

## Tutorial de instalação de Linux para TV Box - modelo Tx2



Elaborado pela equipe do Smart Campus Unicamp  
<https://smartcampus.prefeitura.unicamp.br/>  
[smartcampus@unicamp.br](mailto:smartcampus@unicamp.br)

Campinas, 29 de junho de 2023

## Tutorial de instalação de Linux para TV Box - modelo Tx2



Inicialmente, deve-se substituir o sistema operacional Android que vem instalado por Linux. Para isso será necessário um cartão de memória (mínimo 8gb), onde iremos gravar o programa Multitool e esse programa será responsável pela gravação da imagem do Linux - distribuição Armbian, específica para a Tv Box (RK322X). As operações de ligar e desligar a tv box devem ser feitas retirando o conector de energia do dispositivo.

### **Passo 1**

Fazer o download do multitool em <https://users.armbian.com/jock/rk322x/multitool/>

O arquivo estará compactado com xz, para descompactar pode-se utilizar o comando `unxz multitool.img.xz` (caso necessário, instale o pacote xz-utils: `sudo apt-get install xz-utils`).

### **Passo 2**

Gravar a imagem do Multitool no cartão de memória. Para isso pode-se utilizar o programa gravador de imagem USB presente em distribuições Linux como Ubuntu, por exemplo. Neste programa você deverá indicar a imagem de origem `multitool.img` e destino o cartão de memória. Outra forma é através do comando: `dd if=multitool.img of=/dev/sdb status=progress` (trocar o caminho e o nome do device `sdb` conforme seu sistema). Aguardar até o término da gravação.

### **Passo 3**

Inserir o cartão de memória na tv box desligada, ligar e aguardar a tela inicial, neste momento será feita a expansão do sistema de arquivos para utilizar todo o cartão, após essa etapa desligar a tv box e retirar o cartão.

### **Passo 4**

Baixar a imagem do Linux, distribuição Armbian, que será utilizada através do endereço:

<https://imola.armbian.com/archive/rk322x-box/archive/>

Em nosso caso utilizamos a imagem:

`Armbian_23.5.1_Rk322x-box_bookworm_current_6.1.30.img.xz`

Ela não possui interface gráfica. Após o download, descompactar a imagem com `unxz` e gravar o arquivo `.img` na pasta Images do cartão de memória na partição Multitool. É nesta pasta que o Multitool irá procurar imagens para gravar na tv box.

### **Passo 5**

Inserir o cartão na tv box desligada, ligar e aguardar o início do Multitool, escolher a opção de gravar imagem, escolher o destino e a imagem que deseja gravar, após o término do processo, escolher a opção shutdown e remover o cartão.

### **Passo 6**

Religando a tv box, o sistema operacional novo já deverá funcionar. Aguarde um momento até solicitar a nova senha de root e criação de um usuário do sistema. Serão solicitadas algumas configurações de timezone, linguagem / teclado e terminal (bash).

Os comandos `sudo rk322x-config`, `armbian-config` e `nmtui` poderão ser úteis para configurar Wi-Fi, teclado layout, etc.

Verificamos que o mac address da interface Wi-Fi muda a cada reboot, caso seja necessário fixar um valor utilize umas das opções:

1 - adicionar a linha abaixo com o mac desejado no arquivo `/etc/network/interfaces`:

```
pre-up ifconfig wlan0 hw ether XX:XX:XX:XX:XX:XX
```

2 - através do Network Manager pelo comando:

```
nmcli con modify SSID 802-11-wireless.cloned-mac-address  
XX:XX:XX:XX:XX:XX
```

3 - adicionar no arquivo `/etc/NetworkManager/system-connections/SSID` no bloco `[wi-fi]` `cloned-mac-address=XX:XX:XX:XX:XX:XX`

### **Referências:**

<https://forum.armbian.com/topic/12656-csc-armbian-for-rk322x-tv-boxes/>

<https://www.youtube.com/watch?v=-jCgbhX5NNU>

## Instalação kiosk

Para uso na tvs, configurar o modo kiosk utilizando o script kiosk-installer.sh.

Esse script irá instalar o gerenciador de janelas openbox, o navegador chromium e configurar para ao iniciar o sistema já logar automaticamente no usuário criado (kiosk) e abrir o navegador na página (url) desejada.

### Passo 1

Crie o arquivo kiosk-installer.sh com o conteúdo abaixo:

```
#!/bin/bash

# be new
apt-get update

# get software
apt-get install \
  unclutter \
  xorg \
  chromium \
  openbox \
  lightdm \
  locales \
  -y

# dir
mkdir -p /home/kiosk/.config/openbox

# create group
groupadd kiosk

# create user if not exists
id -u kiosk &>/dev/null || useradd -m kiosk -g kiosk -s /bin/bash

# rights
chown -R kiosk:kiosk /home/kiosk

# remove virtual consoles
if [ -e "/etc/X11/xorg.conf" ]; then
  mv /etc/X11/xorg.conf /etc/X11/xorg.conf.backup
fi
cat > /etc/X11/xorg.conf << EOF
Section "ServerFlags"
  Option "DontVTSwitch" "true"
EndSection
EOF
```

```

# create config
if [ -e "/etc/lightdm/lightdm.conf" ]; then
    mv /etc/lightdm/lightdm.conf /etc/lightdm/lightdm.conf.backup
fi
cat > /etc/lightdm/lightdm.conf << EOF
[SeatDefaults]
autologin-user=kiosk
user-session=openbox
EOF
cat > /etc/X11/xorg.conf.d/01-armbian-defaults.conf << EOF
# Default Armbian config
Section "Monitor"
    Identifier "LVDS0"
    Option "DPMS" "false"
EndSection

Section "ServerLayout"
    Identifier "ServerLayout0"
    Option "BlankTime" "0"
    Option "StandbyTime" "0"
    Option "SuspendTime" "0"
    Option "OffTime" "0"
EndSection
EOF

# create autostart
if [ -e "/home/kiosk/.config/openbox/autostart" ]; then
    mv /home/kiosk/.config/openbox/autostart
/home/kiosk/.config/openbox/autostart.backup
fi
cat > /home/kiosk/.config/openbox/autostart << EOF
#!/bin/bash

setterm -blank 0 &
setterm -powersave off &
xset -dpms &

unclutter -idle 0.1 -grab -root &

while :
do
    xrandr --auto
    chromium \
        --no-first-run \
        --start-maximized \
        --disable \
        --disable-translate \
        --disable-infobars \

```

```
--disable-suggestions-service \  
--disable-save-password-bubble \  
--disable-session-crashed-bubble \  
--incognito \  
--kiosk "https://www.prefeitura.unicamp.br/apps/noticia.php"  
sleep 5  
done &  
EOF  
  
echo "Done!"
```

## **Passo 2**

Adicione permissão de execução ao script e execute:

```
chmod +x kiosk-installer.sh
```

```
./kiosk-installer.sh
```

Desabilitar hibernation e suspend:

```
sudo systemctl unmask sleep.target suspend.target hibernate.target  
hybrid-sleep.target
```

Reinicie o sistema.

Caso necessário, o arquivo de configuração onde consta a url do site a exibir está em: /home/kiosk/.config/openbox/autostart

## **Referência**

<https://github.com/josfaber/debian-kiosk-installer>

## Instalação Bridge - usando proxy arp

**AP --> TV BOX [WI-FI] --> rede cabeada [eth0] --> cliente rede cabeada**

Configure a interface wlan0 através do comando nmtui, certifique-se que a conexão está ativa e operacional.

### Passo 1

Trocar o nome da interface de rede end0 para eth0. Para isso criar o arquivo /etc/systemd/network/10-eth0.link com o mac da interface e o nome eth0

```
[Match]
MACAddress=xx:xx:xx:xx:xx:xx
[Link]
Name=eth0
```

### Passo 2

Criar e executar o script bridge-tvbox.sh com o conteúdo abaixo:

```
#!/usr/bin/env bash

set -e

[ $EUID -ne 0 ] && echo "run as root" >&2 && exit 1

# parprouted - Proxy ARP IP bridging daemon
# dhcp-helper - DHCP/BOOTP relay agent

apt update && apt install -y avahi-daemon parprouted dhcpcd5 dhcp-helper

systemctl stop dhcp-helper
systemctl enable dhcp-helper

# Enable ipv4 forwarding.
sed -i" s/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/ /etc/sysctl.conf

# Service configuration for standard WiFi connection. Connectivity will
# be lost if the username and password are incorrect.
systemctl restart wpa_supplicant.service

# Enable IP forwarding for wlan0 if it's not already enabled.
grep '^option ip-forwarding 1$' /etc/dhcpcd.conf || printf "option ip-forwarding 1\n" >>
/etc/dhcpcd.conf

# Disable dhcpcd control of eth0.
grep '^denyinterfaces eth0$' /etc/dhcpcd.conf || printf "denyinterfaces eth0\n" >>
/etc/dhcpcd.conf

# Configure dhcp-helper.
```

```
cat > /etc/default/dhcp-helper <<EOF
DHCPHELPER_OPTS="-b wlan0"
EOF
```

```
# Enable avahi reflector if it's not already enabled.
sed -i" 's/#enable-reflector=no/enable-reflector=yes/' /etc/avahi/avahi-daemon.conf
grep '^enable-reflector=yes$' /etc/avahi/avahi-daemon.conf || {
    printf "something went wrong...\n\n"
    printf "Manually set 'enable-reflector=yes in /etc/avahi/avahi-daemon.conf\n"
}
```

```
# I have to admit, I do not understand ARP and IP forwarding enough to explain
# exactly what is happening here. I am building off the work of others. In short
# this is a service to forward traffic from WiFi to Ethernet.
```

```
cat <<'EOF' >/usr/lib/systemd/system/parprouted.service
```

```
[Unit]
```

```
Description=proxy arp routing service
```

```
Documentation=https://raspberrypi.stackexchange.com/q/88954/79866
```

```
Requires=sys-subsystem-net-devices-wlan0.device dhcpcd.service
```

```
After=sys-subsystem-net-devices-wlan0.device dhcpcd.service
```

```
[Service]
```

```
Type=forking
```

```
# Restart until wlan0 gained carrier
```

```
Restart=on-failure
```

```
RestartSec=5
```

```
TimeoutStartSec=30
```

```
# clone the dhcp-allocated IP to eth0 so dhcp-helper will relay for the correct subnet
```

```
ExecStartPre=/bin/bash -c '/sbin/ip addr add $(/sbin/ip -4 -br addr show wlan0 |
/bin/grep -Po "\d+\.\d+\.\d+\.\d+")/32 dev eth0'
```

```
ExecStartPre=/sbin/ip link set dev eth0 up
```

```
ExecStartPre=/sbin/ip link set wlan0 promisc on
```

```
ExecStart=-/usr/sbin/parprouted eth0 wlan0
```

```
ExecStopPost=/sbin/ip link set wlan0 promisc off
```

```
ExecStopPost=/sbin/ip link set dev eth0 down
```

```
ExecStopPost=/bin/bash -c '/sbin/ip addr del $(/sbin/ip -4 -br addr show wlan0 |
/bin/grep -Po "\d+\.\d+\.\d+\.\d+")/32 dev eth0'
```

```
[Install]
```

```
WantedBy=wpa_supplicant.service
```

```
EOF
```

```
systemctl daemon-reload
```

```
systemctl enable parprouted
```

```
systemctl start parprouted dhcp-helper
```

### Passo 3

Conecte um equipamento na porta RJ45 do tv box e configure ip dinâmico.



**Referências:**

<https://gist.github.com/yoss85/a4c022f14ca1d10ae26369415ceb1c05>

<https://unix.stackexchange.com/questions/733080/debian-11-rename-network-interfa>

[ces](#)

## Instalação Bridge - usando nat

**AP --> TV BOX [WI-FI] --> rede cabeada [eth0] --> cliente rede cabeada**

Configure a interface wlan0 através do comando nmtui, certifique-se que a conexão está ativa e operacional.

### Passo1

Instale os pacotes a seguir, e responda Não para salvar as regras atuais do iptables.  
sudo apt-get update && sudo apt-get install dnsmasq iptables iptables-persistent -y

Edite o arquivo /etc/network/interfaces, e comente a configuração de rede end0 (se existir), insira a configuração de rede:

```
allow-hotplug end0
iface end0 inet static
    address 172.24.0.1
    netmask 255.255.255.0
    network 172.24.0.0
    broadcast 172.24.0.255
    dns-nameservers 1.1.1.1 1.0.0.1
#####(Static)#####
up ip addr add 172.24.0.1/24 dev end0
```

### Passo2

Execute os comandos abaixo:

```
ip addr add 172.24.0.1/24 dev end0
iptables -A FORWARD -o wlan0 -i end0 -s 172.24.0.0/24 -m conntrack --ctstate NEW
-j ACCEPT
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -t nat -F POSTROUTING
iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
sh -c "iptables-save > /etc/iptables.ipv4.nat"
sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
/etc/init.d/dnsmasq stop
cp /etc/dnsmasq.conf /etc/dnsmasq.conf-backup
```

### Passo 3

Edite o arquivo /etc/dnsmasq.conf deixando-o apenas com o conteúdo abaixo:

```
interface=end0
listen-address=172.24.0.1
bind-interfaces
server=1.1.1.1
domain-needed
bogus-priv
dhcp-range=172.24.0.100,172.24.0.250
```

Edite o arquivo /etc/sysctl.conf e adicione as linhas abaixo no final do arquivo:  
net.ipv4.conf.default.forwarding=1  
net.ipv4.conf.all.forwarding=1

Edite o arquivo /etc/rc.local e insira as linhas abaixo antes de "exit 0"  
iptables-restore < /etc/iptables.ipv4.nat

#### **Passo 4**

Execute os comandos:  
systemctl enable dnsmasq  
systemctl enable iptables

#### **Passo 5**

Conecte um equipamento na porta RJ45 do tv box e configure o ip estático  
172.24.0.101 / 24, GW 172.24.0.1

#### **Referências:**

<https://forum.armbian.com/topic/20516-how-to-internet-source-from-wifi-to-ethernet-b-ridge-wifi-to-ethernet/>