Command Reference for Custom Embedded Linux OS Build

1. Install Required Packages (One Time)

```
sudo apt update
sudo apt install -y \
git build-essential gcc-arm-linux-gnueabihf \
qemu-system-arm bc bison flex \
libssl-dev libncurses5-dev
```

Verify cross compiler:

```
arm-linux-gnueabihf-gcc --version
```

2. Create Project Directory

```
cd ~/Documents
mkdir -p custom_os/Raspberry_Pi_2B
cd custom_os/Raspberry_Pi_2B
```

3. Clone Linux Kernel Source (Raspberry Pi)

```
git clone --depth=1 https://github.com/raspberrypi/linux.git
cd linux
```

4. Set Cross Compilation Environment

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

5. Configure Kernel for Raspberry Pi 2B

```
make bcm2709_defconfig
make menuconfig
```

(Here you enable: initramfs, devtmpfs, PL011 serial)

6. Build Kernel and Device Trees

```
make -j$(nproc) zImage dtbs
```

Verify output:

```
ls arch/arm/boot/zImage
ls arch/arm/boot/dts/broadcom/bcm2709-rpi-2-b.dtb
```

7. Go Back to Project Root

```
cd ..
```

8. Clone BusyBox (Stable Source)

```
git clone https://github.com/mirror/busybox.git
cd busybox
```

9. Configure BusyBox

```
make defconfig
make menuconfig
```

Enable:

```
Settings → Build static binary (no shared libs)
```

Disable:

```
SHA1 / SHA256 hardware acceleration
```

10. Build and Install BusyBox

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j$(nproc)
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- install
```

BusyBox installs into:

```
busybox/_install/
```

11. Create Root Filesystem

```
cd ..
mkdir rootfs
cp -a busybox/_install/* rootfs/
```

Create required directories:

```
cd rootfs
mkdir -p proc sys dev etc
```

12. Create init Script

```
nano init
```

Content:

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

echo "Welcome ur_name"
exec /bin/sh
```

Make it executable:

```
chmod +x init
```

13. Create initramfs Image

```
cd ..
find rootfs | cpio -o -H newc > rootfs.cpio
```

Verify:

```
file rootfs.cpio
```

Expected:

```
ASCII cpio archive (SVR4 with no CRC)
```

14. Boot Using QEMU (Final Correct Command)

```
qemu-system-arm \
-M raspi2b \
-kernel linux/arch/arm/boot/zImage \
-dtb linux/arch/arm/boot/dts/broadcom/bcm2709-rpi-2-b.dtb \
-initrd rootfs.cpio \
-append "console=ttyAMA0 rdinit=/init" \
-nographic
```

Expected Output

```
Welcome ur_name
/ #
```

Custom Embedded Linux OS booted successfully