

# Steps for Installing Kernel to Compile on Raspberry Pi

This document provides a comprehensive guide on how to install and compile the kernel for Raspberry Pi. It covers the necessary steps from setting up the environment, cloning the kernel repositories, compiling the kernel, transferring files to the SD card, and loading kernel modules. This guide is intended for users who are familiar with Linux command-line operations and have basic knowledge of Raspberry Pi hardware.

## Steps to Install Kernel

1. SSH into Raspberry Pi	<pre>ssh pi@192.168.0.221</pre>
2. Clone the Kernel Repositories	<pre>git clone --branch rpi-5.10.y https://github.com/raspberrypi/linux.git git clone --branch rpi-6.6.y https://github.com/raspberrypi/linux.git</pre>
3. Install Dependencies	<pre>sudo apt install libncurses5-dev libncursesw5-dev pkg-config</pre>
4. Navigate to the Linux Directory	<pre>cd linux</pre>
5. Configure and Compile U-Boot	<pre>make rpi_2_defconfig make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-</pre>
6. Compile the Kernel	<pre>make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- bcm2709_defconfig make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- zImage -j\$(nproc) make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- modules -j\$(nproc) make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- dtbs -j\$(nproc)</pre>
7. Transfer Files to SD Card	<p><b>Important:</b> This step only works when the SD card is detected. It will not work in WSL Ubuntu as it does not detect the SD card.</p> <pre>cp arch/arm/boot/zImage /media/kunal/bootfs/kernel7.img cp arch/arm/boot/dts/*.dtb /media/kunal/bootfs/ # version 5.10 sudo cp arch/arm/boot/dts/broadcom/*.dtb /media/kunal/bootfs/ # version rpi-6.6.y cp arch/arm/boot/dts/overlays/*.dtb* /media/kunal/bootfs/overlays/</pre>
8. Install Kernel Modules	<pre>make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- INSTALL_MOD_PATH=modules_out modules_install cp -r modules_out/lib/modules/5.10.110-v7+ /media/kunal/rootfs/lib/modules/ cp -r modules_out/lib/modules/6.6.72-v7+ /media/kunal/rootfs/lib/modules/</pre>
9. Sync and Unmount	<pre>sync sudo umount /media/kunal/bootfs sudo umount /media/kunal/rootfs</pre>
10. Transfer Kernel Module to Raspberry Pi	<pre>scp usb_driver.ko pi@192.168.0.219:/home/pi/</pre>
11. Load the Kernel Module	<pre>sudo insmod usb_driver.ko lsmod   grep usb_driver dmesg   tail # Check logs for driver loading status</pre>
12. Check File Installation Time	<pre>stat /lib/modules/\$(uname -r)/kernel/drivers/usb/usb_driverbhai.ko</pre>
13. Check System Information	<pre>uname -m # Check architecture (32-bit or 64-bit) cat /proc/cpuinfo # Display CPU information</pre>

## Additional Information

- **BCM2835** is the Broadcom chip used in the older Raspberry Pi 2 model B rev v1.1 models.
- **BCM2836** details:
  - Architecture: ARM Cortex-A7 quad-core processor.
  - Cores: 4 cores, running at 900 MHz.
  - Type: 32-bit CPU.
  - GPU: VideoCore IV.

## U-Boot Configuration

1. Compile U-Boot	<pre>make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-</pre>
2. Copy U-Boot to Boot Directory	<pre>sudo cp /home/pi/u-boot.bin /boot/firmware/kernel7.img ls -lh /boot/firmware/kernel7.img # Verify copy sudo reboot sudo screen /dev/ttyUSB0 115200 # Check using UART</pre>
3. U-Boot Using TFTP	<pre>setenv serverip 192.168.0.133 setenv ipaddr 192.168.0.219 saveenv tftp 0x10000000 u-boot.bin go 0x10000000 mmc @7e202000: 0 EMMC mmc write 0x10000000 0 0x2000</pre>
4. Create UImage	<pre>mkimage -A arm -O linux -T kernel -C none -a 0x00008000 -e 0x00008000 -n "Linux Kernel" -d arch/arm/boot/zImage uImage bootm 0x01000000</pre>
5. Set Environment Variables for Booting	<pre>setenv kernel_addr_r 0x10000000 # Address for kernel setenv fdt_addr_r 0x20000000 # Address for device tree setenv ramdisk_addr_r 0x21000000 # Address for ramdisk (if needed)</pre> <p>tftpboot \${kernel_addr_r} zImage # Load kernel via TFTP tftpboot \${fdt_addr_r} bcm2709-rpi-2-b.dtb # Load DTB bootz \${kernel_addr_r} - \${fdt_addr_r} # Boot with DTB</p>
6. NFS Boot Configuration	<pre>setenv serverip 192.168.0.133 setenv ipaddr 192.168.0.219 setenv rootpath /srv/nfs/rpi-rootfs setenv bootargs root=/dev/nfs nfsroot=\${serverip}:\${rootpath},vers=3 rw ip=dhcp console=ttyAMA0,115200 setenv bootcmd 'tftpboot 0x08000000 zImage; tftpboot 0x09000000 bcm2836-rpi-2-b.dtb; bootz 0x08000000 - 0x26000000' saveenv boot</pre>

This guide should help you successfully compile and install the kernel on your Raspberry Pi. Make sure to follow each step carefully and verify your configurations.