Lab 3

Objective

To implement a full embedded Linux boot process using QEMU with:

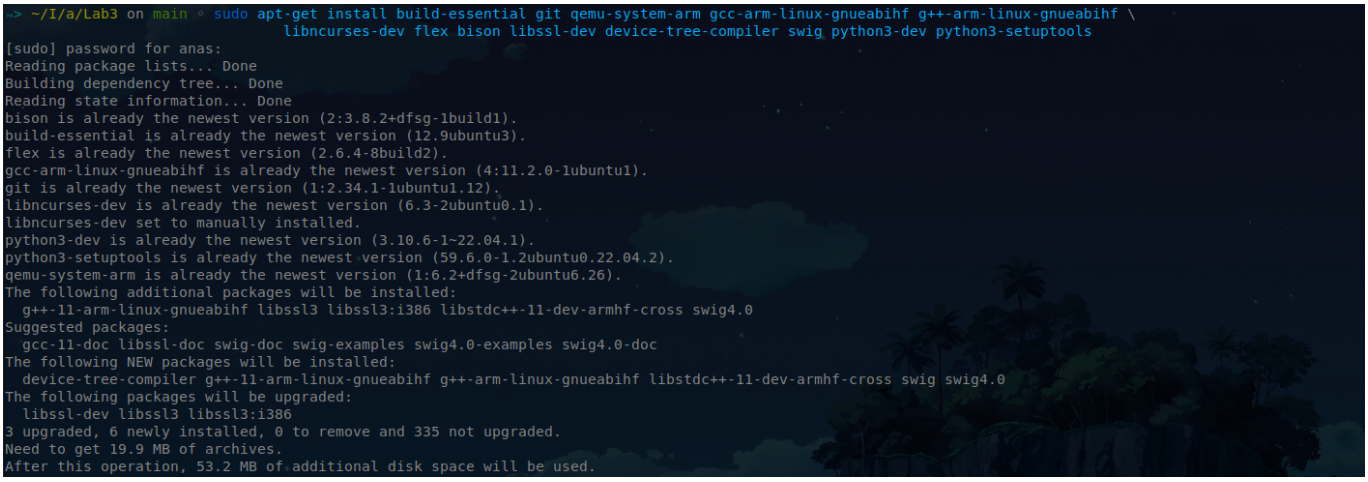
- U-Boot (bootloader)
- Linux kernel compiled for ARM
- Initramfs with custom `/init` script
- Root filesystem (`rootfs.ext4`)

Expected Boot Flow:

U-Boot → Kernel → Initramfs → switch_root → Rootfs

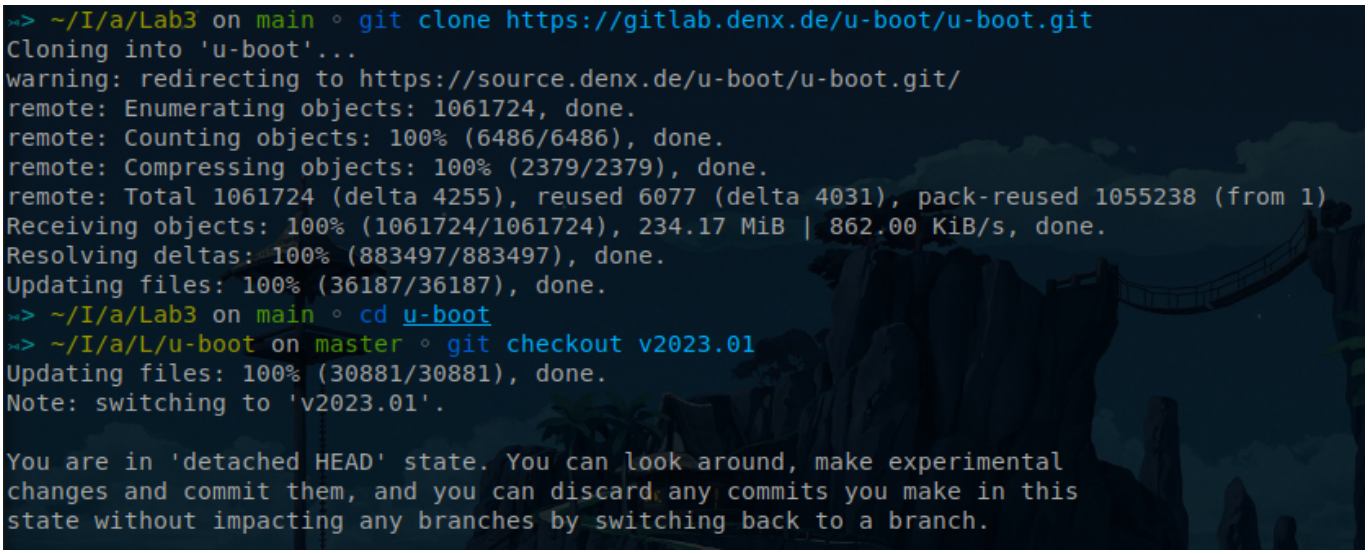
Environment Setup

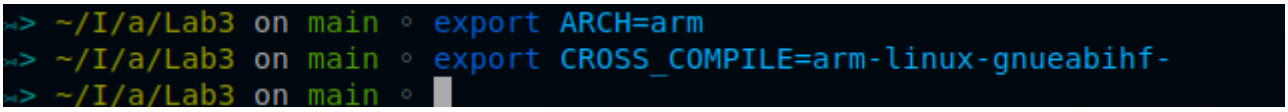
Installed required tools:

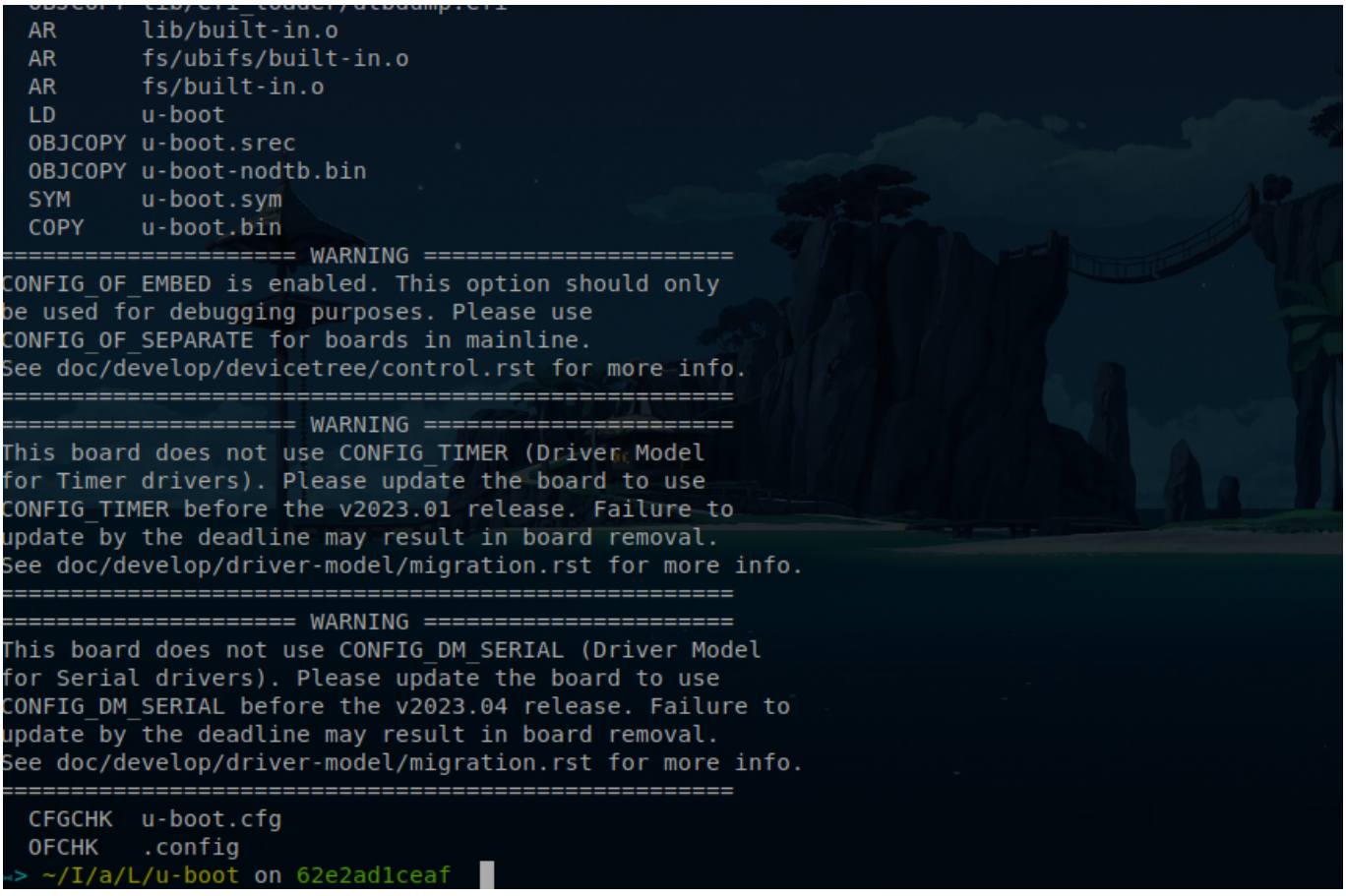


Building U-Boot

```
git clone https://github.com/u-boot/u-boot.git
cd u-boot
git checkout v2022.01

export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabihf-
make vexpress_ca9x4_defconfig
make
```







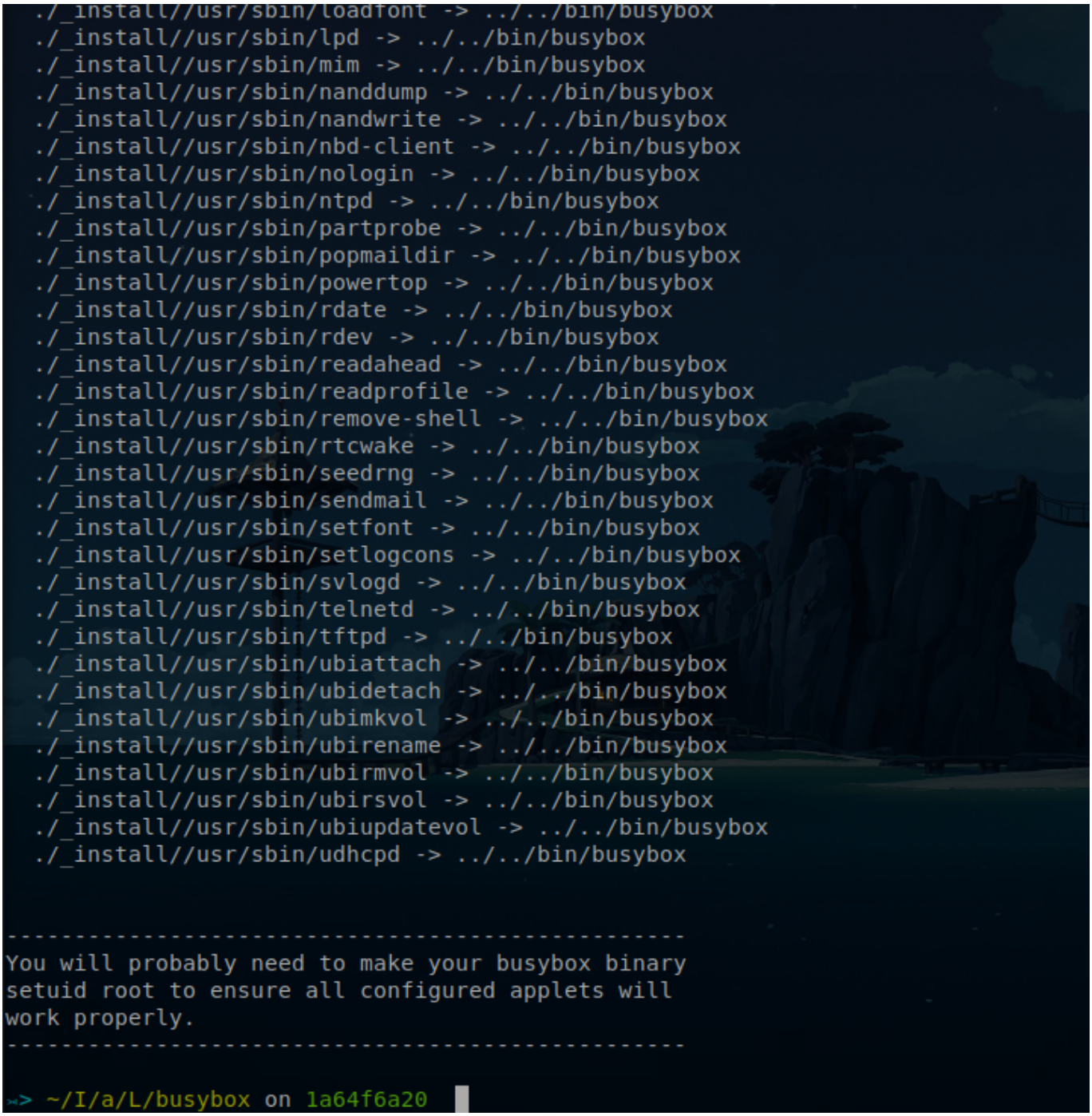
Building Linux Kernel

```
git clone --depth=1
https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git
cd linux

export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabihf-
make vexpress_defconfig
make zImage
```

```
make modules
make dtbs
```

Creating Initramfs with BusyBox

```
wget https://busybox.net/downloads/busybox-1.36.0.tar.bz2
tar -xvf busybox-1.36.0.tar.bz2
cd busybox-1.36.0

export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabihf-
make defconfig
make
make install
```

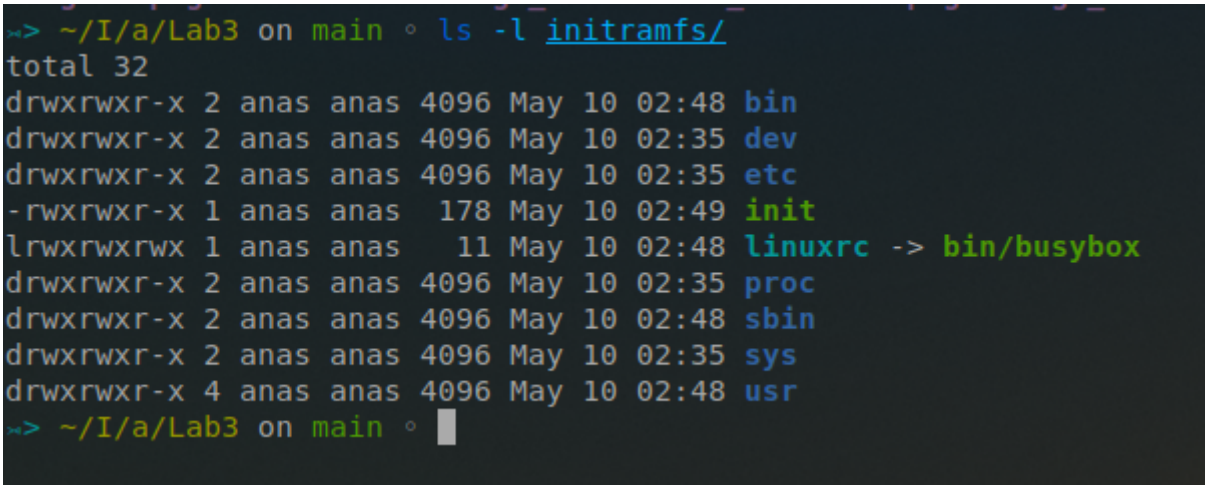


Wrote a `/init` script inside `initramfs/`:

```
#!/bin/sh
mount -t proc none /proc
mount -t sysfs none /sys
mount -t devtmpfs none /dev

mkdir /newroot
mount -t ext4 /dev/mmcblk0p2 /newroot
exec switch_root /newroot /sbin/init
```

Packed initramfs:

```
cd initramfs
find . -print0 | cpio --null -ov --format=newc | gzip -9 >
../initramfs.cpio.gz
```



Root Filesystem (`rootfs.ext4`)

```
mkdir -p rootfs/{bin,sbin,etc,proc,sys,usr/{bin,sbin},dev,tmp,home}
cp -a busybox-1.36.0/_install/* rootfs/
sudo mknod -m 666 rootfs/dev/console c 5 1
sudo mknod -m 666 rootfs/dev/null   c 1 3
```

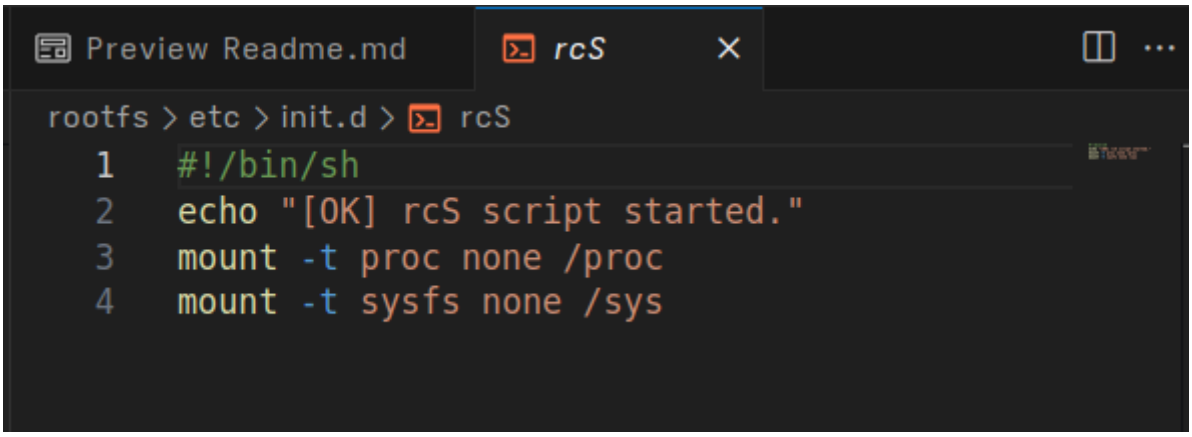
Added:

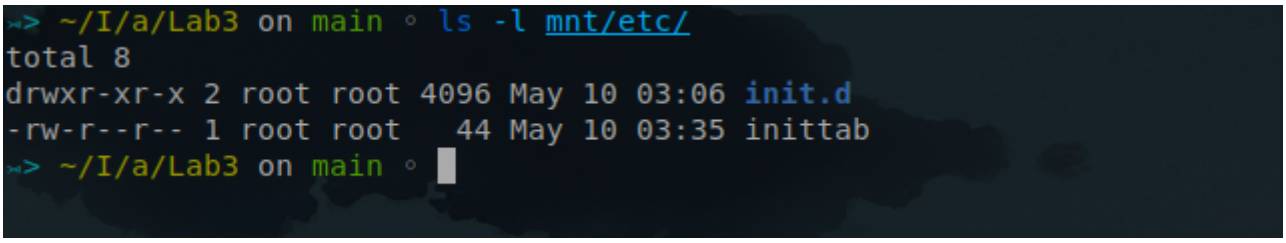
`/etc/inittab`

```
::sysinit:/etc/init.d/rcS
::respawn:/bin/sh
```

`/etc/init.d/rcS`

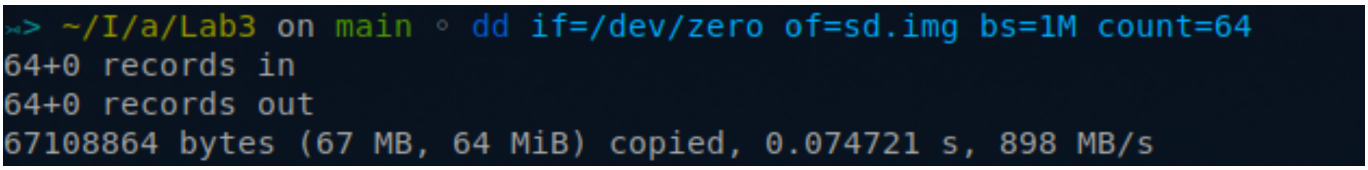
```
#!/bin/sh
echo "[OK] rcS script started."
mount -t proc none /proc
mount -t sysfs none /sys
```





Creating SD Image

```
dd if=/dev/zero of=sd.img bs=1M count=64
fdisk sd.img
# o, n p 1 +16M, n p 2 (default), t 1 c, w
```




```
LOOP=$(sudo losetup --find --show --partscan sd.img)
sudo mkfs.vfat -n BOOT ${LOOP}p1
sudo mkfs.ext4 -L ROOT ${LOOP}p2
```

Filled Boot Partition:

```
mkimage -A arm -T ramdisk -C gzip -n "Initramfs" -d initramfs.cpio.gz
uInitrd

sudo mount ${LOOP}p1 mnt
sudo cp linux/arch/arm/boot/zImage mnt/
sudo cp linux/arch/arm/boot/dts/arm/vexpress-v2p-ca9.dtb mnt/vexpress.dtb
sudo cp uInitrd mnt/
sudo umount mnt
```

```
~> ~/I/a/Lab3 on main ◦ ls -lh mnt/
total 6.8M
-rwxr-xr-x 1 root root 1.1M May 10 03:42 uInitrd
-rwxr-xr-x 1 root root  14K May 10 03:42 vexpress.dtb
-rwxr-xr-x 1 root root 5.7M May 10 03:42 zImage
~> ~/I/a/Lab3 on main ◦
```

Filled Rootfs Partition:

```
sudo mount ${LOOP}p2 mnt
sudo cp -a rootfs/* mnt/
sudo umount mnt
sudo losetup -d "$LOOP"
```

```
~> ~/I/a/Lab3 on main ◦ sudo cp linux/arch/arm/boot/dts/arm/vexpress-v2p-ca9.dtb /tmp/sd_mount/
~> ~/I/a/Lab3 on main ◦ sudo cp busybox/initramfs.cpio.gz /tmp/sd_mount/
~> ~/I/a/Lab3 on main ◦ ls
busybox  linux  sd.img  task.txt  u-boot
~> ~/I/a/Lab3 on main ◦ sudo umount /tmp/sd_mount
~> ~/I/a/Lab3 on main ◦ sudo losetup -d /dev/loop25
~> ~/I/a/Lab3 on main ◦
```

Boot in QEMU with U-Boot

```
qemu-system-arm -M vexpress-a9 \
  -m 512M \
  -kernel u-boot/u-boot \
  -drive file=sd.img,format=raw,if=sd \
  -nographic
```

```
 ~/I/a/Lab3 on main ◦ qemu-system-arm -M vexpress-a9 \
                          -m 512M \
                          -kernel u-boot/u-boot \
                          -drive file=sd.img,format=raw,if=sd \
                          -nographic
```

```
cp [.b, .w, .l, .q] source target count
Wrong Image Format for bootm command
ERROR: can't get kernel image!
=> fatload mmc 0:1 0x60000000 zImage
5940760 bytes read in 1058 ms (5.4 MiB/s)
=> fatload mmc 0:1 0x61000000 uInitrd
1080542 bytes read in 179 ms (5.8 MiB/s)
=> fatload mmc 0:1 0x62000000 vexpress.dtb
14329 bytes read in 12 ms (1.1 MiB/s)
=> setenv bootargs "console=ttyAMA0 loglevel=3 init=/init"
=> bootz 0x60000000 0x61000000 0x62000000
Kernel image @ 0x60000000 [ 0x000000 - 0x5aa618 ]
## Loading init Ramdisk from Legacy Image at 61000000 ...
   Image Name:   Initramfs
   Image Type:   ARM Linux RAMDisk Image (gzip compressed)
   Data Size:    1080478 Bytes = 1 MiB
   Load Address: 00000000
   Entry Point:  00000000
   Verifying Checksum ... OK
## Flattened Device Tree blob at 62000000
   Booting using the fdt blob at 0x62000000
```

Final Result: Booted Root Shell

```
   Loading Ramdisk to 7fa1f000, end 7fb26c9e ... OK
   Loading Device Tree to 7fa18000, end 7fa1e7f8 ... OK

Starting kernel ...

GIC CPU mask not found - kernel will fail to boot.
GIC CPU mask not found - kernel will fail to boot.
[OK] rcS script started.
/bin/sh: can't access tty; job control turned off
~ # ls
bin         etc         linuxrc     proc        sys         usr
dev         home        lost+found  sbin        tmp
~ #
```

THANK YOU for this course! We learned alot of cool things.