

Mauro Lopez  
LAB 9 Report  
ECEN-449  
Section 502  
November 29, 2018

## Introduction:

In this lab we are going to learn more about built in kernels for Linux. This built in drives will load during the boot in process, so that you can call them as soon as is done the boot in process.

## Procedures:

We start off by first looking at the Linux menu configuration. We do this by typing the following command:

```
$ make ARCH=blackfin menuconfig
```

This will open a window and allow you to look through the configurations of different drives.

Once we have an idea of how different configurations depend on different drives. We can start by implementing the multiply IP from our lab 6.

We begin by first creating a new directory for the our multiply drive and call it “multiplier\_driver”. We then create a Makefile and add the following line to it.

```
Ob-$(CONFIG_MULTIPLIER_DRIVER) += multiplier.o
```

We copy all of our source files for our multiplier from our lab 6 modules directory.

NOTE: avoid overriding the Makefile.

The next step is to also create a Kconfig file and add the following to it.

Config MULTIPLIER\_DRIVER

Tristate “multiplier\_driver”

Depends on ARM

Default y if ARM

Help

Refer to ECEN449TAMU

Once we have those files ready we need to edit the Makefile in the Driver directory as well as the Kconfig file.

Add the following to the makefile:

```
#ECEN 449
```

```
Ob-$(CONFIG_MULTIPLIER_DRIVER) += multiplier_driver/
```

To the Kconfig:

Source “drivers/multiplier\_driver/Kconfig”

Once we have done the above commands we can launch the menuconfig.

We do this by using the same command as before:

```
$ make ARCH=arm menuconfig
```

In this menu we should see our driver set to “built in”.

We then need to recompile our linux source files and generate our uImage.

Once we have compiled and unzipped our zImage file, we can use our BOOT.bin from lab 5 as well as our devicetree.dtb to boot Linux in our ZYBO board.

We follow the same steps from above to add our IR\_DEMOD driver.

## Results:

Once we have successfully added our driver, we should be able to see it as the system is boot it up. Something similar to this:

```
mauro31@lin08-424cvlb:~  
File Edit View Search Terminal Help  
zynq-dr e0002000.ps7-usb: Unable to init USB phy, missing?  
usbcore: registered new interface driver usb-storage  
mousedev: PS/2 mouse device common for all mice  
i2c /dev entries driver  
Xilinx Zynq CpuIdle Driver started  
sdhci: Secure Digital Host Controller Interface driver  
sdhci: Copyright(c) Pierre Ossman  
sdhci-pltfm: SDHCI platform and OF driver helper  
sdhci-arasan e0100000.ps7-sdio: No vmc regulator found  
sdhci-arasan e0100000.ps7-sdio: No vmc regulator found  
mmc0: SDHCI controller on e0100000.ps7-sdio [e0100000.ps7-sdio] using ADMA  
ledtrig-cpu: registered to indicate activity on CPUs  
usbcore: registered new interface driver usbhid  
usbhid: USB HID core driver  
Mapping virtual address...  
Physical Address: 0x43c00000  
Virtual Address: 0x608e0000  
Registered a device with dynamic Major number of 245  
Create a device file for this device with this command:  
'mknod /dev/multiplier c 245 0'.  
TCP: cubic registered  
NET: Registered protocol family 17  
can: controller area network core (rev 20120528 abi 9)  
NET: Registered protocol family 29  
can: raw protocol (rev 20120528)  
can: broadcast manager protocol (rev 20120528 t)  
can: netlink gateway (rev 20130117) max_hops=1  
zynq_pm_ioremap: no compatible node found for 'xlnx,zynq-ddrc-a05'  
zynq_pm_late_init: Unable to map DDR3 IO memory.  
Registering SWP/SWPB emulation handler  
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)  
ALSA device list:  
No soundcards found.  
RAMDISK: gzip image found at block 0  
mmc0: new high speed SDHC card at address aaaa  
mmcblk0: mmc0:aaaa S508G 7.40 GiB  
mmcblk0: p1  
EXT2-fs (ram0): warning: mounting unchecked fs, running e2fsck is recommended  
VFS: Mounted root (ext2 filesystem) on device 1:0.  
devtmpfs: mounted  
Freeing unused kernel memory: 212K (40627000 - 4065c000)  
Starting rcS...  
++ Mounting filesystem  
++ Setting up mdev  
++ Starting telnet daemon  
++ Starting http daemon  
++ Starting ftp daemon  
++ Starting dropbear (ssh) daemon  
random: dropbear urandom read with 1 bits of entropy available  
rcS Complete  
zynq> mount /dev/mmcblk0p1 /mnt
```

We can see our multiplier drive print out statements.

When we run our devtest executable file we should get the following:

```
mauro31@lin08-424cvlb:~$ ./devtest
File Edit View Search Terminal Help
0 X 13 = 0
Result Correct!

Writing 0 to register 0
Writing 14 to register 1
0 X 14 = 0
0 X 14 = 0
Result Correct!

Writing 0 to register 0
Writing 15 to register 1
0 X 15 = 0
0 X 15 = 0
Result Correct!

Writing 0 to register 0
Writing 16 to register 1
0 X 16 = 0
0 X 16 = 0
Result Correct!

Writing 1 to register 0
Writing 0 to register 1
1 X 0 = 0
1 X 0 = 0
Result Correct!

Writing 1 to register 0
Writing 1 to register 1
1 X 1 = 1
1 X 1 = 1
Result Correct!

Writing 1 to register 0
Writing 2 to register 1
1 X 2 = 2
1 X 2 = 2
Result Correct!

Writing 1 to register 0
Writing 3 to register 1
1 X 3 = 3
1 X 3 = 3
Result Correct!

Writing 1 to register 0
Writing 4 to register 1
1 X 4 = 4
1 X 4 = 4
Result Correct!
█
```

Once we implement the IR\_DEMOD driver into linux we should also get the boot in sequence with the following print out statements:

```
Applications Places Terminal Thu 12:26 ● mauro31@lin08-424cvlb:~$ ./devtest
File Edit View Search Terminal Help
Freeing initrd memory: 3608K (5f7aa000 - 5fb30000)
hw perfevents: enabled with armv7_cortex_a9 PMU driver, 7 counters available
futex hash table entries: 512 (order: 3, 32768 bytes)
jffs2: version 2.2. (NAND) (SUMMARY) © 2001-2006 Red Hat, Inc.
msgmni has been set to 1024
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
dma-pl330 f8003000.ps7-dma: Loaded driver for PL330 DMAc-241330
dma-pl330 f8003000.ps7-dma: DBUFF-128x8bytes Num Chans-8 Num Peri-4 Num Events-16
xuartps e0001000.serial: ttyPS0 at MMIO 0xe0001000 (irq = 82, base_baud = 3125000) is a xuartps
console [ttyPS0] enabled
xdevcfg f8007000.ps7-dev-cfg: ioremap 0xf8007000 to 6086a000
[drm] Initialized drm 1.1.0 20060810
brd: module loaded
loop: module loaded
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
ehci-pci: EHCI PCI platform driver
zynq-dr e0002000.ps7-usb: Unable to init USB phy, missing?
usbcore: registered new interface driver usb-storage
mousedev: PS/2 mouse device common for all mice
i2c /dev entries driver
Xilinx Zynq CpuIdle Driver started
sdhci: Secure Digital Host Controller Interface driver
sdhci: Copyright(c) Pierre Ossman
sdhci-pltfm: SDHCI platform and OF driver helper
sdhci-arasan e0100000.ps7-sdio: No vmmc regulator found
sdhci-arasan e0100000.ps7-sdio: No vqmmc regulator found
mmc0: SDHCI controller on e0100000.ps7-sdio [e0100000.ps7-sdio] using ADMA
ledtrig-cpu: registered to indicate activity on CPUs
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
Mapping virtual address...
Physical Address: 0x43c00000
Virtual Address: 0x6080e000
Registered a device with dynamic Major number of 247
Create a device file for this device with this command:
'mknod /dev/irq test c 247 0'.
zynq_pm_ioremap: no compatible node found for 'xlnx,zynq-ddrc-a05'
zynq_pm_late_init: Unable to map DDRc IO memory.
Registering SWP/SWPB emulation handler
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
RAMDISK: gzip image found at block 0
mmc0: new high speed SDHC card at address aaaa
mmcblk0: mmc0:aaaa S508G 7.40 GiB
mmcblk0: p1
EXT2-fs (ram0): warning: mounting unchecked fs, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem) on device 1:0.
devtmpfs: mounted
Freeing unused kernel memory: 172K (4046d000 - 40498000)
Starting rc5...
```

As we add and remove drives from our menuconfig we can see the size of our uImage change. For instance here is the initial size of the uImage with the multiplier\_drive added to it:

```
mauro31@lin08-424cvlb:~/fall18/ecen449/lab4/linux-3.14
File Edit View Search Terminal Help
CC drivers/rpmsg/virtio_rpmsg_bus.mod.o
LD [M] drivers/rpmsg/virtio_rpmsg_bus.ko
CC drivers/usb/gadget/function/usb_f_ss_lb.mod.o
LD [M] drivers/usb/gadget/function/usb_f_ss_lb.ko
CC drivers/usb/gadget/legacy/g_zero.mod.o
LD [M] drivers/usb/gadget/legacy/g_zero.ko
CC drivers/usb/gadget/libcomposite.mod.o
LD [M] drivers/usb/gadget/libcomposite.ko
CC drivers/virtio/virtio.mod.o
LD [M] drivers/virtio/virtio.ko
CC drivers/virtio/virtio_ring.mod.o
LD [M] drivers/virtio/virtio_ring.ko
CC fs/configfs/configfs.mod.o
LD [M] fs/configfs/configfs.ko
CC net/8021q/8021q.mod.o
LD [M] net/8021q/8021q.ko
CC net/ipv4/ip_tunnel.mod.o
LD [M] net/ipv4/ip_tunnel.ko
CC net/ipv4/ipip.mod.o
LD [M] net/ipv4/ipip.ko
CC net/ipv4/tunnel4.mod.o
LD [M] net/ipv4/tunnel4.ko
CC net/ipv6/ipv6.mod.o
LD [M] net/ipv6/ipv6.ko
CC net/ipv6/sit.mod.o
LD [M] net/ipv6/sit.ko
CC net/ipv6/xfrm6_mode_beet.mod.o
LD [M] net/ipv6/xfrm6_mode_beet.ko
CC net/ipv6/xfrm6_mode_transport.mod.o
LD [M] net/ipv6/xfrm6_mode_transport.ko
CC net/ipv6/xfrm6_mode_tunnel.mod.o
LD [M] net/ipv6/xfrm6_mode_tunnel.ko
[mauro31@lin08-424cvlb linux-3.14]$ make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi- UIMAGE_LOADADDR=0x8000 uImage
CHK include/config/kernel.release
CHK include/generated/uapi/linux/version.h
CHK include/generated/utsrelease.h
make[1]: 'include/generated/mach-types.h' is up to date.
CALL scripts/checksyscalls.sh
CHK include/generated/compile.h
CHK kernel/config.data.h
^[[A Kernel: arch/arm/boot/Image is ready
Kernel: arch/arm/boot/zImage is ready
UIMAGE arch/arm/boot/uImage
Image Name: Linux-3.18.0-xilinx
Created: Thu Nov 29 11:14:37 2018
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 3449872 Bytes = 3368.23 kB = 3.29 MB
Load Address: 00000000
Entry Point: 00000000
Image arch/arm/boot/uImage is ready
[mauro31@lin08-424cvlb linux-3.14]$ make ARCH=arm menuconfig
```

Once we remove the networking support, Device Drivers/Multimedia support, and Device Drivers/Soundcard support, we get the following. Note: I removed the multiplier\_drive and added the IR\_DEMOD driver instead, the difference still visible.

```
mauro31@lin08-424cvlb:~/fall18/ecen449/lab4/linux-3.14
File Edit View Search Terminal Help
Kernel: arch/arm/boot/Image is ready
GZIP arch/arm/boot/compressed/piggy.gzip
AS arch/arm/boot/compressed/piggy.gzip.o
LD arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
Building modules, stage 2.
MODPOST 13 modules
CC crypto/ansi_cprng.mod.o
LD [M] crypto/ansi_cprng.ko
CC crypto/krng.mod.o
LD [M] crypto/krng.ko
CC crypto/rng.mod.o
LD [M] crypto/rng.ko
CC drivers/remoteproc/mb_remoteproc.mod.o
LD [M] drivers/remoteproc/mb_remoteproc.ko
CC drivers/remoteproc/remoteproc.mod.o
LD [M] drivers/remoteproc/remoteproc.ko
CC drivers/remoteproc/zynq_remoteproc.mod.o
LD [M] drivers/remoteproc/zynq_remoteproc.ko
CC drivers/rpmsg/virtio_rpmsg_bus.mod.o
LD [M] drivers/rpmsg/virtio_rpmsg_bus.ko
CC drivers/usb/gadget/function/usb_f_ss_lb.mod.o
LD [M] drivers/usb/gadget/function/usb_f_ss_lb.ko
CC drivers/usb/gadget/legacy/g_zero.mod.o
LD [M] drivers/usb/gadget/legacy/g_zero.ko
CC drivers/usb/gadget/libcomposite.mod.o
LD [M] drivers/usb/gadget/libcomposite.ko
CC drivers/virtio/virtio.mod.o
LD [M] drivers/virtio/virtio.ko
CC drivers/virtio/virtio_ring.mod.o
LD [M] drivers/virtio/virtio_ring.ko
[mauro31@lin08-424cvlb linux-3.14]$ make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi- UIMAGE_LOADADDR=0x8000 uImage
CHK include/config/kernel.release
CHK include/generated/uapi/linux/version.h
CHK include/generated/utsrelease.h
make[1]: 'include/generated/mach-types.h' is up to date.
CALL scripts/checksyscalls.sh
CHK include/generated/compile.h
CHK kernel/config.data.h
Kernel: arch/arm/boot/Image is ready
Kernel: arch/arm/boot/zImage is ready
UIMAGE arch/arm/boot/uImage
Image Name: Linux-3.18.0-xilinx
Created: Thu Nov 29 11:51:19 2018
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 2501976 Bytes = 2443.34 kB = 2.39 MB
Load Address: 00000000
Entry Point: 00000000
Image arch/arm/boot/uImage is ready
[mauro31@lin08-424cvlb linux-3.14]$
```

## Conclusion:

Overall the lab was helpful in teaching another way to implement drives directory into Linux. I also learned more about configuring the menu and how it affects the size of the uImage. We also saw that, built in drives also can run executable files such as the multiplier.

## Question:

one of the advantage of loadable kernel modules is that you can add them externally. You can put the drive into the sd card without having to recompile the uImage. With a built-in module you can have the program initialize at the beginning of the boot in system. Meaning there is less room for the program to have errors in managing the memory.

## FEEDBACK:

Overall, I think the labs are helpful. They do help learn the material and their design very well with very little errors. Even when there are errors on the lab manuals, the TA is very helpful in providing aid to students. However, I do think that the time frame provided to demo and complete the labs is not the best. For instance, students should be allow to demo their previous labs after

the due date with a percentage taken off. The reason for this is that it encourages students to at least finish the labs and learn what the lab intends them to learn. Or there should be an option that if a student didn't finish the lab, a solution manual should be provided so that the students that didn't complete the lab can at least work on the next lab.