

Linux embarcado

Jefferson Alves

Kernel – Arquitetura ARM
Faculdade de Tecnologia SENAI Anchieta
São Paulo 21, dezembro

1. Configurar/verificar ambiente de desenvolvimento

1.1. Editar o arquivo “~/.bashrc” ativando a PATH do toolchain conforme a figura abaixo. Adicionar as variáveis de ambiente que será utilizado na compilação do



```
alias pip-u-all='pip freeze --local > /tmp/env_p && pip install -U -r /tmp/env_p'
alias gtracker-sync='rsync -avz $GTRACKER_HOME jefferson@$DEPLOY_GTRACKER_PATH'

#export PATH=$PATH:$HOME/toolchains/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian/bin
export PATH=$PATH:$HOME/toolchains/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian-x64/bin
#export PATH=$PATH:$HOME/toolchains/linaro/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi-hf/bin
#export PATH=$PATH:$HOME/toolchains/x-tools/armv8-rpi3-linux-gnueabi-hf/bin

export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabi-hf-
export SENAI=/home/jefferson/build
```

Kernel

2. Código fonte

2.1. Realizar a clonagem(cópia) do código fonte do kernel em uma workspace local. Para isso utilizar o comando no local onde se deseja fazer a clonagem:

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



```
jefferson@nb-acer: /media/jefferson/Dados/workspace$ git clone --depth=1 https://github.com/raspberrypi/linux
Cloning into 'linux'...
remote: Counting objects: 60284, done.
remote: Compressing objects: 100% (56399/56399), done.
Receiving objects: 6% (4036/60284), 4.52 MiB | 230.00 KiB/s
```

Periodicamente usar o comando abaixo, para manter o código atualizado com o repositório.

```
git pull
```

3. Configuração da compilação

3.1. Entrar no diretório onde foi feita a clonagem

```
cd Linux
```

3.2. Definir a configuração inicial para Raspberry PI 0W

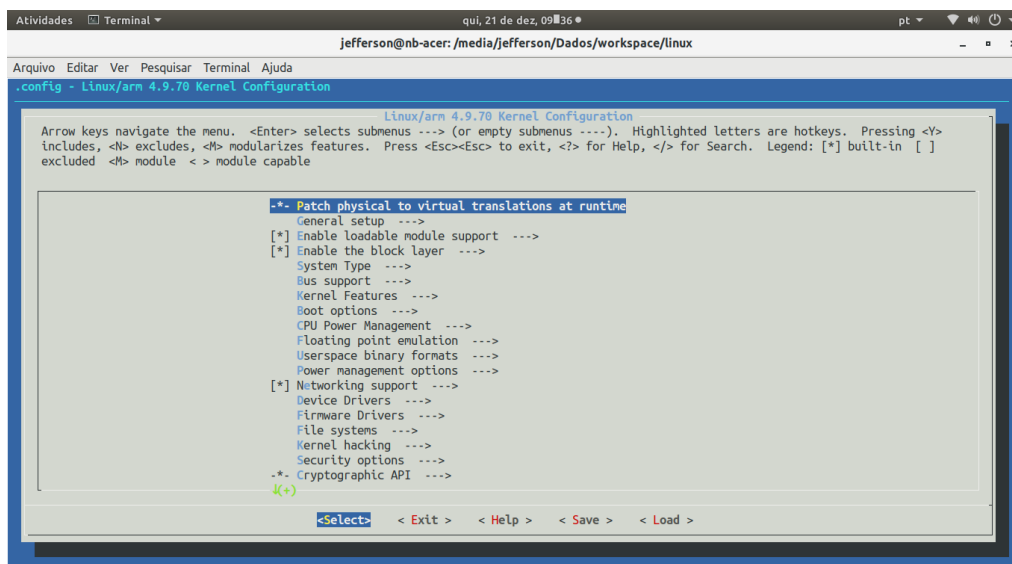
```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-  
bcmrpi_defconfig
```



```
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$ make bcmrpi_defconfig  
HOSTCC scripts/basic/fixdep  
HOSTCC scripts/kconfig/conf.o  
SHIPPED scripts/kconfig/zconf.tab.c  
SHIPPED scripts/kconfig/zconf.lex.c  
SHIPPED scripts/kconfig/zconf.hash.c  
HOSTCC scripts/kconfig/zconf.tab.o  
HOSTLD scripts/kconfig/conf  
#  
# configuration written to .config  
#  
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$
```

3.3. Menu de configuração

Quando necessário utilizar o comando `make menuconfig` para ter acesso as configurações e alterá-las conforme a necessidade.



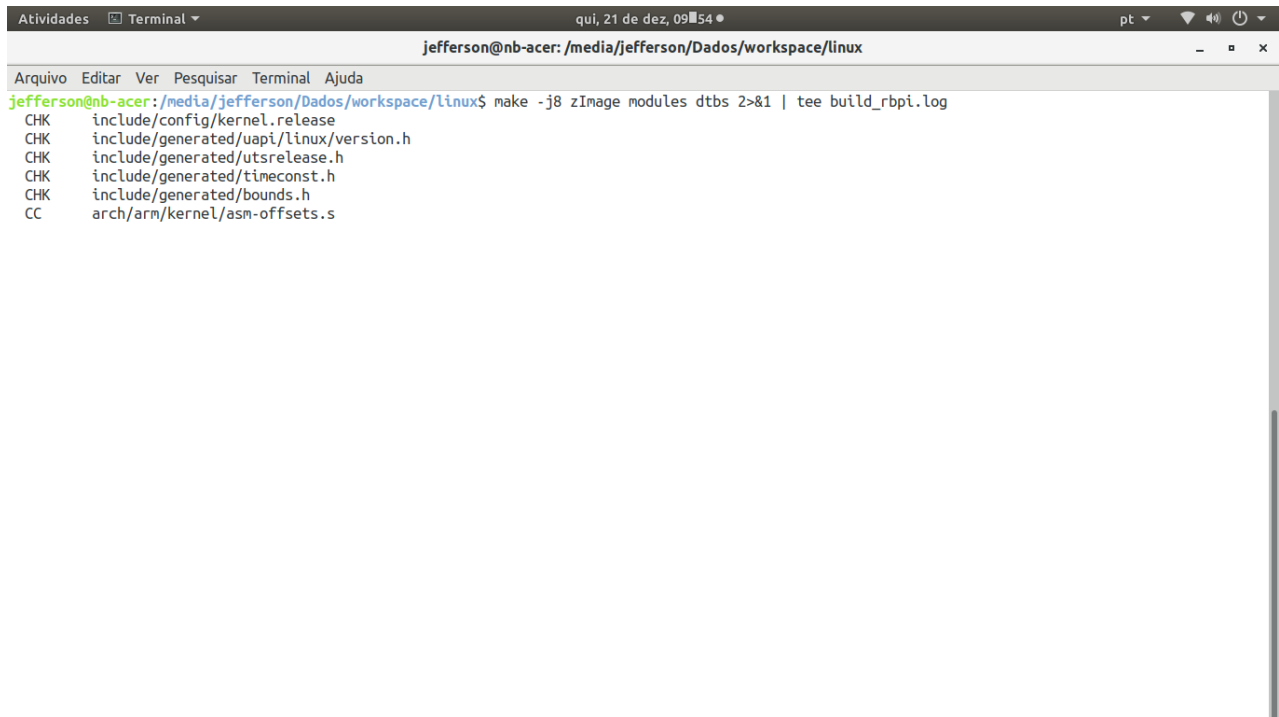
```
.config - Linux/arm 4.9.70 Kernel Configuration  
Linux/arm 4.9.70 Kernel Configuration  
Arrow keys navigate the menu. <Enter> selects submenus --- (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y>  
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]  
excluded <M> module < > module capable  
[*] Patch physical to virtual translations at runtime  
General setup ---  
[*] Enable loadable module support ---  
[*] Enable the block layer ---  
System Type ---  
Bus support ---  
Kernel Features ---  
Boot options ---  
CPU Power Management ---  
Floating point emulation ---  
Userspace binary formats ---  
Power management options ---  
[*] Networking support ---  
Device Drivers ---  
Firmware Drivers ---  
File systems ---  
Kernel hacking ---  
Security options ---  
[*] Cryptographic API ---  
4(+)  
<Select> < Exit > < Help > < Save > < Load >
```

4. Compilação

Para compilar basta usar o comando abaixo

```
make -j8 zImage modules dtbs 2>&1 | tee build.log
```

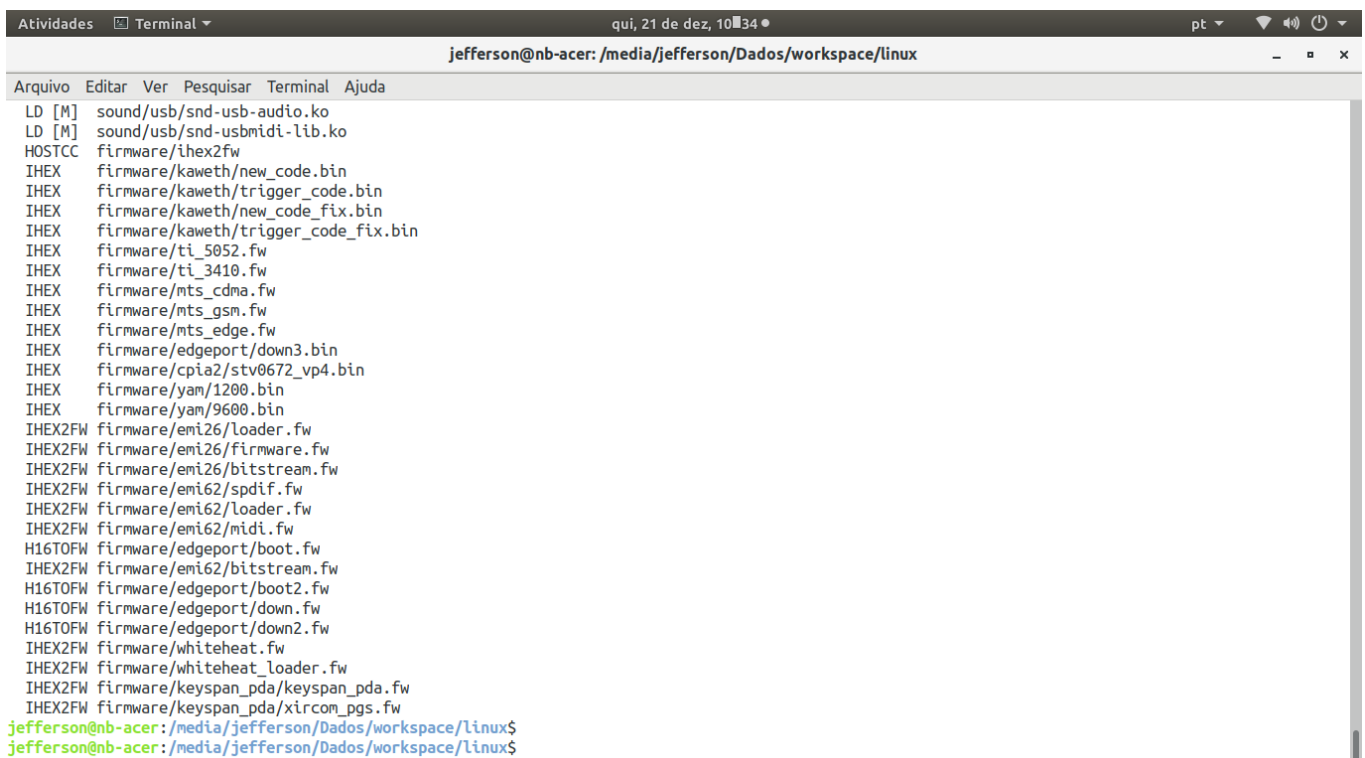
O comando tee irá criar um arquivo com a saída da compilação e mostrar na tela simultaneamente.



A terminal window titled "jefferson@nb-acer: /media/jefferson/Dados/workspace/linux" showing the execution of the command `make -j8 zImage modules dtbs 2>&1 | tee build_rbp.log`. The output shows several files being checked (CHK) and compiled (CC):

```
jefferson@nb-acer:/media/jefferson/Dados/workspace/linux$ make -j8 zImage modules dtbs 2>&1 | tee build_rbp.log
CHK      include/config/kernel.release
CHK      include/generated/uapi/linux/version.h
CHK      include/generated/utsrelease.h
CHK      include/generated/timeconst.h
CHK      include/generated/bounds.h
CC      arch/arm/kernel/asm-offsets.s
```

Fim da compilação



A terminal window titled "jefferson@nb-acer: /media/jefferson/Dados/workspace/linux" showing the completion of the compilation process. The output lists numerous files being linked (LD), host compiled (HOSTCC), and hex2fw converted (IHEX2FW), including various kernel modules and firmware files:

```
LD [M]    sound/usb/snd-usb-audio.ko
LD [M]    sound/usb/snd-usbmidi-lib.ko
HOSTCC    firmware/ihex2fw
IHEX      firmware/kaweth/new_code.bin
IHEX      firmware/kaweth/trigger_code.bin
IHEX      firmware/kaweth/new_code_fix.bin
IHEX      firmware/kaweth/trigger_code_fix.bin
IHEX      firmware/ti_5052.fw
IHEX      firmware/ti_3410.fw
IHEX      firmware/mts_cdma.fw
IHEX      firmware/mts_gsm.fw
IHEX      firmware/mts_gsm.fw
IHEX      firmware/edgeport/down3.bin
IHEX      firmware/cpia2/stv0672_vp4.bin
IHEX      firmware/yam/1200.bin
IHEX      firmware/yam/9600.bin
IHEX2FW   firmware/emi26/loader.fw
IHEX2FW   firmware/emi26/firmware.fw
IHEX2FW   firmware/emi26/bitstream.fw
IHEX2FW   firmware/emi62/spdif.fw
IHEX2FW   firmware/emi62/loader.fw
IHEX2FW   firmware/emi62/midi.fw
H16TOFW   firmware/edgeport/boot.fw
IHEX2FW   firmware/emi62/bitstream.fw
H16TOFW   firmware/edgeport/boot2.fw
H16TOFW   firmware/edgeport/down.fw
H16TOFW   firmware/edgeport/down2.fw
IHEX2FW   firmware/whiteheat.fw
IHEX2FW   firmware/whiteheat_loader.fw
IHEX2FW   firmware/keysan_pda/keysan_pda.fw
IHEX2FW   firmware/keysan_pda/xircom_pgs.fw
jefferson@nb-acer:/media/jefferson/Dados/workspace/linux$
jefferson@nb-acer:/media/jefferson/Dados/workspace/linux$
```

5. Preparação do SD-Card

5.1. Iremos atualizar apenas o Kernel e os módulos. Por isso iremos utilizar uma imagem de uma versão anterior para copiar os demais.

```
unzip -p 2017-11-29-raspbian-stretch.zip | sudo dd  
of=/dev/sdX bs=4M conv=fsync
```

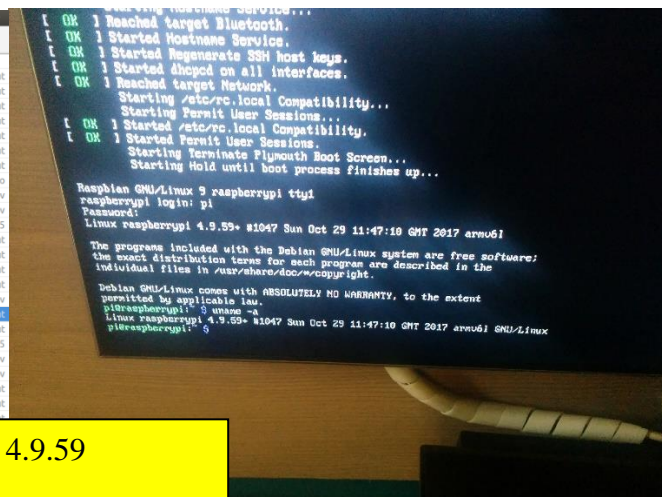
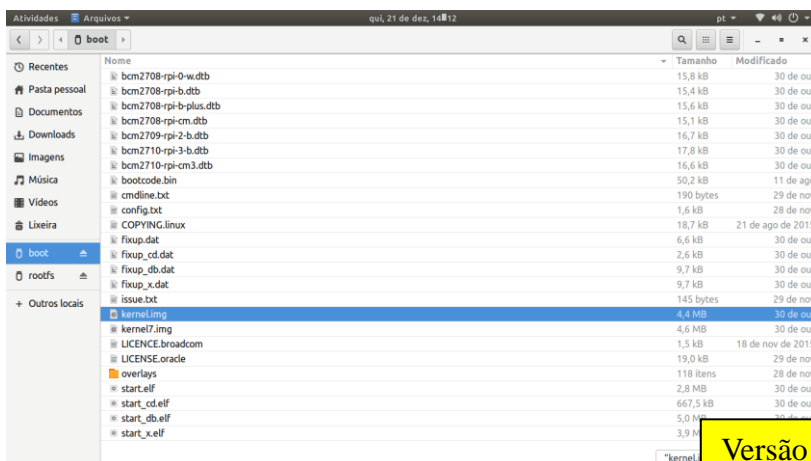
5.2. Executando o commando `lsblk` é possível verificar as partições boot e rootfs

```
jefferson@mb-acer:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0      0 83,7M  1 loop /snap/core/3440
loop1       7:1      0 144M   1 loop /snap/brackets/21
loop2       7:2      0 144,2M 1 loop /snap/brackets/46
loop3       7:3      0 83,8M  1 loop /snap/core/3604
sda          8:0      0 931,5G  0 disk
--sdal       8:1      0 100M   0 part
--sda2       8:2      0 294,2G  0 part
--sda3       8:3      0 472M   0 part
--sda4       8:4      0 1K      0 part
--sda5       8:5      0 536,2G  0 part /media/jefferson/Dados
--sda6       8:6      0 100,6G  0 part /
sr0         11:0     1 1024M  0 rom

jefferson@mb-acer:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0      0 83,7M  1 loop /snap/core/3440
loop1       7:1      0 144M   1 loop /snap/brackets/21
loop2       7:2      0 144,2M 1 loop /snap/brackets/46
loop3       7:3      0 83,8M  1 loop /snap/core/3604
sda          8:0      0 931,5G  0 disk
--sdal       8:1      0 100M   0 part
--sda2       8:2      0 294,2G  0 part
--sda3       8:3      0 472M   0 part
--sda4       8:4      0 1K      0 part
--sda5       8:5      0 536,2G  0 part /media/jefferson/Dados
--sda6       8:6      0 100,6G  0 part /
sdb         8:16     1 7,4G   0 disk
--sdb1       8:17     1 41,8M  0 part /media/jefferson/boot
--sdb2       8:18     1 7,4G   0 part
sr0         11:0     1 1024M  0 rom

jefferson@mb-acer:~$
```

5.3. Realizando o boot com essa instalação temos a seguinte versão do Kernel **original**



Versão 4.9.59
Data
Sun Oct 29 2017
11:47:10

6. Instalação do Kernel compilado

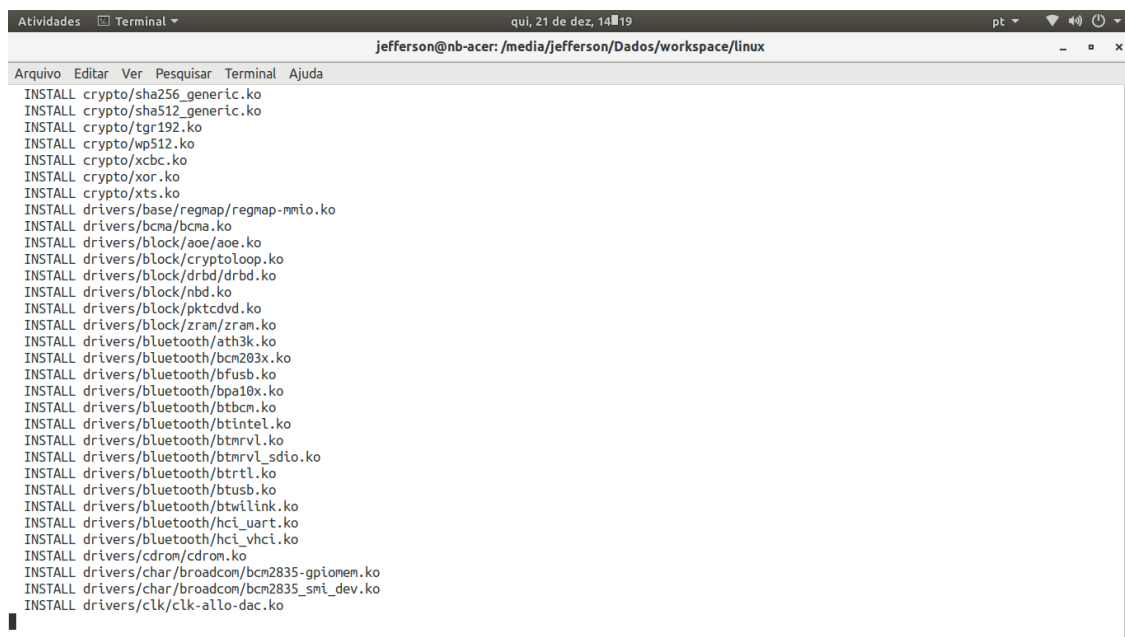
6.1. Para facilitar o trabalho montamos as partições nos seguintes diretórios

```
mkdir mnt
mkdir mnt/fat32
mkdir mnt/ext4
sudo mount /dev/sdb1 mnt/fat32
sudo mount /dev/sdb2 mnt/ext4
```

6.2. Realizamos a instalação dos módulos com o seguinte comando

```
sudo make ARCH=arm NSTALL_MOD_PATH=mnt/ext4 modules_install
```

Instalando os módulos



```
Arquivo Editar Ver Pesquisar Terminal Ajuda
qu, 21 de dez, 14:19
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux
INSTALL crypto/sha256_generic.ko
INSTALL crypto/sha512_generic.ko
INSTALL crypto/tgr192.ko
INSTALL crypto/wp512.ko
INSTALL crypto/xcrc32.ko
INSTALL crypto/xor.ko
INSTALL crypto/xts.ko
INSTALL drivers/base/regionmap/regmap-mmio.ko
INSTALL drivers/bcma/bcma.ko
INSTALL drivers/block/aoe/aoe.ko
INSTALL drivers/block/cryptoloop.ko
INSTALL drivers/block/drbd/drbd.ko
INSTALL drivers/block/nbd.ko
INSTALL drivers/block/pktcdvd.ko
INSTALL drivers/block/zram/zram.ko
INSTALL drivers/bluetooth/ath3k.ko
INSTALL drivers/bluetooth/bcm203x.ko
INSTALL drivers/bluetooth/bfusb.ko
INSTALL drivers/bluetooth/bpa10x.ko
INSTALL drivers/bluetooth/btbcm.ko
INSTALL drivers/bluetooth/btintel.ko
INSTALL drivers/bluetooth/btmrvl.ko
INSTALL drivers/bluetooth/btmrvl_sdio.ko
INSTALL drivers/bluetooth/btrtl.ko
INSTALL drivers/bluetooth/btusb.ko
INSTALL drivers/bluetooth/btwinlink.ko
INSTALL drivers/bluetooth/hci_uart.ko
INSTALL drivers/bluetooth/hci_vhci.ko
INSTALL drivers/cdrom/cdrom.ko
INSTALL drivers/char/broadcom/bcm2835-gpiomem.ko
INSTALL drivers/char/broadcom/bcm2835-smi-dev.ko
INSTALL drivers/clk/clk-allow-dac.ko
```

6.3. Para instala o Kernel basta realizar de alguns arquivos seguindo a sequência abaixo. O comando da primeira linha faz um backup do kernel original.

```
sudo cp mnt/fat32/$KERNEL.img mnt/fat32/$KERNEL-backup.img
sudo cp arch/arm/boot/zImage mnt/fat32/$KERNEL.img
sudo cp arch/arm/boot/dts/*.dtb mnt/fat32/
sudo cp arch/arm/boot/dts/overlays/*.dtb* mnt/fat32/overlays/
sudo cp arch/arm/boot/dts/overlays/README mnt/fat32/overlays/
```

```
Atividades Terminal
Kernel.docx - LibreOffice Writer
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux
Arquivo Editar Ver Pesquisar Terminal Ajuda
INSTALL mnt/ext4/lib/firmware/cpia2/stv0672_vp4.bin
INSTALL mnt/ext4/lib/firmware/yam/1200.bin
INSTALL mnt/ext4/lib/firmware/yam/9600.bin
DEPMOD 4.9.70+
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$ sudo cp mnt/fat32/$KERNEL.img mnt/fat32/$KERNEL-backup.img
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$ sudo cp arch/arm/boot/zImage mnt/fat32/$KERNEL.img
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$ sudo cp arch/arm/boot/dts/*.dtb mnt/fat32/
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$ sudo cp arch/arm/boot/dts/overlays/*.dtb* mnt/fat32/overlays/
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$ sudo cp arch/arm/boot/dts/overlays/README mnt/fat32/overlays/
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$ sudo umount mnt/fat32
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$ sudo umount mnt/ext4
jefferson@nb-acer: /media/jefferson/Dados/workspace/linux$
```

Resultado após backup e copia dos novos arquivos

Nome	Tamanho	Modificado
bcm2710-rpi-cm3.dtb	14,8 kB	14:20
bcm2835-rpi-a.dtb	9,0 kB	14:20
bcm2835-rpi-a-plus.dtb	9,1 kB	14:20
bcm2835-rpi-b.dtb	9,2 kB	14:20
bcm2835-rpi-b-plus.dtb	9,4 kB	14:20
bcm2835-rpi-b-rev2.dtb	9,3 kB	14:20
bcm2835-rpi-zero.dtb	9,0 kB	14:20
bcm2836-rpi-2-b.dtb	10,3 kB	14:20
bootcode.bin	50,2 kB	11 de ago
cmdline.txt	142 bytes	31 de dez de 1979
config.txt	1,6 kB	28 de nov
COPYING.linux	18,7 kB	21 de ago de 2015
fixup.dat	6,6 kB	30 de out
fixup_cd.dat	2,6 kB	30 de out
fixup_db.dat	9,7 kB	30 de out
fixup_x.dat	9,7 kB	30 de out
issue.txt	145 bytes	29 de nov
kernel.img	4,4 MB	14:20
kernel7.img	4,6 MB	30 de out
kernel-backup.img	4,4 MB	14:20
LICENSE.broadcom	1,5 kB	18 de nov de 2015
LICENSE.oracle	19,0 kB	29 de nov
overlays	119 itens	14:20
start.elf	2,8 MB	30 de out
start_cd.elf	667,5 kB	30 de out
start_db.elf	5,0 MB	30 de out
start_x.elf		

Através do comando `uname -a` podemos observar a versão do Kernel compilada e atualizada com a data da compilação 21/12/2017(hoje).

```
Starting Plymouth Boot Screen...
[ OK ] Started Permit User Sessions.
[ OK ] Started /etc/rc.local Compatibility.
Starting Hold until boot process finishes up...
Starting Terminate Plymouth Boot Screen...

Raspbian GNU/Linux 9 raspberrypi tty1
raspberrypi login: pi
Password:
Last login: Wed Nov 29 02:37:56 UTC 2017 on tty1
Linux raspberrypi 4.9.70+ #0 Thu Dec 21 12:16:39 -02 2017 armv6l

The programs included with the Debian GNU/Linux system are free software:
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ uname -a
Linux raspberrypi 4.9.70+ #0 Thu Dec 21 12:16:39 -02 2017 armv6l GNU/Linux
pi@raspberrypi:~$
```

Versão 4.9.70
Data
Thu Dec 21 2017
12:16:39