



node > Broadcasts > Kernel radio <-> linux.fm

Kernel radio <-> linux.fm

▶ Kernel-radio : Stable Linux kernel source code lecture
Generative infos

[linux.stable/include/uapi/linux/ip6_tunnel.h](#)

Kernel radio offers you a reading of the source code of the stable Linux kernel, file after file. This stream is a tribute and a reincarnation of the late Linux.fm project which has long proposed the same reading.

The Linux kernel is a major technopolitical issue, most major digital companies contribute to it, its development concentrates many interests and influences, it is one of the first guardians of the security and integrity of our connected devices. The reading is carried out by a synthetic voice, the technical description of the device is detailed below



Technical description

The workflow : git pull of the linux stable kernel > bash script that launch the streaming fed by a loop / virtual soundcard > bash script that find recursively all the files of the kernel source and send each one to a text to speach software that says the text in the linked loop / virtual soundcard.

<https://git.kernel.org/pub/scm/linux/kernel/git/stable/>

Prepare the server

Install a dummy sound card and a loop virtual soundcard with also

```
sudo apt install alsa-utils sox oggwfed espeak mbrla-us1 mbrla
sudo modprobe snd-dummy
sudo modprobe snd-aloop
```

It gives several input / output, to see them, the user muts be in the audio group

```
sudo usermod -a -G audio YourUser
```

Then log-out and in and list devices for playing or recording

```
aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: Dummy [Dummy], device 0: Dummy PCM [Dummy PCM]
  Subdevices: 7/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 1: Loopback [Loopback], device 0: Loopback PCM [Loopback PCM]
  Subdevices: 7/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 1: Loopback [Loopback], device 1: Loopback PCM [Loopback PCM]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
```

For recording :

```
arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: Dummy [Dummy], device 0: Dummy PCM [Dummy PCM]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 1: Loopback [Loopback], device 0: Loopback PCM [Loopback PCM]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 1: Loopback [Loopback], device 1: Loopback PCM [Loopback PCM]
  Subdevices: 7/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
```

To activate on start-up, add the line to the file :

```
sudo nano /etc/modules-load.d/modules.conf

# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored
snd-dummy
snd-aloop
```

```
systemctl restart systemd-modules-load.service
```

A good tech documentation :

- https://alsa.opensrc.org/Jack_and_Loopback_device_as_Alsa-to-jack_bridge#Compiling_snd-aloop_if_needed
- <https://www.alsa-project.org/main/index.php/Matrix:Module-dummy>
- <https://www.alsa-project.org/main/index.php/Matrix:Module-aloop>

Scripts

- We listen to the loop sound card to have a continus stream to send to icecast : arecord -c 2 -f cd -r 48000 -D hw:1,1,0
 - We play on the linked loop soundcard to the input : aplay -c 2 -D hw:1,0,0
- You can't use the same input / output virtual device for another script but you have 7 more**

update the kernel source & launch the find process and the stream

launchkernelfm.sh

```
#!/bin/bash

#####
# update the kernel source & launch the find process and the stream #
#####
cd /data/pnode/algo/linux-stable
git pull

cd /data/pnode/algo/kernel-radio
./launchkernelfind.sh &

sleep 1

while true
do
    arecord -c 2 -f cd -r 48000 -D hw:1,1,0 | sox -t raw --ignore-length -r 44100 -e signed
done
```

find all files of the os and send 2 espeak one by one

launchkernelfind.sh

```
#!/bin/bash

#####
# find all files of the os and send 2 espeak one by one #
#####
sleep 3

while true
do
    find /data/pnode/algo/linux-stable -exec ./espeakkernel.sh {} \;
done
```

speach synths of the filename and its content + update icecast metadata

espeakkernel.sh

```
#!/bin/bash

#####
# speach synths of the filename and its content #
#####
# usage: ./espeakkernel.sh PATH_TO_FILE #
#####

echo $(basename "$1") | espeak -a 90 -p 59 -s 140 -g 7 -v mb-us1 --stdin --stdout| sox -t

path=$(echo $1|cut -d '/' -f5-)

curl -k -X "GET" "http://212.83.128.6/admin/metadata?mount=/kernel-lecture.ogg&mode=updin

sleep 0.5
espeak -a 100 -p 59 -s 140 -g 7 -v mb-us1 -f "$1" --stdout| sox -t raw --ignore-length -r
exit 0
```

kill all the process

killkernelfm.sh

```
#!/bin/bash

#####
# kill all the process #
#####
pkill -f launchkernelfm.sh
pkill -f launchkernelfind.sh
pkill -f espeakkernel.sh
```

Start the system as a service

Create a systemd file to create a service :

```
sudo nano /lib/systemd/system/kernelradio.service
```

```
[Unit]
Description=Stable Linux kernel source code lecture

[Service]
Type=simple
#Type=forking
ExecStart=/data/pnode/algo/kernel-radio/launchkernelfm.sh
StandardOutput=null
Restart=always
RestartSec=1
ExecStop=/data/pnode/algo/kernel-radio/killkernelfm.sh

# Configures the time to wait before service is stopped forcefully.
TimeoutStopSec=30

[Install]
WantedBy=multi-user.target
Alias=kernelradio.service
```

Start the service and enable at start-up

```
sudo systemctl start kernelradio.service
sudo systemctl status kernelradio.service
sudo systemctl enable kernelradio.service
```