

# N8x0 support in linux-omap

Current kernel: 2.6.31-rc1-omap1

## Startup plan

- [2/3] find optimal debugging techniques and real HW/qemu split;
- [3/4] make minimal working/debuggable configuration;
- [0] compile domain glossary.

## Debug interfaces

## Subsystems' status

## Debugging in qemu

Need CONFIG\_DEBUG\_INFO=y in kernel configuration for symbolic debugging to work.

## Kernel command line parameters

- initcall\_debug=1 - make kernel print all \_init function calls during kernel\_init;

## Magic numbers

- 0x80000000 - here the kernel is loaded
- 0x80008000 - here we jump after decompression (.text.head that has VA of 0xc0008000 has PA 0x80008000 until MMU is active)
- 0xc0026000 - 'arm-linux-gnu-objdump -x vmlinux' says that .text starts here

## Script for gdb session

```
target remote 127.0.0.1:1234
break *0x80008000
```

To debug compression-related stuff from the very beginning (start, arch/arm/boot/compressed/head.S) till start\_kernel:

```
add-symbol-file ~/ws/osll/omap/20090610/linux-omap-2.6/arch/arm/boot/compressed/vmlinux 0x80000000
```

To debug kernel from stext (arch/arm/kernel/head.S) until MMU is active:

```
add-symbol-file ~/ws/osll/omap/20090610/linux-omap-2.6/vmlinux 0x80026000 -s .text.head 0x80008000
```

To debug kernel from start\_kernel (init/main.c):

```
add-symbol-file ~/ws/osll/omap/20090610/linux-omap-2.6/vmlinux 0xc0026000
```

## "Blank screen" debugging

Whatever happens, ^C breaks into the running kernel. If the screen is blank, dmesg-like log may be viewed through

```
x/10000s log_buf
```

Or even through

```
dump memory kmsg.log log_buf log_buf+10000
```

## Debugging on real HW

### Dealing with retu watchdog

Need to periodically touch it, like this:

```
while true
do
  echo 63 > /sys/devices/platform/retu-watchdog/period
  sleep 1
done
```

From: <http://wiki.osll.ru/> - Open Source & Linux Lab

Permanent link: <http://wiki.osll.ru/doku.php/etc:users:jcmvbkbc:omap-support-pieces?rev=1246785562>

Last update: **2009/07/05 13:19**

