CUSTOM BRIGHTNESS ADJUSTMENT TOOL

My laptop specs:

- Model: HP Victus 15 fb0050ax
- NVIDIA GeForce RTX 3050 dedicated GPU
- AMD Radeon Graphics Integrated GPU
- AMD Ryzen 5 5600H Processor
- 8 GB RAM

Issue: Can't adjust the brightness of my laptop using the vendor specified keys i.e. Fn + F2 to decrease brightness and Fn + F3 to increase brightness in Linux distros.

Solution:
In every Linux system there is a folder where all brightness related files are stored. It generally exists in /sys/class/backlight directory.

As my laptop is currently taking output from AMD Radeon iGPU, my path of this folder is:

/sys/class/backlight/amdgpu_bl1/

Note: Name of the folder can change. Here it was amdgpu_bl1. In Intel processors it will be different.

In this directory all settings and statuses are there like what should be max brightness of display, and what is the current brightness of my screen.

In this directory there is a file called as 'brightness' which is the file we are concerned about. This file stores the current brightness value of your display.

As you press your key combination, if try to decrease the brightness, the changes get reflected here, i.e the value gets decremented by some value and vice versa.

Now the issue in my laptop was that the values in the file were changing as expected but the actual brightness of my laptop was not changing. I could see the brightness slider saying that you're successful in changing the brightness but in reality, it was not changing the actual brightness.

So, after research I found out that there is a command xrandr upon using it I was able to change the actual brightness of my screen. But one note to make here. On using xrandr to change the brightness value, you can't update the contents of brightness file which I talked about earlier.

On running xrandr command you get all information related to your display including the name of your display. You should make a note of it in order to do change brightness using xrandr. The exact command is:

xrandr --output display_name --brightness brightness_value

By using this you can change the brightness of your laptop. There are other tools too, but other tools failed and this tool was my savior.

Now let's really mix things up. Trust me, it's very interesting and simple as well. You'll love the mathematics after this.

Take the brightness file (that exists in /sys/class/backlight/…). The file has some upper bound and lower bound of the value that it can have. The upper limit is of 255 and lower limit is of 2.

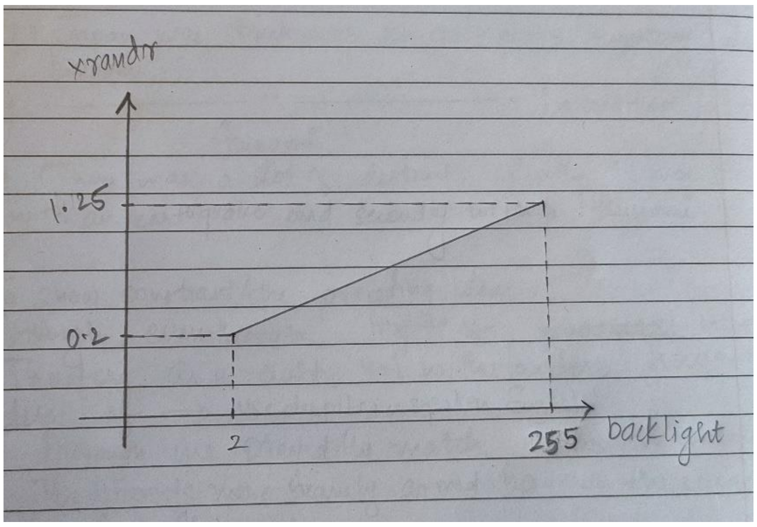
Similarly, I have kept my personal upper and lower bounds to xrandr as well. So, the upper limit xrandr can use is 1.25 and lower limit that xrandr can use is 0.2

Now here the things start getting interesting! Let backlight be brightness value in file and xrandr be value which I'm going to use with command.

For backlight = 2, I want xrandr = 0.2
For backlight = 255, I want xrandr = 1.25

And between these upper and lower bounds, I want a linear variation in the brightness of my screen. And for linear variation, as you know, there is nothing better than a straight line.

So, the resultant graph is,



So, by using point-slope form of a straight line,

$$\frac{xrandr - 1.25}{backlight - 255} = \frac{1.25 - 0.2}{255 - 2}$$

On simplification, the equation comes out to be,

$$\text{xrandr} = 0.00415019763(\text{backlight} - 255) + 1.25$$

On backlight = 255, xrandr = 1.25
backlight = 2, xrandr = 0.2

So here you go, you get value which you want to use with xrandr which varies linearly with respect to backlight.

Now how did I use this equation to solve my problem? It's quite simple.

Step 1 is to open the brightness file. I'll also create and open a previous brightness file which stores the previous brightness value i.e old value.

Store the value of brightness file in current_b and previous brightness value in prev_b.

Now when should I update my screen brightness? Obviously when it gets changed. So, I declared another variable called as change.

$$\text{change} = \text{current\_b} - \text{prev\_b}$$

So, if change ≠ 0 it means that the value of brightness file has changed. It means it is time to change our screen brightness too. For this after calculating change, I used an if condition that,

if (change ≠ 0) {

// Calculate the value to be used with xrandr using the formula and use the command and change the screen brightness.

```
        // Similarly update the value of previous brightness file so that we can calculate the change whenever the program runs again.

}
```

Now the code works fine. It changes the value of brightness when it detects change. But here's another catch. The program will only run once and then exit. We want it to run it the entire time our laptop is running. So, for this purpose we use an infinite loop. The infinite loop ensures that the program runs till our laptop is running. So, to run a while loop infinitely:

```
while (true) {

        // Code here will run infinitely until and unless you don't terminate the program using Ctrl+C

}
```

So, we've successfully created a program which runs infinitely and gives us assurance that it'll change brightness for us whenever it detects a change.

But each time when you reboot your system, how will the program start automatically? It can't automatically start. You need to start it manually. But as you know, programmers are lazy people. They like to automate stuff as much as possible. To automate it, you need to create a desktop file.

A desktop file is used to define the application shortcuts and launchers. Like in windows we can create shortcuts to applications; similarly in Linux that feature is achieved with .desktop files.

So, I created a desktop file which upon running, executes our C++ program. So, I got a tool to execute the program without typing the command. I can simply click on the desktop file icon and get my brightness working. Being more lazy, I thought what if I even automate this launching step. Like on every startup I have to manually click on that application desktop file to start my program. Automating this will save that effort.

To automate that, I used a feature of GNOME desktop environment. GNOME provides a tool called as GNOME Tweaks which you can use to tweak your system. In that app there is a feature called as 'Startup Applications'. Basically, the feature helps you to start an app as soon as the computer starts. So, I added my application to this list (Startup applications was also one of the reasons why I created a desktop file because you can't add a normal C++ compiled program to this list). This is it! Using it, I've fully automated the starting process of my application. I can now normally change my brightness using the vendor specified keys.

This project basically acts as an bridge between the brightness file and the xrandr command to change the brightness of my laptop screen.