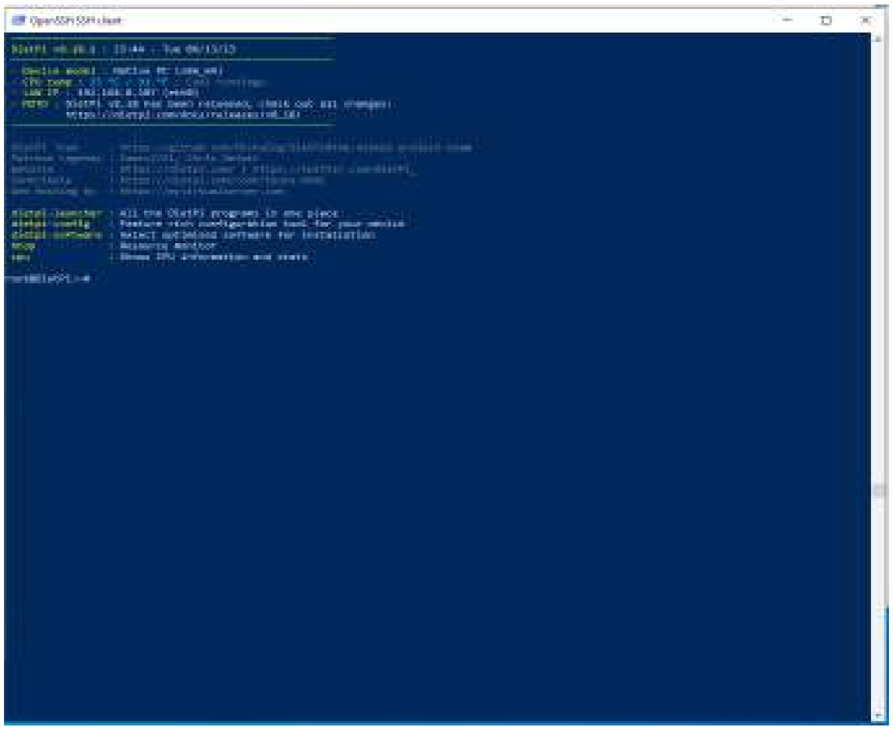
After you install DIETPI this is what you will see
Type dietpi-config ENTER
Select AUDIO OPTIONS
Select SOUND CARD ENTER
Scroll to your sound card ENTER
ESC to go back to AUDIO OPTIONS
ESC to return to the first screen



```
[illegible]
[illegible]

dietpi-launcher  : All the DietPi programs in one place
dietpi-config    : Feature-rich configuration tool for your device
dietpi-software  : Select optimised software for installation
htop             : Resource monitor
cpu              : Shows CPU information and stats

root@DietPi:~#
```

We need to tell MPD where your music is and which DAC to use. This is done in the MPD Config-File which we edit with the command: nano /etc/mpd.conf
Please study this file carefully and make the following changes (the # is important !):

Blitz's /etc/mpd.conf file looks like this (adjust to to your dac and music directory):

```
music_directory        "/mnt/Musik"
```
use the name of you NAS or, it like me and unlike Blitz, the name you have given your attached music drive. I name mine music – *this drive is mounted and named in dietpi-drive_manager*

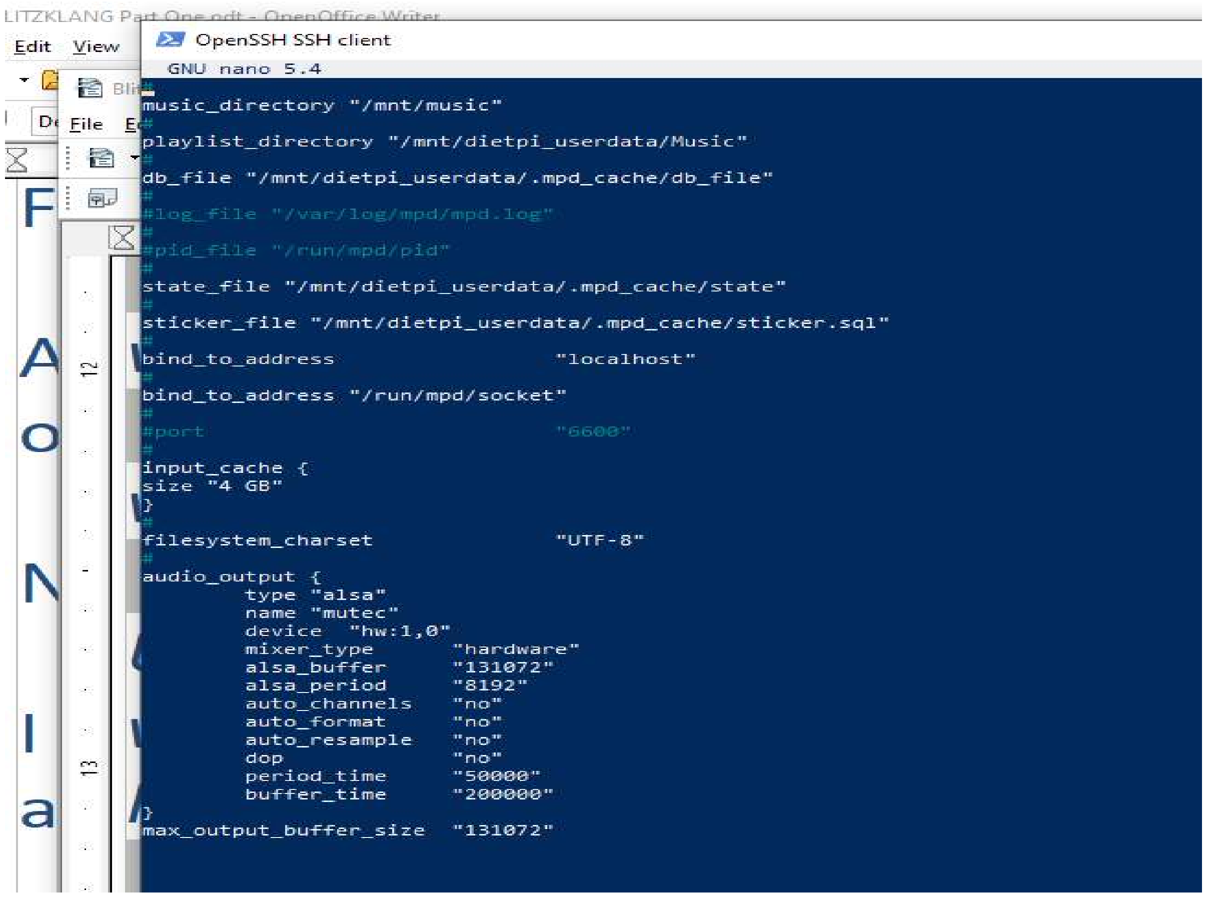
```
playlist_directory            "/var/lib/mpd/playlists"
db_file               "/var/lib/mpd/tag_cache"
log_file                     "/var/log/mpd/mpd.log"
pid_file                     "/run/mpd/pid"
```
I left these as default in my setup – when I would use these settings I could no longer reach MPD – I figure this is because of the attached drive
```
state_file                    "/var/lib/mpd/state"
sticker_file                  "/var/lib/mpd/sticker.sql"
input_cache {
size "4 GB"
}
```

```
filesystem_charset             "UTF-8"
audio_buffer_size "8192"
buffer_before_play "100%"

audio_output {
        type "alsa"
        name "Andrea"
```
whatever name you choose
```
        device "hw:1,0"
```
what you chose in dietpi-config/Audio Options/DAC
```
        mixer_type "hardware"
```
Use this setting for fixed volume/no control – use “software” if you want the system to be able to adjust volume. Hardware is better.
```
        alsa_buffer "131072"
        alsa_period "8192"
        auto_channels         "no"
        auto_format           "no"
        auto_resample         "no"
        dop "no"
        period_time "50000"
        buffer_time "200000"
}

max_output_buffer_size "131072"
```

You can minimize the size of this folder along with making it easier to deal with by removing all of the text. Not that it makes a difference for sound quality. Mine is slightly larger than Blitz's – kept those areas he is using which I figure are for NAS in case I made change in the future- the only parts that are active are in white type.



```
#
music_directory "/mnt/music"
#
playlist_directory "/mnt/dietpi_userdata/Music"
#
db_file "/mnt/dietpi_userdata/.mpd_cache/db_file"
#
#log_file "/var/log/mpd/mpd.log"
#
#pid_file "/run/mpd/pid"
#
state_file "/mnt/dietpi_userdata/.mpd_cache/state"
#
sticker_file "/mnt/dietpi_userdata/.mpd_cache/sticker.sql"
#
bind_to_address                 "localhost"
#
bind_to_address "/run/mpd/socket"
#
#port                           "6600"
#
input_cache {
size "4 GB"
}
#
filesystem_charset              "UTF-8"
#
audio_output {
        type "alsa"
        name "mutec"
        device  "hw:1,0"
        mixer_type          "hardware"
        alsa_buffer         "131072"
        alsa_period         "8192"
        auto_channels       "no"
        auto_format         "no"
        auto_resample       "no"
        dop                 "no"
        period_time         "50000"
        buffer_time         "200000"
}
max_output_buffer_size  "131072"
```

You always need to reboot after making changes any changes to conf files or stop and restart a service, which is a bit more taping but is faster. (Did you mean taping, I am not familiar with the term)

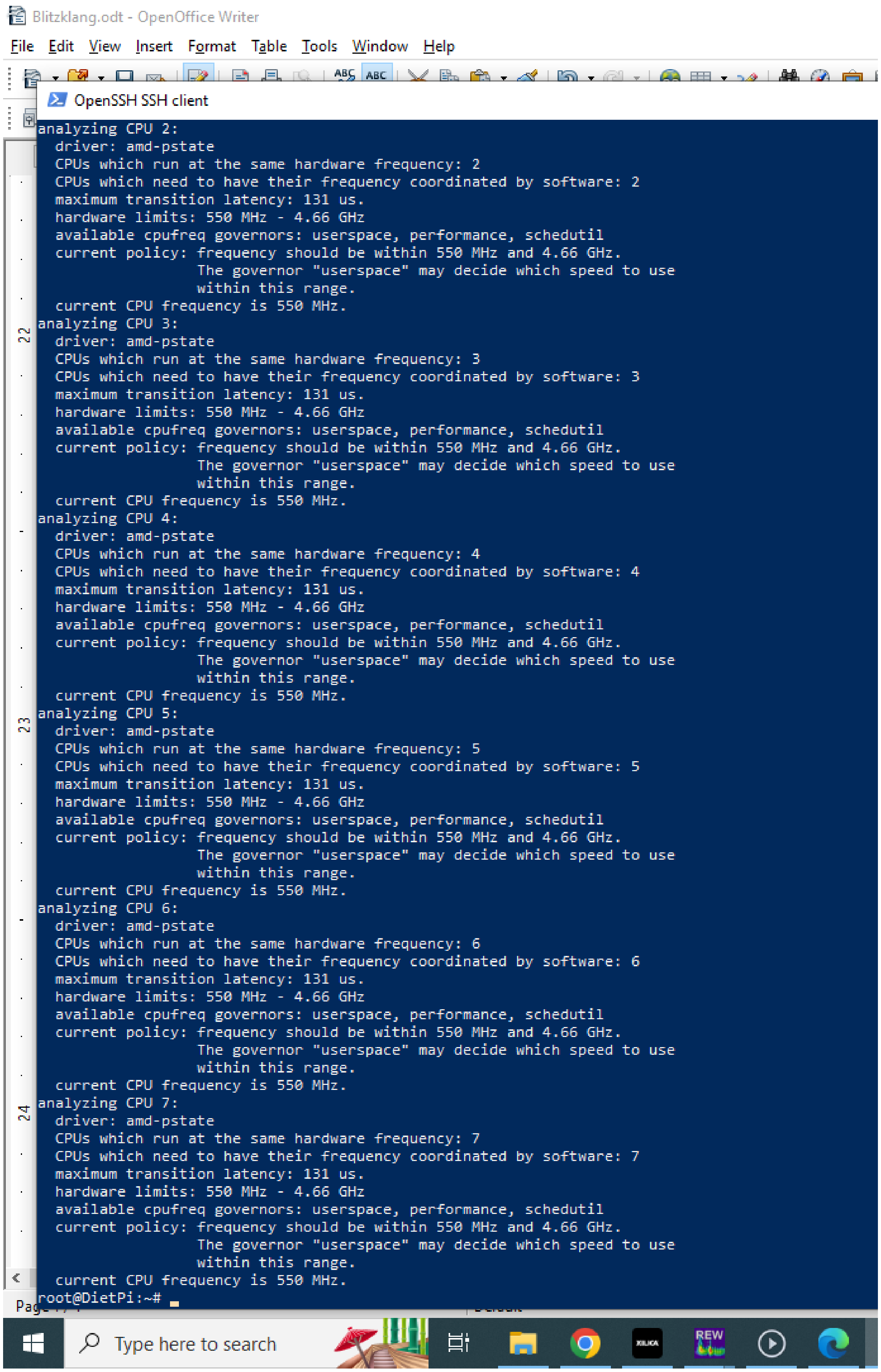
Please install a tool to understand precise what is going on with your cpu frequency and your governor and cpu driver:

```
apt-get install -y cpufrequtils
```

After install please type

```
cpufreq-info
```

You will see a similar picture like this, but with different content. Please make a hardcopy and send it to me:
I think we should say post it


Blitzklang.odt - OpenOffice Writer
File Edit View Insert Format Table Tools Window Help
OpenSSH SSH client
Type here to search

```
analyzing CPU 2:
  driver: amd-pstate
  CPUs which run at the same hardware frequency: 2
  CPUs which need to have their frequency coordinated by software: 2
  maximum transition latency: 131 us.
  hardware limits: 550 MHz - 4.66 GHz
  available cpufreq governors: userspace, performance, schedutil
  current policy: frequency should be within 550 MHz and 4.66 GHz.
                  The governor "userspace" may decide which speed to use
                  within this range.
  current CPU frequency is 550 MHz.
analyzing CPU 3:
  driver: amd-pstate
  CPUs which run at the same hardware frequency: 3
  CPUs which need to have their frequency coordinated by software: 3
  maximum transition latency: 131 us.
  hardware limits: 550 MHz - 4.66 GHz
  available cpufreq governors: userspace, performance, schedutil
  current policy: frequency should be within 550 MHz and 4.66 GHz.
                  The governor "userspace" may decide which speed to use
                  within this range.
  current CPU frequency is 550 MHz.
analyzing CPU 4:
  driver: amd-pstate
  CPUs which run at the same hardware frequency: 4
  CPUs which need to have their frequency coordinated by software: 4
  maximum transition latency: 131 us.
  hardware limits: 550 MHz - 4.66 GHz
  available cpufreq governors: userspace, performance, schedutil
  current policy: frequency should be within 550 MHz and 4.66 GHz.
                  The governor "userspace" may decide which speed to use
                  within this range.
  current CPU frequency is 550 MHz.
analyzing CPU 5:
  driver: amd-pstate
  CPUs which run at the same hardware frequency: 5
  CPUs which need to have their frequency coordinated by software: 5
  maximum transition latency: 131 us.
  hardware limits: 550 MHz - 4.66 GHz
  available cpufreq governors: userspace, performance, schedutil
  current policy: frequency should be within 550 MHz and 4.66 GHz.
                  The governor "userspace" may decide which speed to use
                  within this range.
  current CPU frequency is 550 MHz.
analyzing CPU 6:
  driver: amd-pstate
  CPUs which run at the same hardware frequency: 6
  CPUs which need to have their frequency coordinated by software: 6
  maximum transition latency: 131 us.
  hardware limits: 550 MHz - 4.66 GHz
  available cpufreq governors: userspace, performance, schedutil
  current policy: frequency should be within 550 MHz and 4.66 GHz.
                  The governor "userspace" may decide which speed to use
                  within this range.
  current CPU frequency is 550 MHz.
analyzing CPU 7:
  driver: amd-pstate
  CPUs which run at the same hardware frequency: 7
  CPUs which need to have their frequency coordinated by software: 7
  maximum transition latency: 131 us.
  hardware limits: 550 MHz - 4.66 GHz
  available cpufreq governors: userspace, performance, schedutil
  current policy: frequency should be within 550 MHz and 4.66 GHz.
                  The governor "userspace" may decide which speed to use
                  within this range.
  current CPU frequency is 550 MHz.
root@DietPi:~# _
```

Missing CPU0 & 1 – they say the same things

Before we start to make the tickless kernel let's check our current version

At this point Blitz assumes we are getting the idea of how to use LINUX

uname –r
I found I had to do uname first and then ask for uname -r

This will show you your current Kernel version and you should see something like 5.10
That is what both Blitz and I saw

We will be downloading the kernel from kernel.org (and yes, we want exactly this version of the kernel)

wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.17.tar.gz

wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.17.tar.sign

tar xvzf linux-5.19.17.tar.gz
This extracts the files we just downloaded

cd linux-5.19.17
We work on the kernel in this directory so the original kernel is not affected. If you mess up you will still have music if you are using it already and you should since it already sounds good

So, you downloaded, unpacked it and have gone into its folder

Now we need to install some tools to build it:

apt-get install build-essential linux-source bc kmod cpio flex libncurses5-dev libelf-dev libssl-dev dwarves bison

you will asked to say yes to proceed with the install

Now we copy your old configuration from dietpi into the new kernel version:

make olddefconfig

Now we make the fun part, so you can select what you want to do and change the dietpi config file:

make menuconfig

Than comes the configuration.

BY the way…please look over each setting carefully, not just those I explicitly named. Yours should look like mine (Blitz's).

DO NOT CHANGE ANY OTHER stuff !!! Or your Linux I was Kernel might be smoked. I smoked a dozen of kernels.
I was on my way to having that many

You find in the left upper corner the path of the menu where you do the settings, if you cant find stuff.
You will have to enlarge the screen and move over to the left side to see this – then you will have to move back to where the work is done.

It is a good idea to sure you are in the folder where our newly downloaded kernel is, which we have done previously by

cd linux-5.19.17

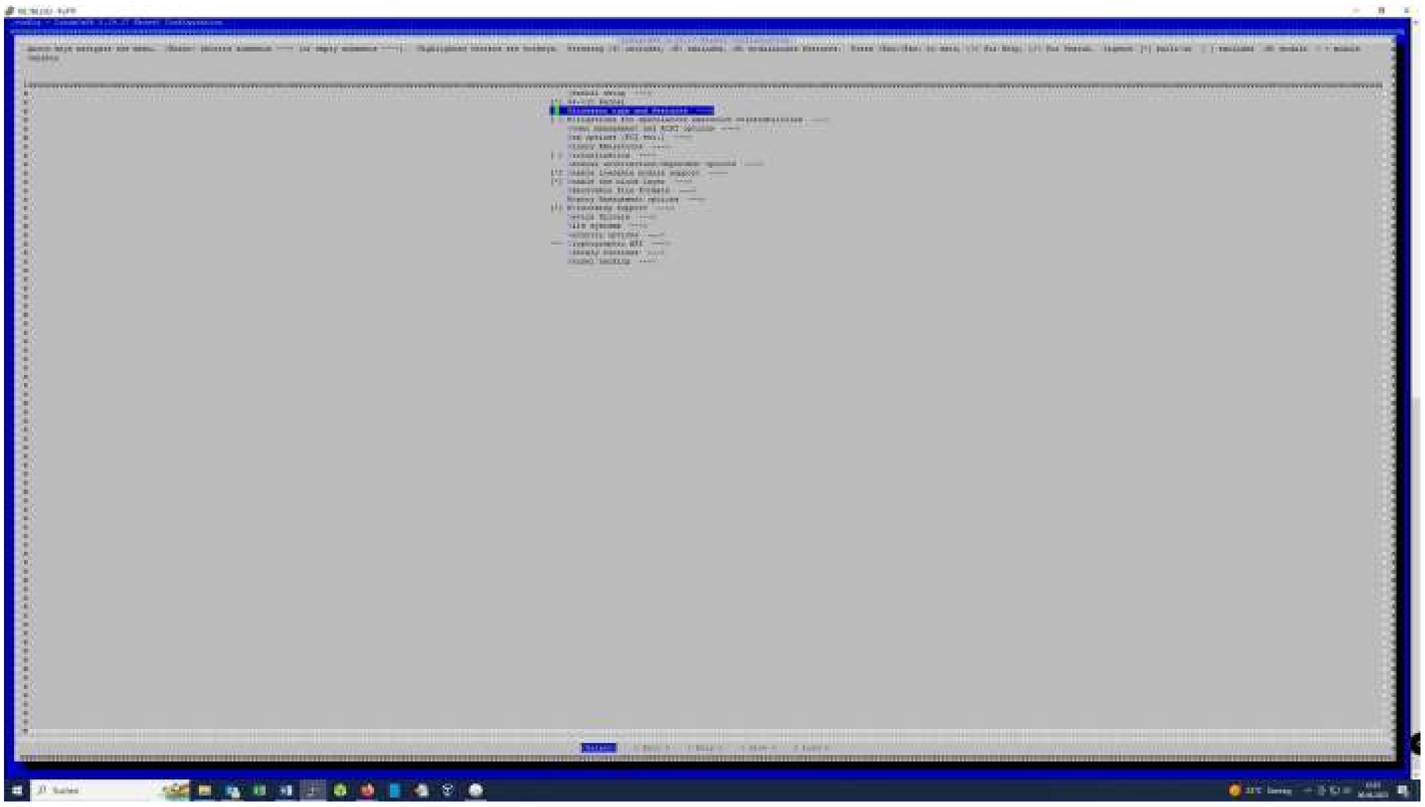
Type

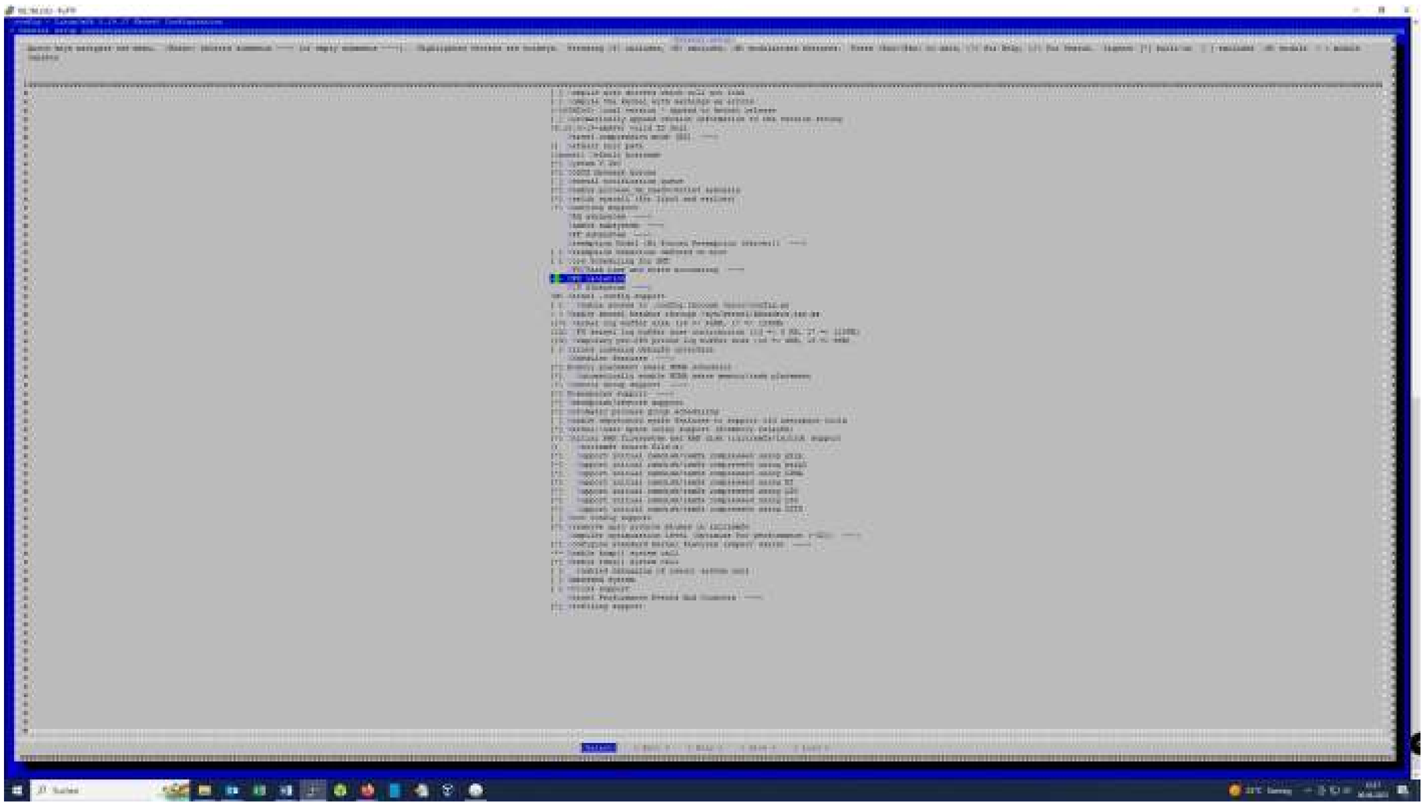
make menuconfig

(please ensure that the terminal windows is large enough so the whole menu can be displayed)

This is now what we want: x-check that yours looks like mine, sometimes you will need to go into the menus. There will be a screenshot to show you when this happens.
Press Y to get the X in the box – Press N to delete it – Press M to specify MODULE

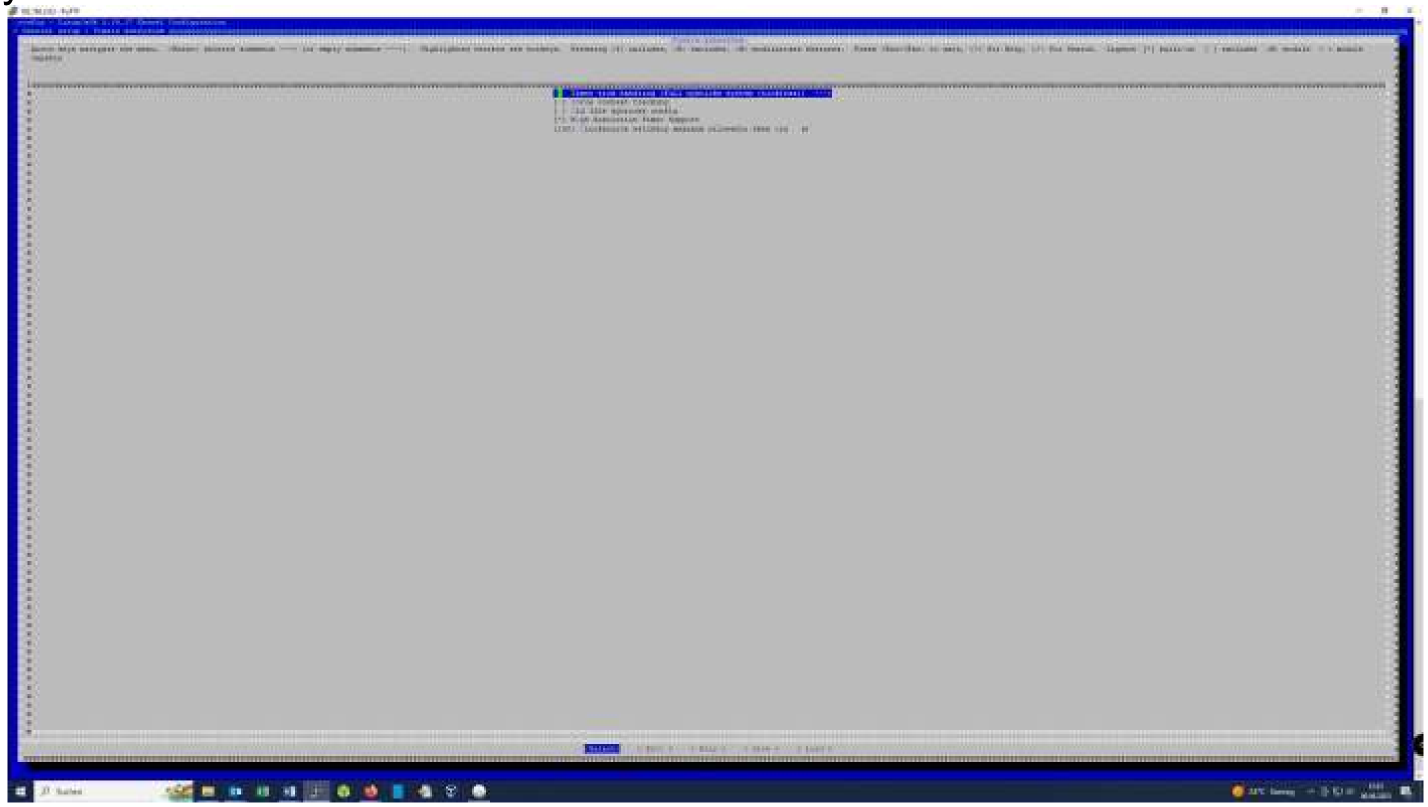
We disable anything for spec. Vulnerabilities and Virtualization:
Third & fourth lines



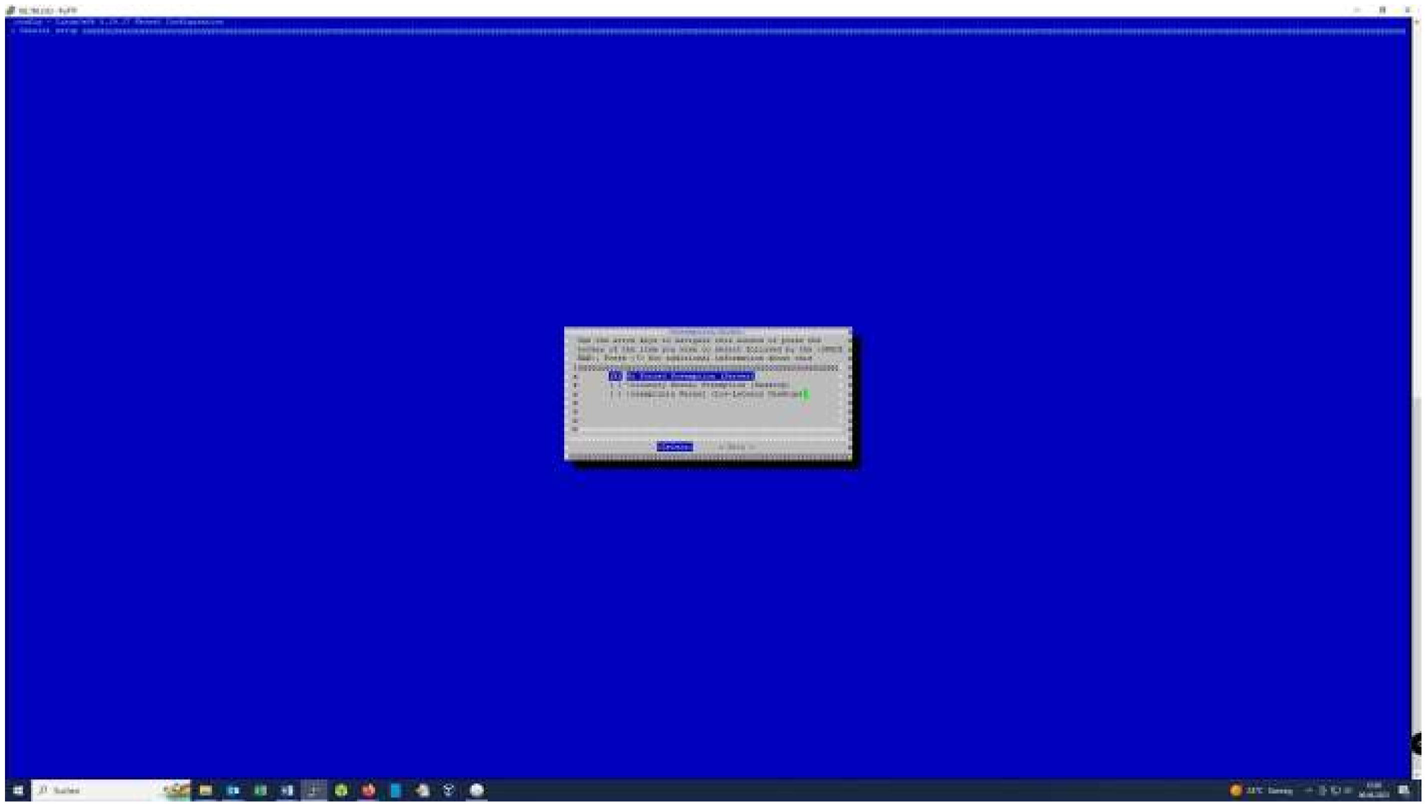


We want to setup tickless here and high resolution timer:

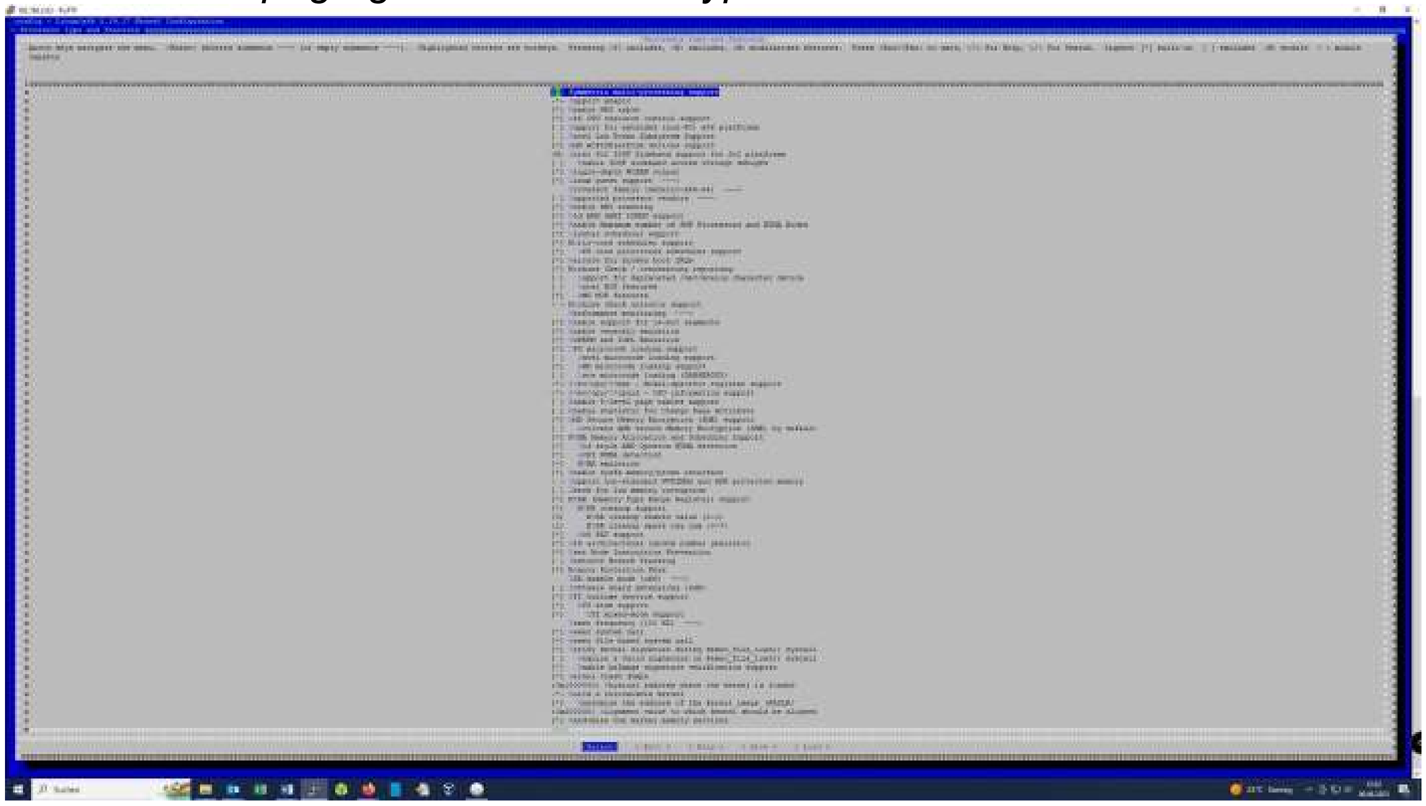
Where you see the green dot TIMER TICK HANDLING – ENTER – and choose FULL DYNTICKS SYSTEM (tickless) – ENTER – when you return your screen will look like this



We use the server mode here to minimize the thread overhead handling, We want no overhead and max throughput instead of low latency.
Path General/Preemption Model – ENTER and choose NO FORCED PREEMPTION – ENTER and you will see your choices

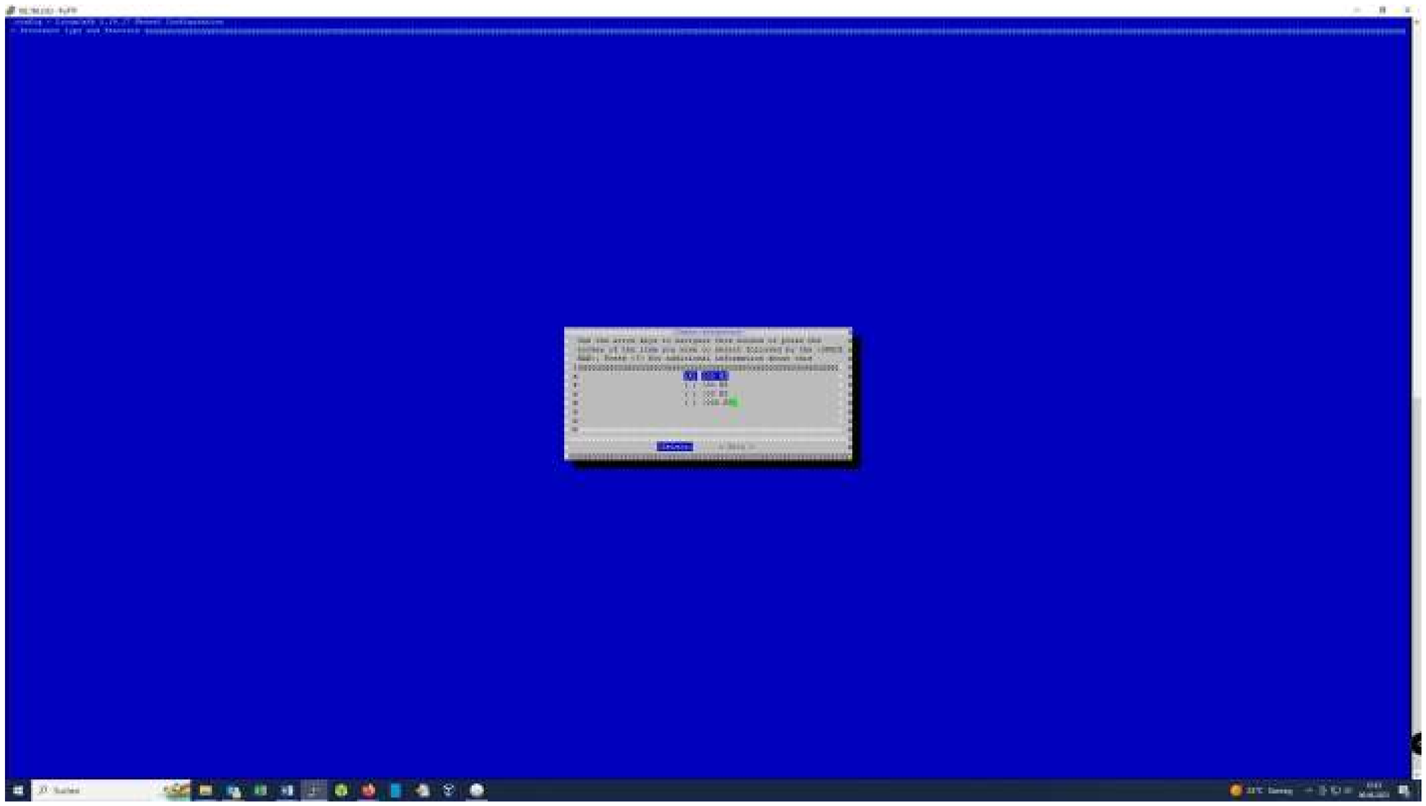


We get rid of the intel stuff and enable all the AMD stuff per below
From the first page go to Processor Type and Features



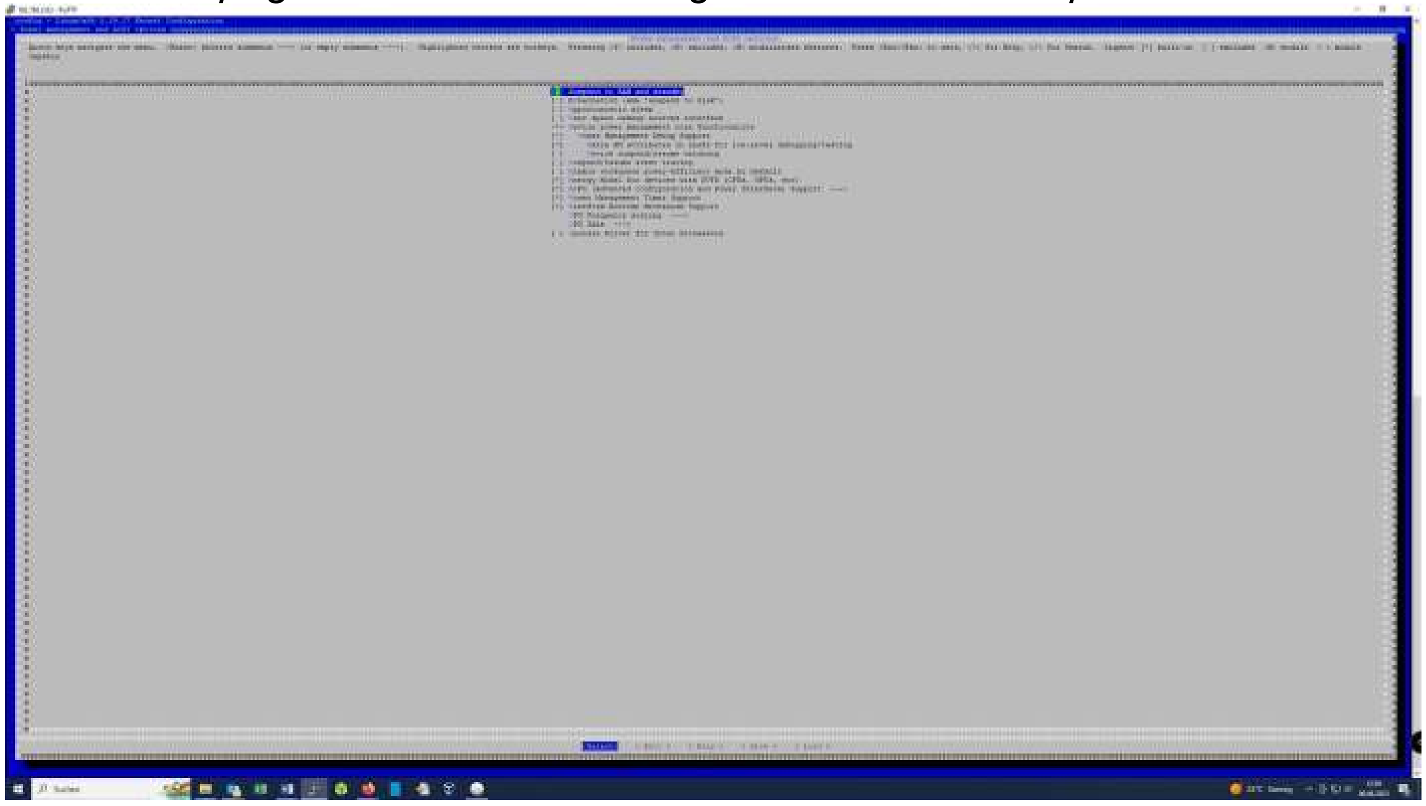
We set the timer frequency for highest throughput to 100HZ, less interrupts, less noise.

This is on the Processor Type and Features page -TIMER FREQUENCY ENTER and then select 100 Hz - ENTER

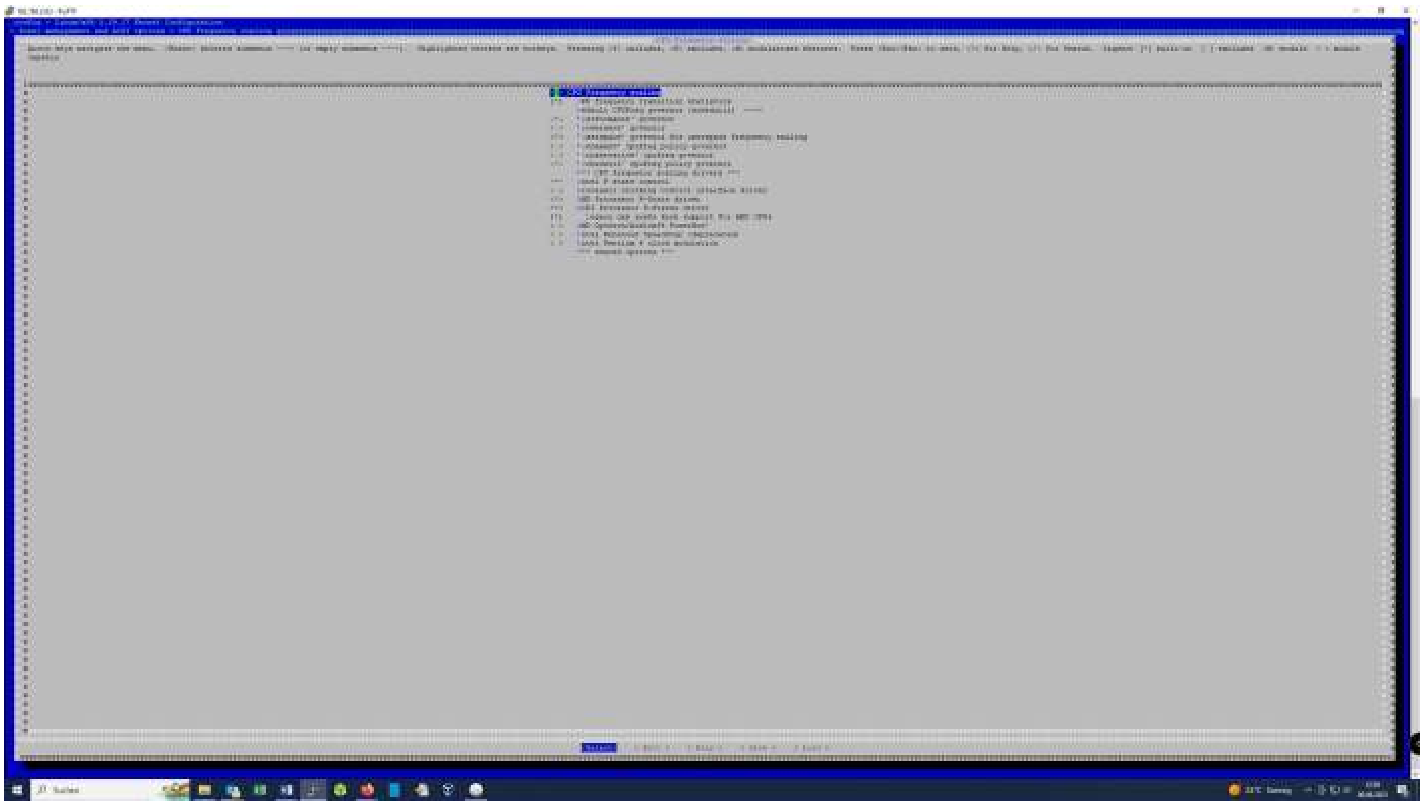


No Hibernation and stuff:

On the first page select Power Management and ACPI Options

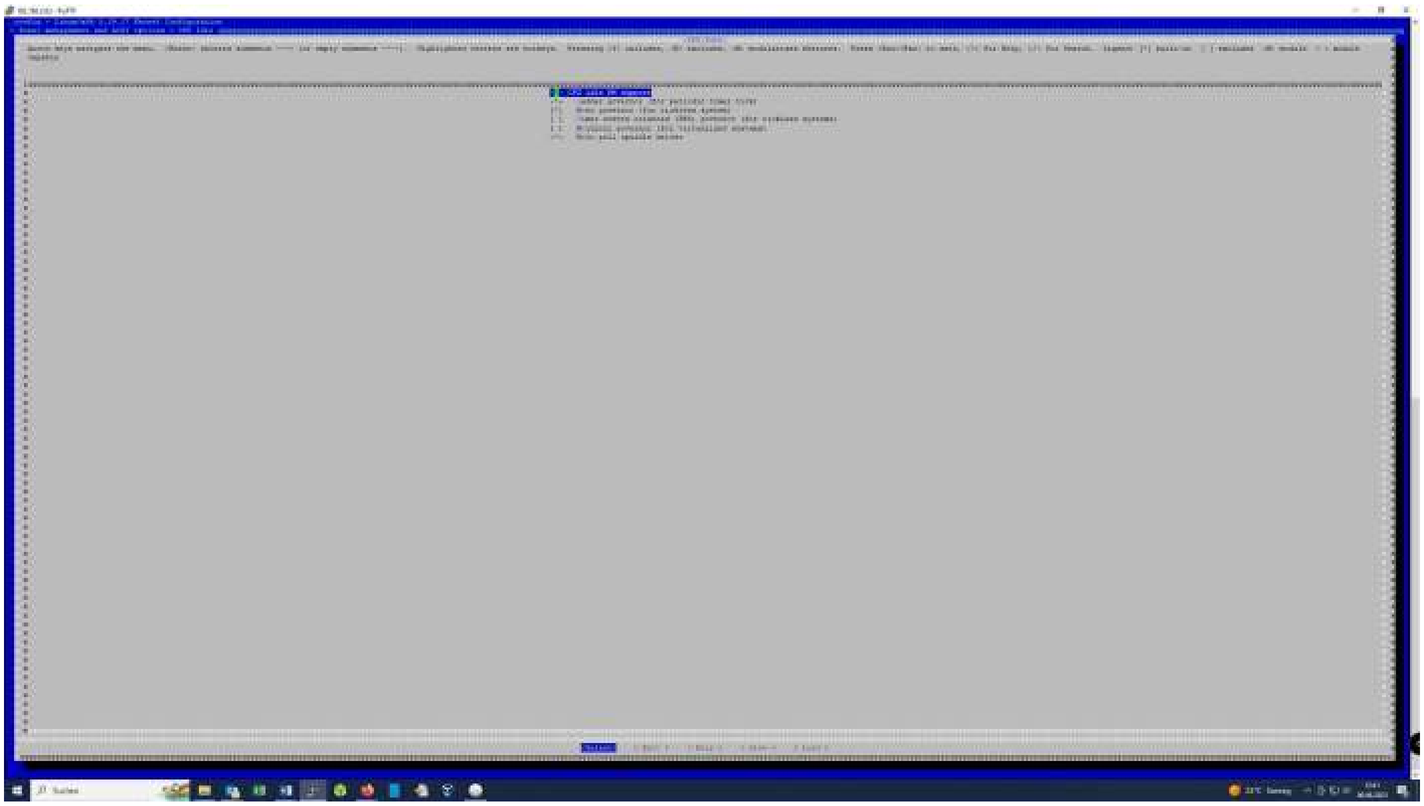


Governor Userspace enabled. AMD P-state & Driver enabled:
Within Power Management and ACPI Options is the CPU Frequency Scaling option - ENTER



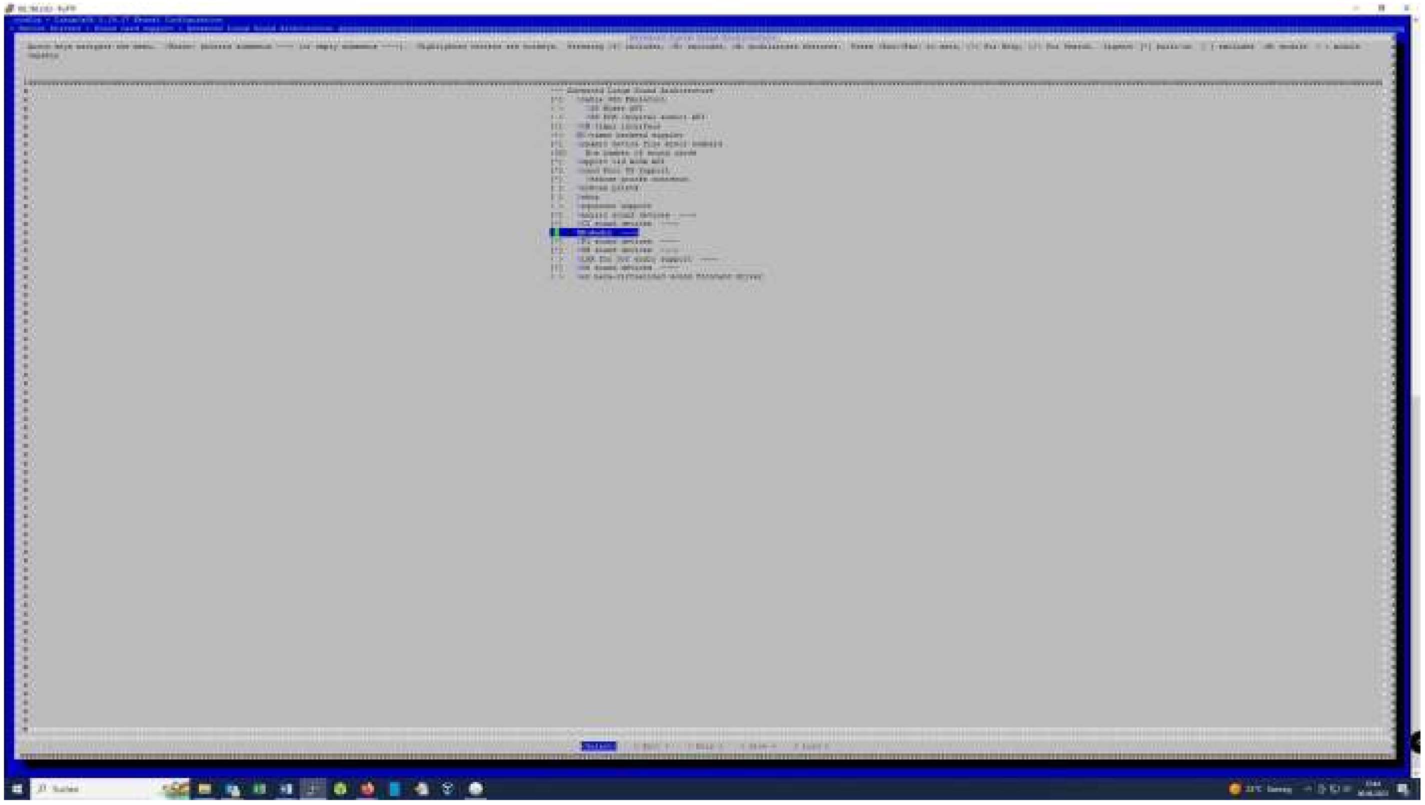
```
[*] CPU Frequency scaling
[*]   CPU frequency transition statistics
      Default CPUFreq governor (schedutil)  --->
<*>   'performance' governor
< >   'powersave' governor
<*>   'userspace' governor for userspace frequency scaling
< >   'ondemand' cpufreq policy governor
< >   'conservative' cpufreq governor
-*-   'schedutil' cpufreq policy governor
      *** CPU frequency scaling drivers ***
[ ]   Intel P state control
< >   Processor Clocking Control interface driver
<*>   AMD Processor P-State driver
<*>   ACPI Processor P-States driver
[*]     Legacy cpb sysfs knob support for AMD CPUs
< >   AMD Opteron/Athlon64 PowerNow!
< >   Intel Enhanced SpeedStep (deprecated)
< >   Intel Pentium 4 clock modulation
      *** shared options ***
```

Within Power Management and ACPI Options is the CPU Idle option - ENTER



HR-Timer enabled:

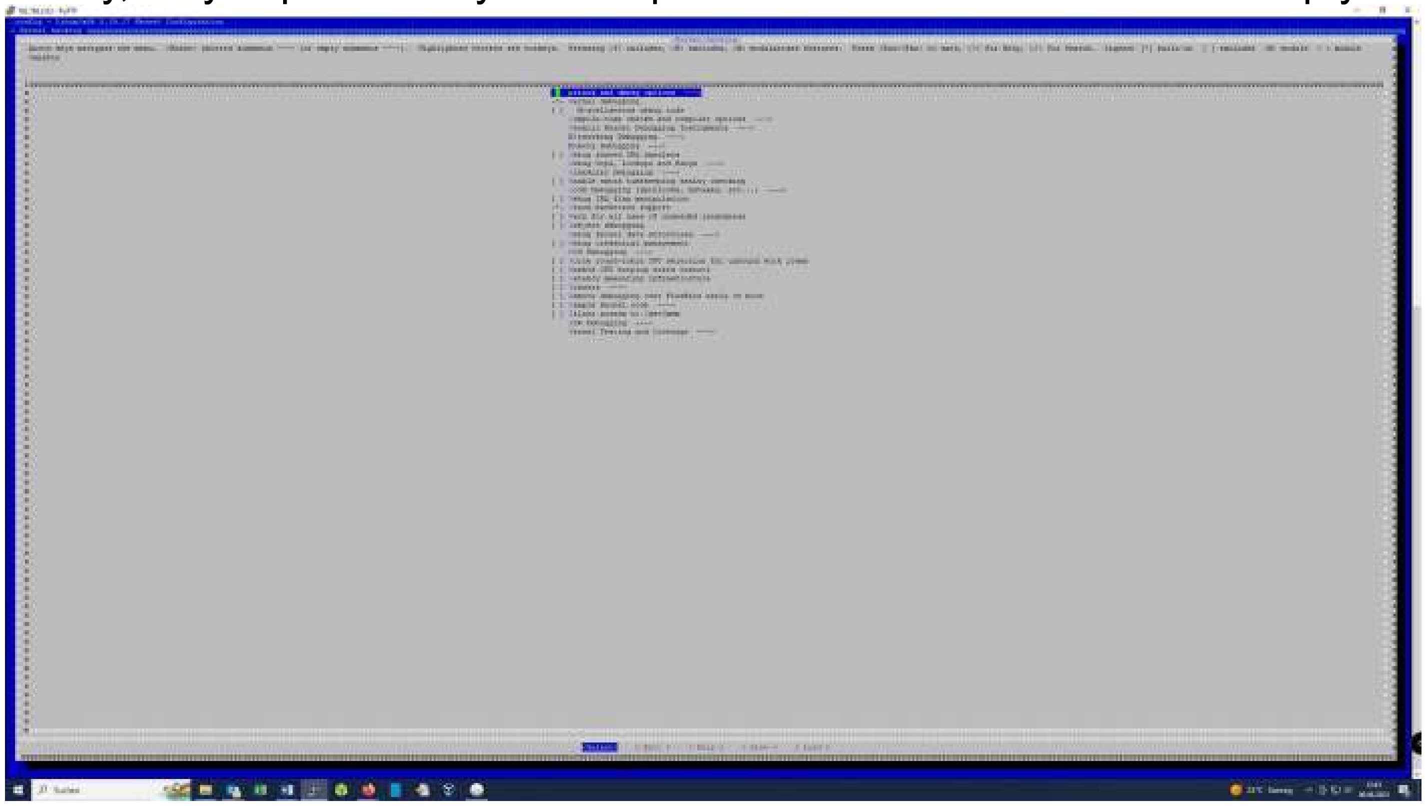
*Back to first page – Device Drivers then to Sound card support -use Y to insert the * - ENTER – Advanced Linux Sound Architecture – use Y to insert the * - ENTER*



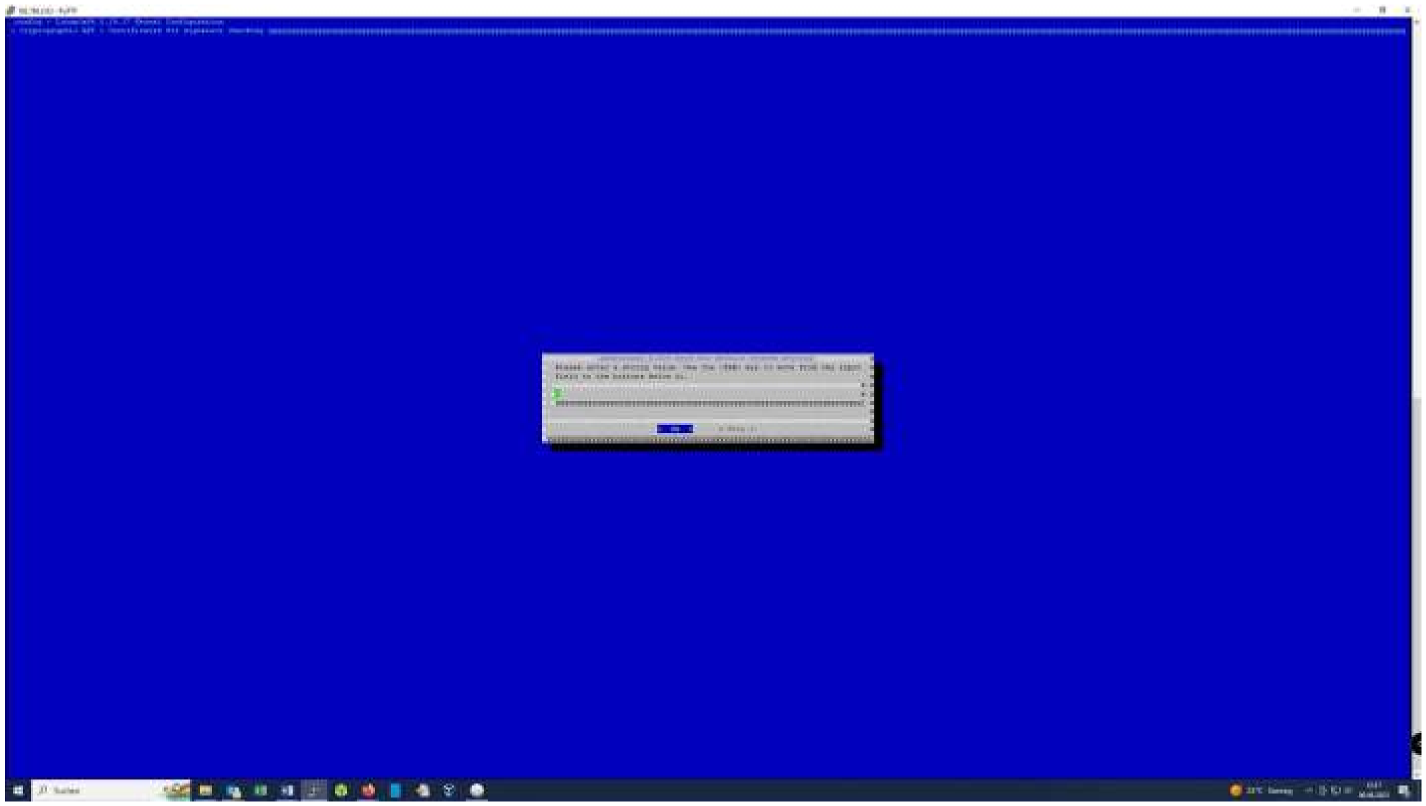
Basically no debugging overhead:

First page – look down for Kernel hacking – ENTER

Finally, very important or your compile will later die: Ensure this is empty:



This in within the first page menu – go to CRYPTOGRAPHIC APL – ENTER – then scroll down to ADDITIONAL X509 KEYS FOR DEFAULT SYSTEM KEYRING – ENTER – and clear the field - ENTER



Now, please save and continue with the following steps:
Do not think you should rename the file like I did

Once the configuration is done and saved:

make

make modules_install

make install

update-grub

reboot

Once rebooted, do

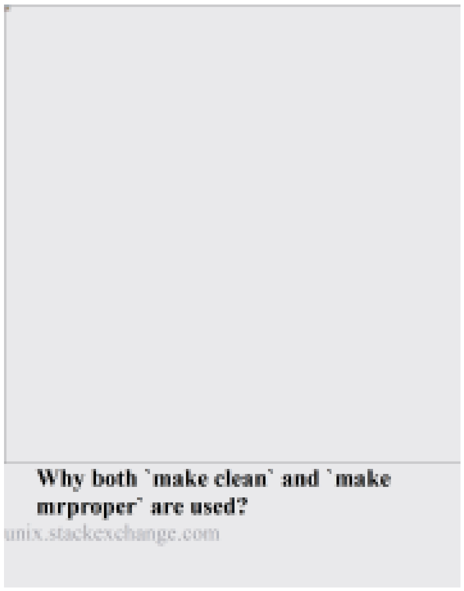
uname –r

should show you your current Kernel version and now it should be 5.19.17.

If you screw up your kernel like I did many times this will get you back to where you can begin again. If you did not skip down to below the horizontal line

Please google "Mr Proper Linux"

https://unix.stackexchange.com/questions/387640/why-both-make-clean-and-make-mrproper-are-used



Why both `make clean` and `make mrproper` are used?
unix.stackexchange.com

Use these commands

```
make clean
make mrproper
make distclean
```

will clean up the mess you created and then you should be able to compile again (In the 5.19 folder).

Then start new from scratch...and this time just do EXACTLY what Blitz advised to do. There is absolutely no room for creativity. Did I say anything about renaming something ? No.
The above is when I had to sheepishly admit I thought I was supposed to name the config file something other than the choice given

More good advice and an explanation from Blitz
We have already a new name...it is the name of the folder...linux-5.19.17...the new config file is in there, the old config file is stored somewhere completely else and wont be lost at all. No need to backup anything.

You as well have to do things EXACTLY in the sequence I described. If you miss only one step, it will not work (like the certificates).
I missed the clearing out of the certificates field the first time

If this still does not work, delete the whole 5.19. folder and start from scratch.
In my experience the cleaning process works just fine

So we come to one of the most important step…we give now the Kernel the command to make use of what we prepared.

We do a lot of stuff here, but most important we isolate the cores, activate tickless mode and use the AMD energy driver instead of the generic.

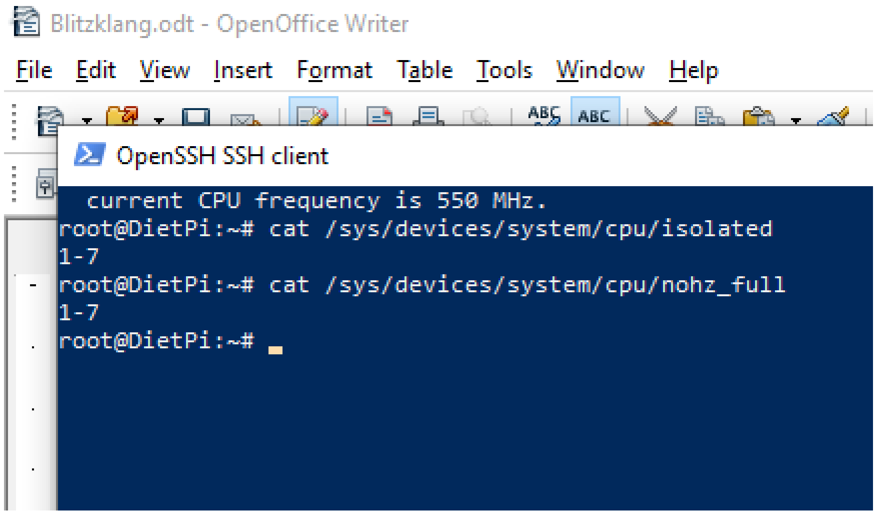
We need to change one line in your boot loader and you have to be very careful doing that or you wont have a system anymore and can start from scratch. So, PLEASE…be careful:

First let’s check your current isolation status and tickless status:

```
cat /sys/devices/system/cpu/isolated

cat /sys/devices/system/cpu/nohz_full
```

It will be probably return nothing, while when done it looks like:


Blitzklang.odt - OpenOffice Writer
File Edit View Insert Format Table Tools Window Help
OpenSSH SSH client

```
 current CPU frequency is 550 MHz.
root@DietPi:~# cat /sys/devices/system/cpu/isolated
1-7
root@DietPi:~# cat /sys/devices/system/cpu/nohz_full
1-7
root@DietPi:~# _
```

So…core 1-7 are on my machine isolated and work in full tickless mode.

You get that now activated with

nano /etc/default/grub
Yours may look differently, that is fine…**ONLY THE HIGHLIGHTED LINE with**

```
GRUB_CMDLINE_LINUX_DEFAULT="consoleblank=0 amd_pstate.shared_mem=1
mitigations=off elevator=none tsc=perfect quiet irqaffinity=0 nosoftlockup
nmi_watchdog=0 nohz=on isolcpus=nohz,domain,1-7 nohz_full=1-7 rcu_nocbs=1-7
no_balance_cores=1-7"
```

Is what you need to make look the same as mine.

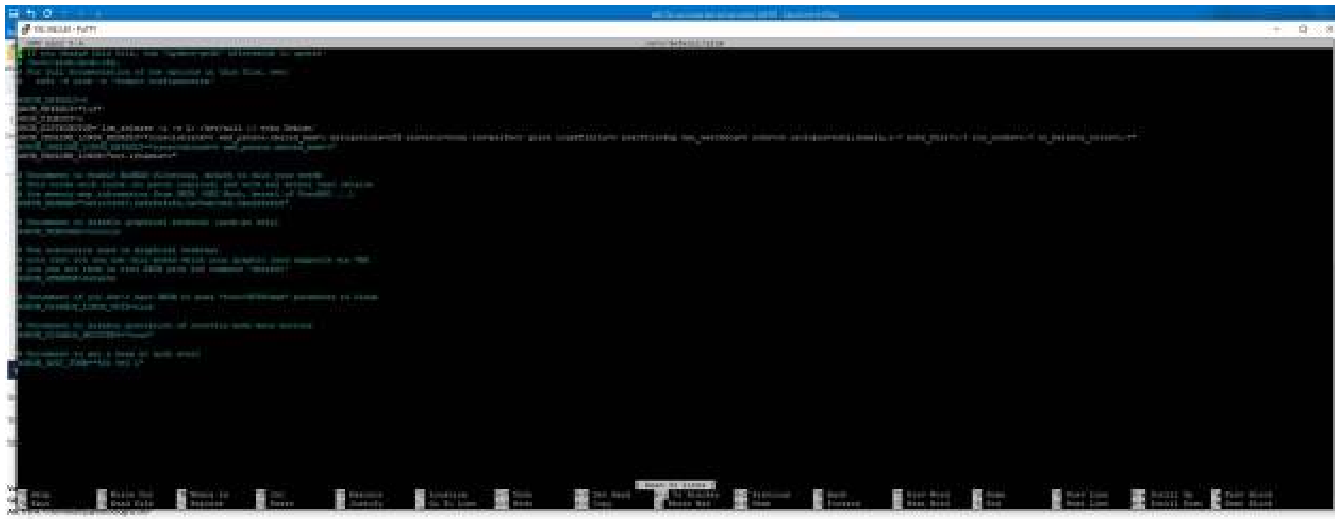
Save the file.

„update grub“

„reboot“

Check your isolation status and tickless status and run cpufreq-info as before.

It should show you we are now in business.

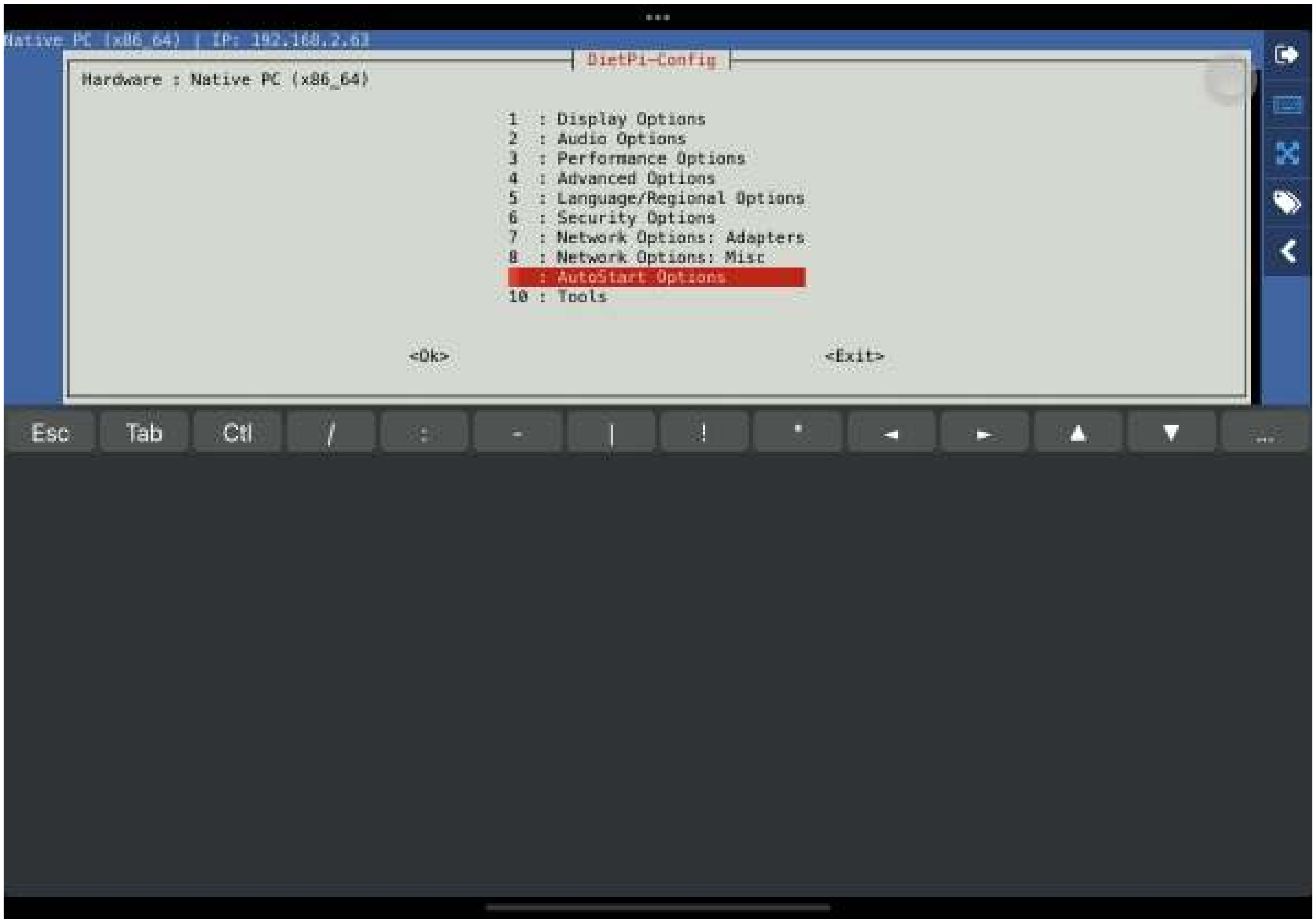


```
[illegible]
```

Ok,

lets set the frequency...

you go into dietpi-config:

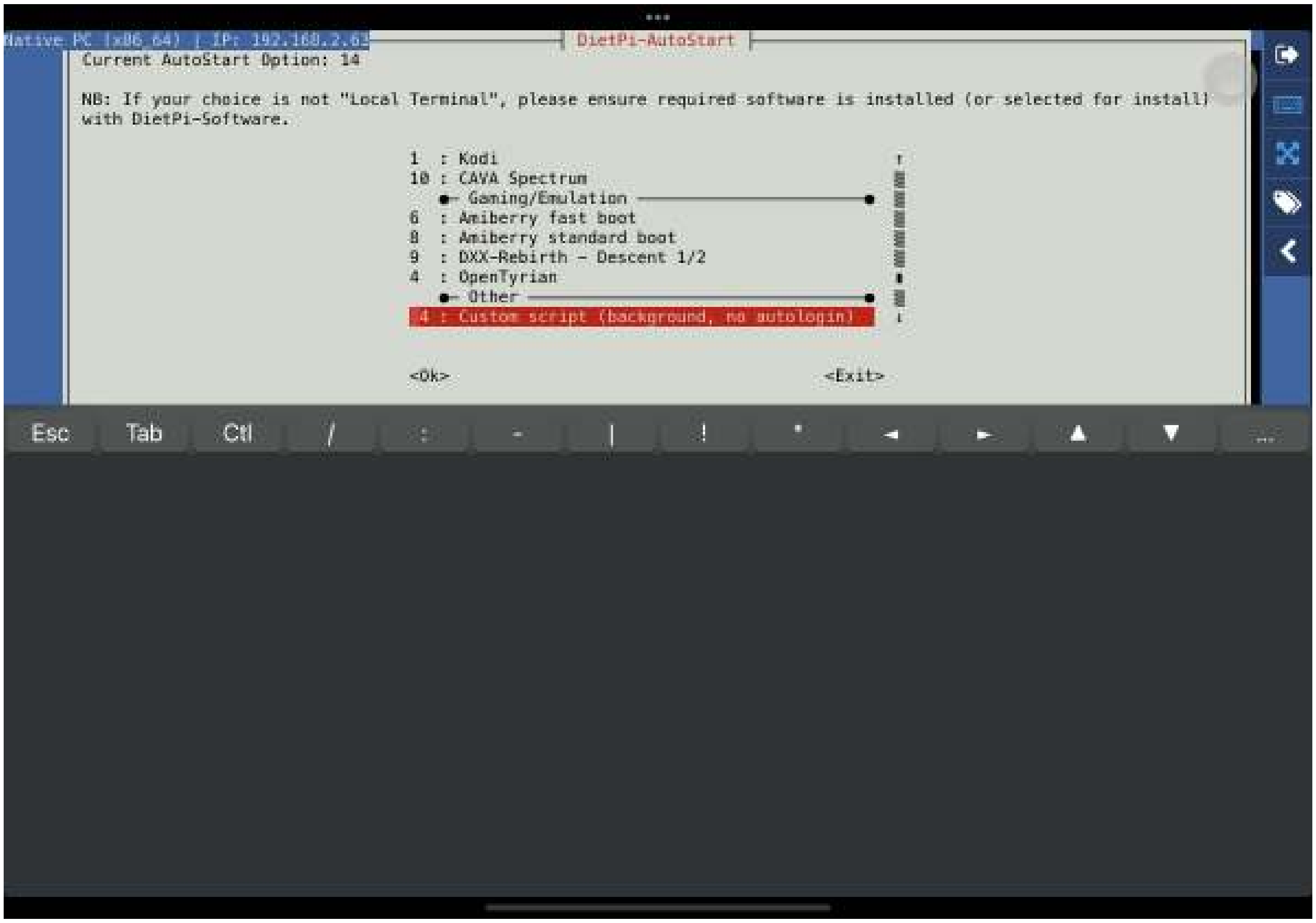


```
Native PC (x86_64) | IP: 192.168.2.63
                              DietPi-Config
Hardware : Native PC (x86_64)

                    1  : Display Options
                    2  : Audio Options
                    3  : Performance Options
                    4  : Advanced Options
                    5  : Language/Regional Options
                    6  : Security Options
                    7  : Network Options: Adapters
                    8  : Network Options: Misc
                       : AutoStart Options
                    10 : Tools

            <Ok>                              <Exit>
```

And in autostart options you select 14



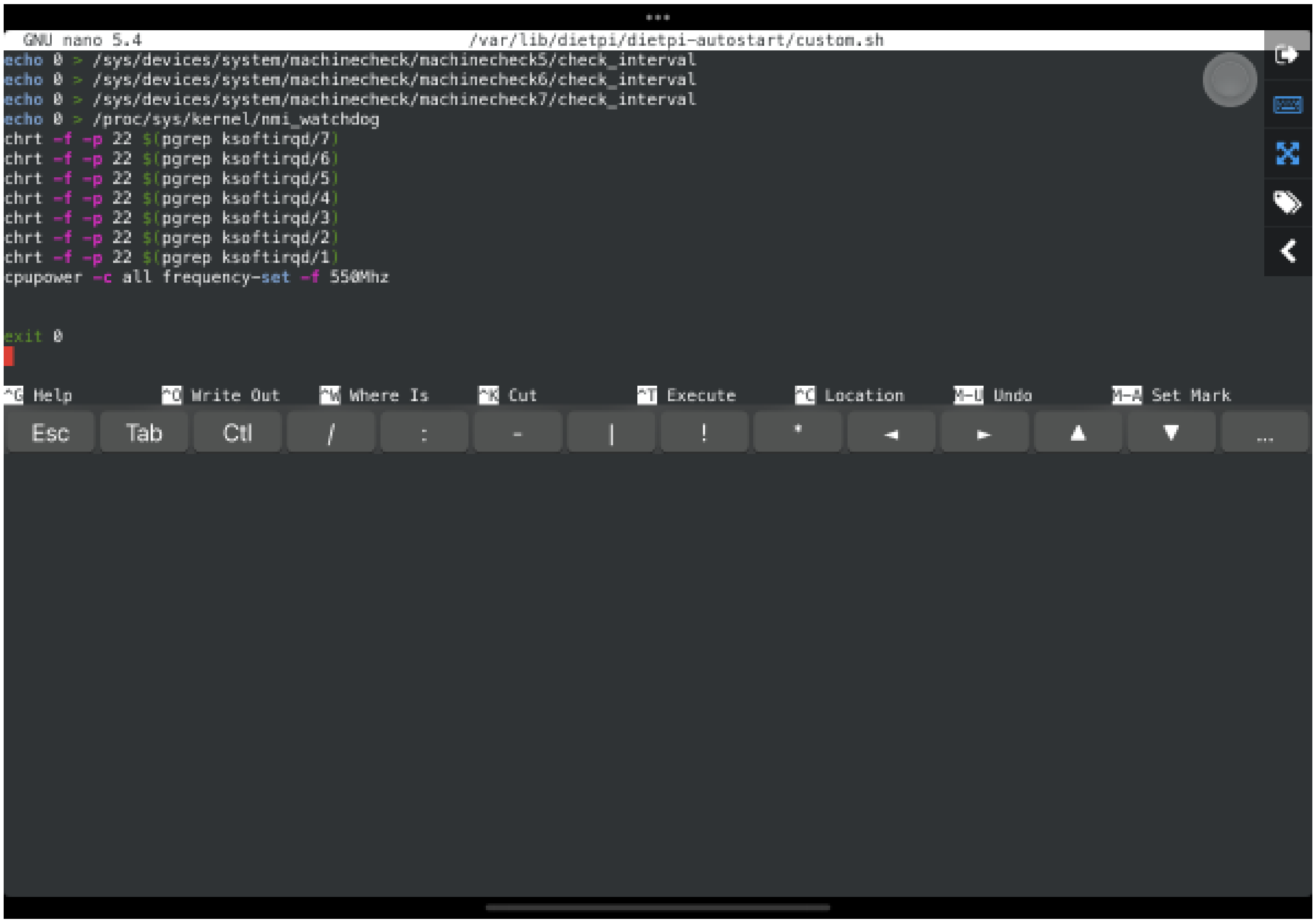
```
Native PC (x86_64) | IP: 192.168.2.61 ──┤ DietPi-AutoStart ├──
 Current AutoStart Option: 14

 NB: If your choice is not "Local Terminal", please ensure required software is installed (or selected for install)
 with DietPi-Software.

                     1  : Kodi
                     10 : CAVA Spectrum
                       ●─ Gaming/Emulation ──────────────●
                     6  : Amiberry fast boot
                     8  : Amiberry standard boot
                     9  : DXX-Rebirth - Descent 1/2
                     4  : OpenTyrian
                       ●─ Other ─────────────────────────●
                     14 : Custom script (background, no autologin)

                     <Ok>                                   <Exit>
```

and you insert for the moment only the line with
cpupower -c all frequency-set -f 550Mhz



```
GNU nano 5.4                    /var/lib/dietpi/dietpi-autostart/custom.sh
echo 0 > /sys/devices/system/machinecheck/machinecheck5/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck6/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck7/check_interval
echo 0 > /proc/sys/kernel/nmi_watchdog
chrt -f -p 22 $(pgrep ksoftirqd/7)
chrt -f -p 22 $(pgrep ksoftirqd/6)
chrt -f -p 22 $(pgrep ksoftirqd/5)
chrt -f -p 22 $(pgrep ksoftirqd/4)
chrt -f -p 22 $(pgrep ksoftirqd/3)
chrt -f -p 22 $(pgrep ksoftirqd/2)
chrt -f -p 22 $(pgrep ksoftirqd/1)
cpupower -c all frequency-set -f 550Mhz


exit 0
```

Save and reboot...

please run cpufreq-info again, make a screenshot and show me the result.

So lets go into the final round.

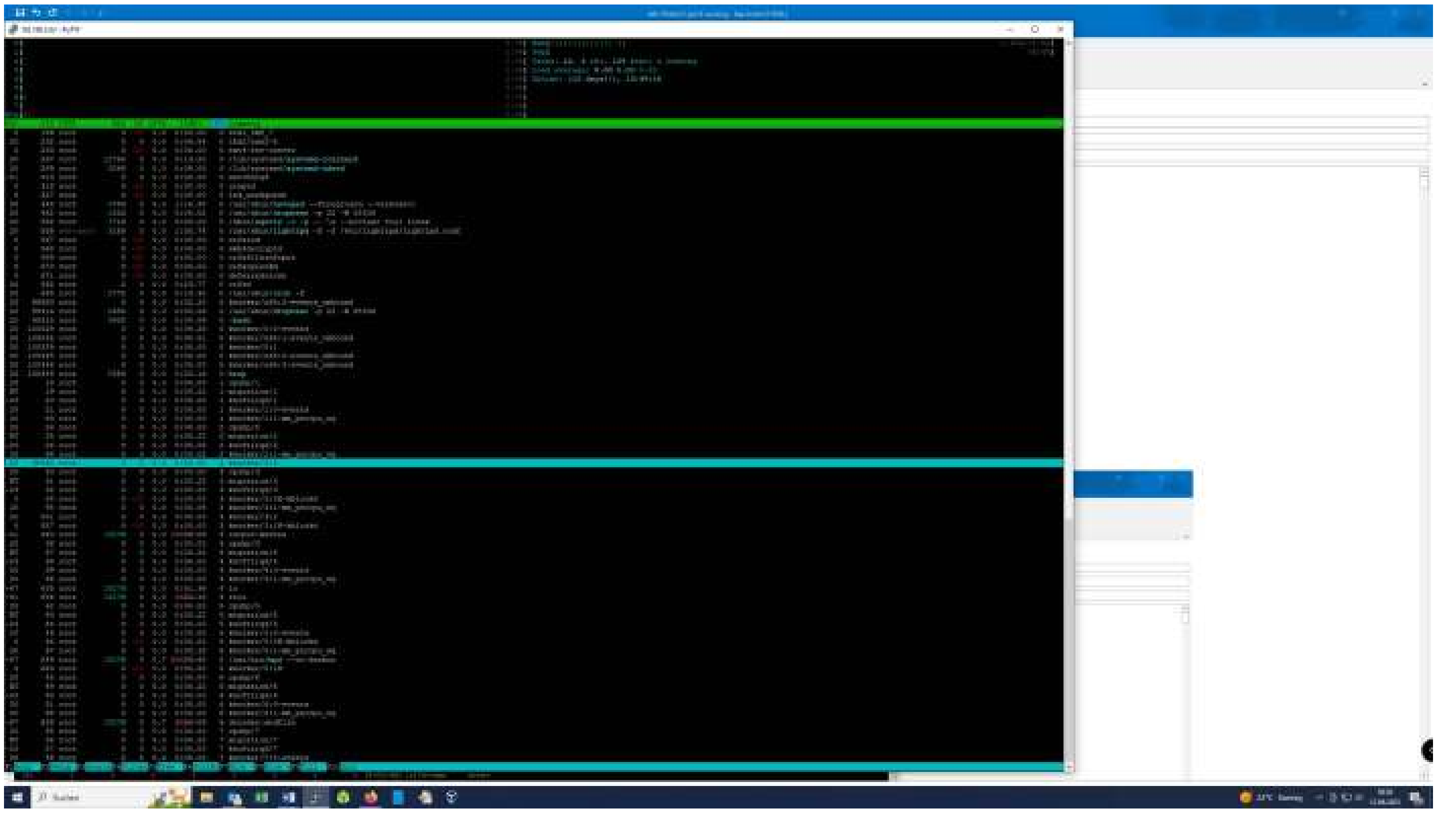
Lets remember:

We want isolated, clean CPU cores, so our audio processing is not polluted. Therefore, the CPU cores are distributed:

Core 0 – Housekeeping Core for all OS-related tasks and NON-Audio stuff
Core 1 – Reserved for LAN or SATA
Core 2 – Reserved for USB-Audio-Output
Core3-7 Reserved for MPD and its child processes

So, if you remember my HTOP screenshot in the beginning. Yours should look like that now. Please check and send me a screenshot.

It should look like this:



Well, now there is a second source of pollution besides services/apps/process: Interrupts.

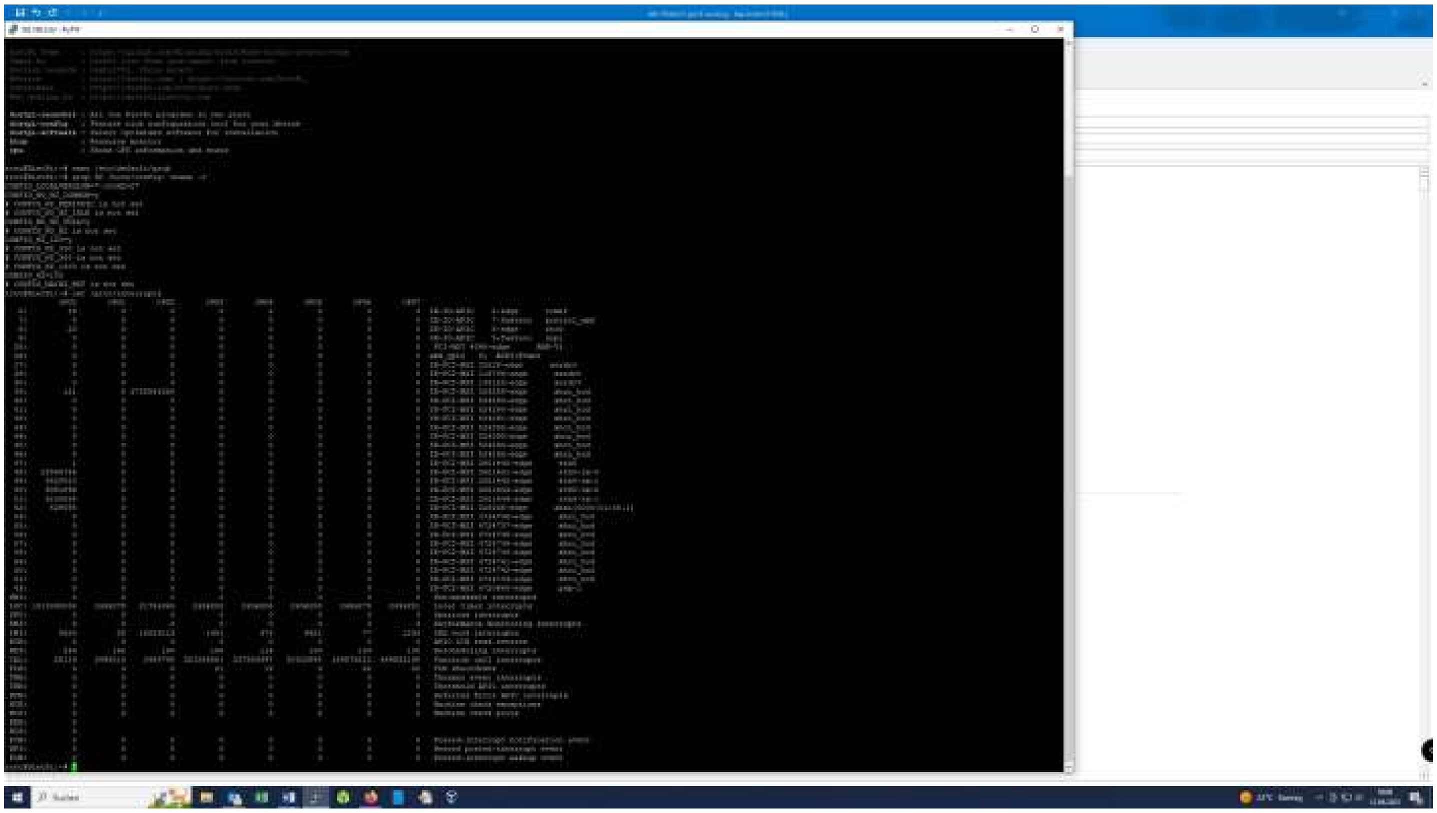
Interrupts are more hardware-near and they…well…interrupt and ask for CPU-Attention as the name says. They are not shown by HTOP.

So, we need a different tool for that to study them and they are on each PC different.

The magic scomand to show what is going on is

```
cat /proc/interrupts
```

You will see something like:



```
[illegible]
```

Ok, The trick is now:

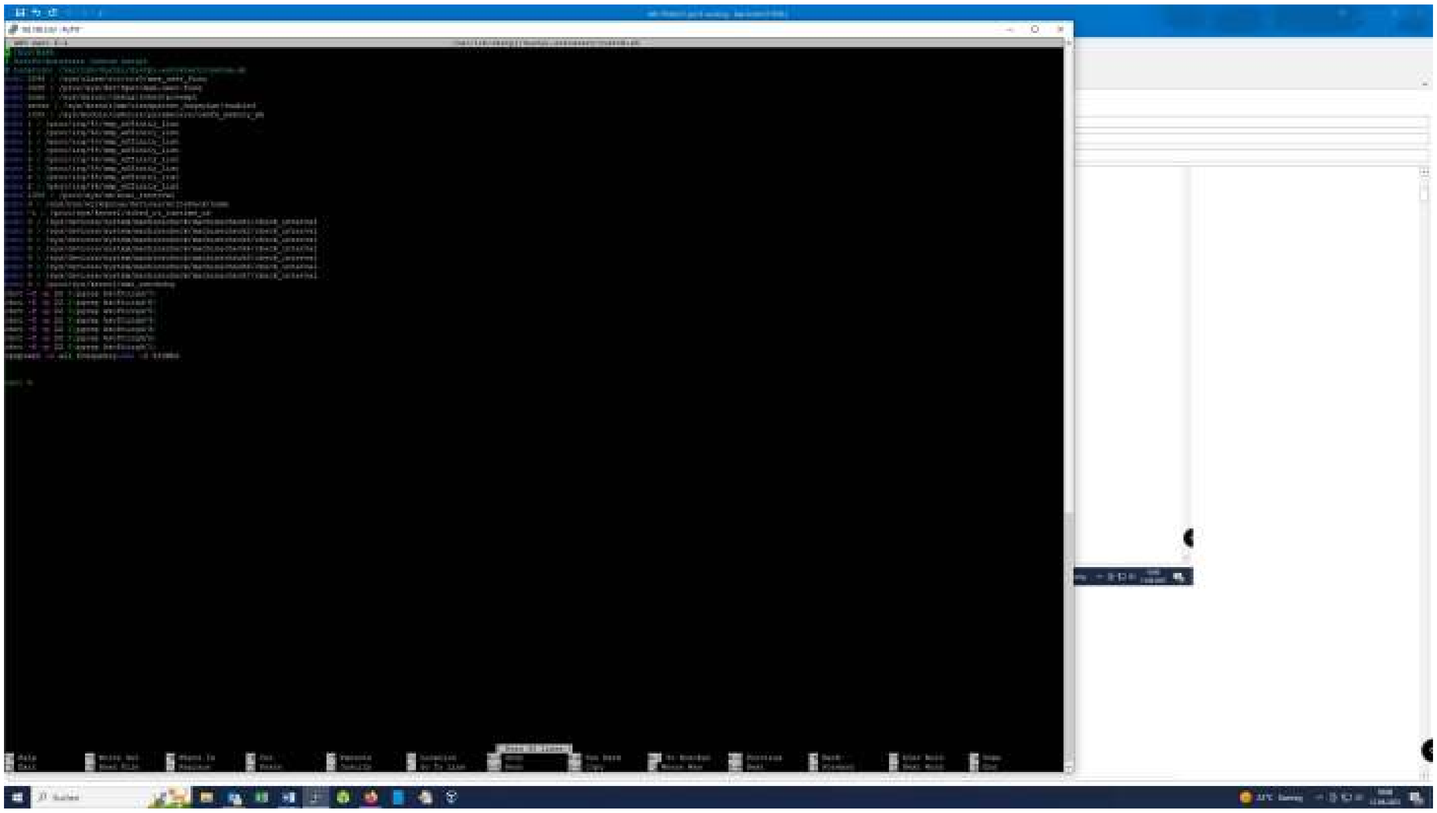
- -Restart your machine without playing any music. Type the command above and make a screenshot
- -Play a piece of music, make the command again and make a screenshot.
- -Wait 10 sec, Play a different piece of music and make a screenshot.
- -Send me those screenshots and lets analyze them.

If you look at my example above you see:

- -I have put ethernet even on core 0 since it is not audio relevant anymore, we use the input cache of MPD and play from ram.
- -Audio-USB is playing on core 2
- -All other stuff is on Core 0
- -Some housekeeping interrupts which are necessary to have a functional CPU, remain on all cores.

So, how did I achieve that ?

My Autostart-script in Dietpi Option 14 looks like this:



[illegible]

So, you see that I send interrupt 39 (and some other USB-Interrupts) to core 2 and now it runs on core 2.

Your numbers maybe different !!! And if you use a different USB-Port, the interrupt number may change, that is why I have specd more than only one interrupt for core 2…different USB-ports.

The other statements which have no irq in it are optimization statements for further audio improvements which came from different other Audio-PC-Projects. You can copy them into your autostartfile.

To make it more convenient for you, here is the ascii text of my file:

```
#!/bin/bash
# DietPi-Autostart custom script
# Location: /var/lib/dietpi/dietpi-autostart/custom.sh
echo 2048 > /sys/class/rtc/rtc0/max_user_freq
echo 2048 > /proc/sys/dev/hpet/max-user-freq
echo none > /sys/kernel/debug/sched/preempt
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo 1000 > /sys/module/usbcore/parameters/usbfs_memory_mb
echo 1 > /proc/irq/41/smp_affinity_list
echo 1 > /proc/irq/42/smp_affinity_list
echo 1 > /proc/irq/43/smp_affinity_list
echo 1 > /proc/irq/44/smp_affinity_list
echo 2 > /proc/irq/39/smp_affinity_list
echo 2 > /proc/irq/54/smp_affinity_list
echo 2 > /proc/irq/45/smp_affinity_list
echo 2 > /proc/irq/46/smp_affinity_list
echo 1000 > /proc/sys/vm/stat_interval
echo 0 > /sys/bus/workqueue/devices/writeback/numa
echo -1 > /proc/sys/kernel/sched_rt_runtime_us
echo 0 > /sys/devices/system/machinecheck/machinecheck1/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck2/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck3/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck4/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck5/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck6/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck7/check_interval
echo 0 > /proc/sys/kernel/nmi_watchdog
chrt -f -p 22 $(pgrep ksoftirqd/7)
chrt -f -p 22 $(pgrep ksoftirqd/6)
chrt -f -p 22 $(pgrep ksoftirqd/5)
chrt -f -p 22 $(pgrep ksoftirqd/4)
chrt -f -p 22 $(pgrep ksoftirqd/3)
chrt -f -p 22 $(pgrep ksoftirqd/2)
chrt -f -p 22 $(pgrep ksoftirqd/1)
cpupower -c all frequency-set -f 550Mhz
```

exit and save

Have fun..I think we are done…

let’s check if everything works as expected…please send me the screenshots specified above (as well from HTOP).
…and let me know what you hear…

Below is what I am using in my installation

```
#!/bin/bash
# DietPi-Autostart custom script
# Location: /var/lib/dietpi/dietpi-autostart/custom.sh
echo 2048 > /sys/class/rtc/rtc0/max_user_freq
echo 2048 > /proc/sys/dev/hpet/max-user-freq
echo none > /sys/kernel/debug/sched/preempt
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo 1000 > /sys/module/usbcore/parameters/usbfs_memory_mb
echo 1 > /proc/irq/41/smp_affinity_list
echo 1 > /proc/irq/42/smp_affinity_list
echo 1 > /proc/irq/43/smp_affinity_list
echo 1 > /proc/irq/44/smp_affinity_list
echo 2 > /proc/irq/40/smp_affinity_list
echo 2 > /proc/irq/54/smp_affinity_list
echo 2 > /proc/irq/45/smp_affinity_list
echo 2 > /proc/irq/46/smp_affinity_list
echo 1000 > /proc/sys/vm/stat_interval
echo 0 > /sys/bus/workqueue/devices/writeback/numa
echo -1 > /proc/sys/kernel/sched_rt_runtime_us
echo 0 > /sys/devices/system/machinecheck/machinecheck1/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck2/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck3/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck4/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck5/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck6/check_interval
echo 0 > /sys/devices/system/machinecheck/machinecheck7/check_interval
echo 0 > /proc/sys/kernel/nmi_watchdog
chrt -f -p 22 $(pgrep ksoftirqd/7)
chrt -f -p 22 $(pgrep ksoftirqd/6)
chrt -f -p 22 $(pgrep ksoftirqd/5)
chrt -f -p 22 $(pgrep ksoftirqd/4)
chrt -f -p 22 $(pgrep ksoftirqd/3)
chrt -f -p 22 $(pgrep ksoftirqd/2)
chrt -f -p 22 $(pgrep ksoftirqd/1)
cpupower -c all frequency-set -f 550Mhz
```

exit and save