search](#)

Often users may find that when attempting to boot the Linux kernel, the kernel fails at some point during the boot sequence. Sometimes this can be caused by a software bug, but sometimes this can also be caused through mis-configuration of the kernel. This article provides some tips to help diagnose the why a particular version of a kernel is not booting and what to look for.

Contents

[hide]

- [1 Problem #1 - No more output is seen on the console after "Starting Kernel..."](#)
- [2 Problem #2 - No more output is seen on the console after "booting the kernel"](#)
 - [2.1 Cause #1 - The linux console boot parameter is incorrect](#)
 - [2.2 Cause #2 A - Mis-match between boot-loader and kernel machine numbers](#)
 - [2.3 Cause #2 B - Mis-match between device tree blob and the actual hardware](#)
 - [2.4 Cause #3 - A software bug](#)
- [3 Problem #3 - No console prompt seen after kernel boot](#)

Problem #1 - No more output is seen on the console after "Starting Kernel..."[\[edit\]](#)

The first problem a user may encounter when booting the kernel, is that no further output after "Starting kernel..." is seen on the console. For example:

```
## Booting kernel from Legacy Image at 80300000 ...
Image Name:   Linux-2.6.31
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    1750680 Bytes = 1.7 MB
Load Address: 80008000
Entry Point:  80008000
Verifying Checksum ... OK
Loading Kernel Image ... OK
OK
```

```
Starting kernel ...
```

The above console output is printed by the u-boot boot-loader when starting linux and the message "Starting kernel..." is last message that u-boot will print before starting the kernel. The above problem is typically caused by mis-configuring the tty interface that the linux kernel uses by default for displaying the console messages. The default tty interface used the linux kernel for OMAP devices is configured in the kernel menuconfig. You can check the tty interface being used for an OMAP device by viewing the kernel .config file and seeing which CONFIG_OMAP_LL_DEBUG_UARTx is selected. For example, the OMAP3 beagle-board uses UART3 for displaying the console messages. Hence, when using the OMAP3 beagle-board the following should be found in the .config file:

```
# CONFIG_OMAP_LL_DEBUG_UART1 is not set
# CONFIG_OMAP_LL_DEBUG_UART2 is not set
CONFIG_OMAP_LL_DEBUG_UART3=y
```

The default tty interface used by the linux kernel for OMAP can be configured by starting the linux menuconfig utility (by executing "make menuconfig") and going to "System Type --> TI OMAP Implementations --> Low-level debug console UART".

Problem #2 - No more output is seen on the console after "booting the kernel"[[edit](#)]

Another problem a user may encounter when booting the kernel, is that no further output after "booting the kernel" is seen on the console. For example:

```
## Booting kernel from Legacy Image at 80300000 ...
  Image Name:   Linux-2.6.31
  Image Type:   ARM Linux Kernel Image (uncompressed)
  Data Size:    1750680 Bytes = 1.7 MB
  Load Address: 80008000
  Entry Point:  80008000
  Verifying Checksum ... OK
  Loading Kernel Image ... OK
OK

Starting kernel ...

Uncompressing
Linux.....
..... done, booting the
kernel.
```

This problem can be caused by a number of reasons. The most common reasons are listed below.

Cause #1 - The linux console boot parameter is incorrect[[edit](#)]

If the "console" boot parameter passed by the boot-loader to the kernel is incorrect then the above failure will be observed. This could be as simple as a typographical error. For example, by default the OMAP3 beagle-board displays console messages on the UART3 port and the default configuration of UART3 port is 115200 baud, 8-bit data, no parity and no flow control. Hence, viewing the u-boot boot arguments you should see something like the following:

```
OMAP3 beagleboard.org # printenv bootargs
bootargs=console=ttyS2,115200n8 root=/dev/mmcblk0p2 rw rootwait
```

The console parameter is case-sensitive and so make sure it is written correctly for the board you are using and there are no spaces. For example, "console=ttyS2, 115200n8" would not work.

On Linux kernels version 2.6.36 and newer, use ttyO2 instead of ttyS2 (that's capital-O, not zero).

The best way to avoid such errors is to use a script for your boot parameters. Sample [Teraterm Scripts](#) are available. Similar scripts can be created for Linux [Minicom](#) users.

Cause #2 A - Mis-match between boot-loader and kernel machine numbers[[edit](#)]

Linux does not allow you to boot a kernel built for one hardware platform on some other piece of hardware, even if the underlying processor is the same. There is no reason why you would ever want to! When the kernel starts, one of the first things it does is to check that the machine number passed by the boot-loader matches the machine number that the kernel was built for. If the machine numbers do not match the kernel will not boot. This is a good thing!

You can check by re-building your kernel with CONFIG_DEBUG_LL enabled. To enable this start the linux menuconfig utility (by executing "make menuconfig") and go to "Kernel hacking" and select "Kernel low-level debugging functions". For example, if you were to enable this option and attempt to boot an OMAP3 EVM kernel on an OMAP3 beagle-board the following message would be seen.

```
Error: unrecognized/unsupported machine ID (r1 = 0x0000060a) .
```

```
Available machine support:
```

| ID (hex) | NAME |
|----------|-----------|
| 000005ff | OMAP3 EVM |

```
Please check your kernel config and/or bootloader.
```

Cause #2 B - Mis-match between device tree blob and the actual hardware[[edit](#)]

Newer versions of Linux have adopted FDT (flattened device tree) to describe the Hardware instead of hard coding in the kernel image. This is to enable build once and use across platforms. Using this one can boot same kernel image on many a platforms by providing a device tree blob (compiled from device tree source files - dts) that describe teh targeted hardware. Reference [http://elinux.org/Device_Tree_Reference] and [<https://www.devicetree.org/>]

Care needs to be taken when chosing the device tree blob to be loaded base don the actual Hardware - usually passed via bootargs as "fdtfile=<file name>".

When the dtb doesnt match the HW it sometimes results in Kernel being stuck at "Starting kernel..."
or sometimes might result in a wrong device configuration such as memory size mismatch
or trying to initialize a peripheral at wring clock
or one that doesn't even exist.

Cause #3 - A software bug[\[edit\]](#)

If the previous causes did not solve your problem, then there is a chance that a software change is breaking the kernel for the device you are building for. To get more information on exactly where the kernel is failing, it is recommended that you enable CONFIG_DEBUG_LL in the linux kernel configuration. This may print out more information after "booting the kernel" is seen and may help determine where the kernel is failing. Even if you are unable to make any further progress from here, providing as much information as you can will help others determine where the problem is.

Problem #3 - No console prompt seen after kernel boot[\[edit\]](#)

After the kernel has finished booting you may see a console message such as "Please press Enter to activate this console" and when hitting enter a console prompt appear which may be signified by a character such as "#". An example output is shown below.

```
Sending DHCP requests .<6>eth0: link up, 100Mbps, full-duplex, lpa
0xCDE1
., OK
IP-Config: Got DHCP answer from 255.255.255.255, my address is
128.247.107.23
IP-Config: Complete:
    device=eth0, addr=128.247.107.23, mask=255.255.254.0,
gw=128.247.106.2,
    host=128.247.107.23, domain=am.dhcp.ti.com, nis-domain=(none),
    bootserver=255.255.255.255, rootserver=128.247.107.35, rootpath=
Looking up port of RPC 100003/2 on 128.247.107.35
Looking up port of RPC 100005/1 on 128.247.107.35
VFS: Mounted root (nfs filesystem).
```

```
Freeing init memory: 136K
init started: BusyBox v1.11.1 (2008-10-05 04:40:51 CDT)
starting pid 288, tty : '/etc/init.d/rcS'
```

System initialization...

```
Hostname      : OMAP3EVM
Filesystem    : v1.0.0
```

```
Kernel release : Linux 2.6.22.18-omap3
Kernel version  : #12 Mon Oct 6 01:22:49 CDT 2008
```

```
Mounting /proc      : [SUCCESS]
Mounting /sys       : [SUCCESS]
Mounting /dev       : [SUCCESS]
Mounting /dev/pts   : [SUCCESS]
Enabling hot-plug   : [SUCCESS]
Populating /dev     : [SUCCESS]
Disabling Power mgmt : [SUCCESS]
Turn off LCD after 1 hour : [SUCCESS]
Mounting other filesystems : [SUCCESS]
Starting syslogd    : [SUCCESS]
Starting telnetd    : [SUCCESS]
```

System initialization complete.

Please press Enter to activate this console.

The console message indicates that a UNIX shell has been started and you may enter commands through what is known as the command-line. If you do not see such a message or a prompt appear and you are unable to enter commands via the command-line after the kernel boots, then typically indicates that the start-up scripts in file-system are selecting the wrong tty interface for the console. For example, [this article](#) describes how to create a busybox root file-system for an OMAP3 device. If you refer to [this section](#) on configuring the file-system you will find that a file called /etc/inittab is created with the following contents.

```
::sysinit:/etc/init.d/rcS

# /bin/ash
#
# Start an "askfirst" shell on the serial port
ttyS0:askfirst:-/bin/ash

# Stuff to do when restarting the init process
::restart:/sbin/init

# Stuff to do before rebooting
::ctrlaltdel:/sbin/reboot
::shutdown:/bin/umount -a -r
::shutdown:/sbin/swapoff -a
```

The above file is executed once the kernel has booted and you can see near the top of the script a "askfirst" shell is launch on ttyS0. If you attempted to use this file-system as-is

with a OMAP3 Beagle board that uses ttyS2 for console output then you would not see a console prompt after the kernel boots. This is easily corrected by editing the above file and changing ttyS0 to ttyS2.

Keystone=

{{

1. switchcategory:MultiCore=

- For technical support on MultiCore devices, please post your questions in the [C6000 MultiCore Forum](#)
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the [BIOS Forum](#)

Please post only comments related to the article **Kernel - Common Problems Booting Linux** here.

- For technical support on MultiCore devices, please post your questions in the [C6000 MultiCore Forum](#)
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the [BIOS Forum](#)

Please post only comments related to the article **Kernel - Common Problems Booting Linux** here.



Links



[Amplifiers & Linear Audio](#)
[Broadband RF/IF & Digital](#)

[DLP & MEMS High-Reliability Interface Logic](#)

[Processors](#)

- [ARM Processors](#)
- [Digital Signal Processors \(DSP\)](#)

[Switches & Multiplexers](#)
[Temperature Sensors & Control ICs](#)

| | | | |
|---|--------------------------------------|--|---|
| Radio Clocks & Timers Data Converters | Power Management | <ul style="list-style-type: none"> • Microcontrollers (MCU) • OMAP Applications Processors | Wireless Connectivity |
|---|--------------------------------------|--|---|

Retrieved from "https://processors.wiki.ti.com/index.php?title=Kernel_-_Common_Problems_Booting_Linux&oldid=228621"

Category:

- [Linux](#)

Navigation menu

Personal tools

- [Log in](#)
- [Request account](#)

Namespaces

- [Page](#)
- [Discussion](#)



Variants

Views

- [Read](#)
- [View source](#)
- [View history](#)



More

Search



Navigation

- [Main Page](#)
- [All pages](#)
- [All categories](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

Toolbox

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Page information](#)

- This page was last edited on 28 June 2017, at 07:22.
- Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.

- [Privacy policy](#)
- [About Texas Instruments Wiki](#)
- [Disclaimers](#)
- [Terms of Use](#)

