

Refining my tablet OS experience

Using Surface Go 2 with 4G/LTE, Debian GNU/Linux and i3wm

Willem L. Middelkoop

Sep. 28, 2020



Earlier this year I tried to create my own tablet operating system by installing Debian GNU/Linux on a Microsoft Surface Go tablet. I learned a great deal about what I like about tablets. But I still preferred my iPad Pro, mainly because of its polished user experience. This time I set out to refine my tablet software to replace my iPad.

Linux on (Surface) tablets

If you're thinking about installing Linux on a tablet: you're not alone. My earlier post on [installing Debian GNU/Linux on Surface GO](#) received a ton of attention. Apparently, many of you are looking for an alternative for the locked down operating systems that are usually installed on tablets (like iPadOS, Android or Windows 10S).

Microsoft Surface hardware and Linux

As history is full of sweet irony, it's Microsoft that actually makes some pretty handsome hardware that works well with Linux. Their Surface line-up includes different laptops and tablets. The [“SurfaceLinux”](#) reddit page is full of interesting user stories worth checking out. People are doing this and you could do it yourself, too.



Microsoft Surface hardware is popular because it's genuinely nice

First attempt: Surface Go (versus iPad Pro)

After [installing and modifying Debian GNU/Linux on Surface Go](#), dubbed “Willem OS”, I tried to make it my main computer. If you're a frequent visitor of my blog, you'll know that [I have been using iPads and tablets for years as main computer](#). Ultimately I deemed my first version of “WillemOS” not good enough: iPad Pro was light-years ahead when it came to a polished user experience.

From this first attempt I [learned](#) a great deal, it led to a better understanding of what I seek in a tablet experience. **I am looking for simple, yet versatile software to run on my tablet.**

But “simple” is actually very hard to achieve. In order to make things *really* simple, you need to seamlessly integrate hardware and software. Apple does this well, iPad Pro provides a great user experience because:

- **continue where you've left:** it saves everything when you put it in a [bag](#), being (instantly) ready to continue when you take it out. No manual saving or (re)booting required.
- **internet everywhere:** it automatically switches between [WiFi and 4G/LTE networks](#), you don't need to do anything this ‘just works’
- **any orientation and position:** iPad functions great without an external [key-board](#) or mouse, allowing you to use it in many different scenarios (the display automatically adjusts its brightness and white balance to match your surroundings, too!)

It turns out that getting these things working seamlessly with Debian GNU/Linux on Surface is hard. My first tablet OS required a lot of manual interaction and was therefore cumbersome in comparison to iPad Pro.

Second attempt: Surface Go 2

Microsoft released a new Surface Go tablet this year, the Surface Go 2. It has a faster processor (up to 64%!), thinner bezels and an improved screen. It is a very descent upgrade that puts the Surface Go 2 much more on par with the 11-inch iPad Pro in hardware terms. For my second attempt to create the ultimate tablet experience, I decided to use the top model Surface Go 2:

- 10.5" touch screen (1920x1250, 220PPI), 3:2 aspect ratio, Corning® Gorilla® Glass 3
- Intel Core m3 processor (8th gen)
- 8GB RAM
- Wifi 802.11a/b/g/n/ac/ax
- LTE Advanced (Qualcomm® Snapdragon™ X16 LTE Modem)
- Bluetooth 5.0
- Intel UHD Graphics 615
- Dimensions: 245 mm x 175 mm x 8.3 mm
- 256GB SSD
- 512GB MicroSDXC card
- 1x USB-C and 1x Surface Connect port



Microsoft Surface Go 2 with black type cover and Surface Pen - a nice piece of hardware!

Installing Debian GNU/Linux on Surface Go 2

The Surface Go 2 is very similar to its predecessor when it comes to installing a base installation of your favourite GNU/Linux distribution. I followed my own steps from the original Surface Go installation [here](#).



Installing Debian GNU/Linux on Surface Go 2 is very similar to the original Surface Go

In short, you must take the following steps when installing Linux on your Surface:

- 1) Update Windows 10 and install all the firmware patches/updates
- 2) Create a bootable installation medium containing your favourite GNU/Linux distribution
- 3) (Optional) connect a wired ethernet connection
- 4) Boot the Surface from the USB drive
- 5) Install the operating system using the installer
- 6) Finetune and optimise the software experience



Debian GNU/Linux with i3wm ("aka WillemOS") running on Surface Go 2

Refining the experience

The last step of the installation is where you can make the difference between a fantastic and lousy user experience. To “*encourage*” myself to stick to this experiment I decided to sell my iPad Pro: burning the ships (as there would be no way back)!



Selling your iPad Pro is easy if you keep the original boxes - I sold mine in just a few days: enforcing myself to make the new tablet work!

On a small display, running apps full screen makes perfect sense. I know that the conventional PC concept of managing distinct application Windows doesn't provide the best user experience on a small tablet computer. Therefore I decided to try a tiling window manager: [i3wm](#).

Tiling window manager

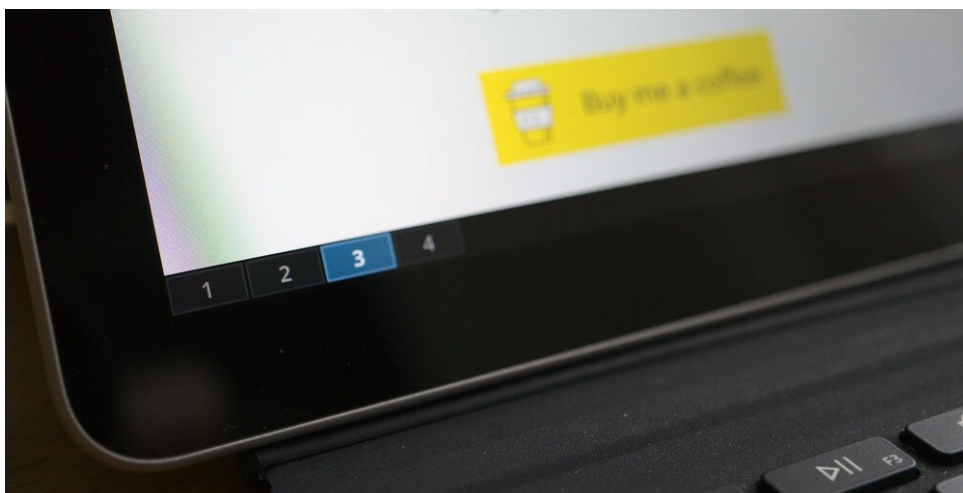
The principle of a tiling window manager is simple: apps take up as much space as there is available. This means that if you run just one app, it will be full screen. Two apps run split screen, etc, etc. You can define rules that automate the arrangement of apps. Using keyboard shortcuts you can quickly switch between apps. You may need some time to figure out what works best for *you* when you use a tiling window manager, but in the end it can save you a lot of time!



Running Firefox (left) and a MOSH/tmux terminal window (right) in splitscreen using the i3wm tiling window manager

Workspaces

Another powerful feature of i3wm is that you can have different ‘workspaces’. They are an easy way to organise groups of apps running together. I use different workspaces for different things that I do. A common approach is to put the web browser on one workspace, mail applications on another one, and the ones with which you work, on the third one.



Multiple workspaces are very powerful: it's like switching TV channels

The great thing about multiple workspaces is that one app (e.g. your browser) can run multiple instances on different workspaces. This enables many different combinations

of apps. This is really where the i3wm window manager is *much* better than iPadOS's take on multi tasking.

The i3wm window manager supports external monitors very well, using the same concept of a workspace. You simply designate one or multiple workspaces to be shown on the extra/external screen. This works well with the Surface Dock and USB-C-to-HDMI accessories for Surface. Multi monitor support is really great.



Three finger swipe to switch workspaces - a trick from macOS - using [fusuma](#) you can use multi touch gestures in i3wm

One thing that you should try is to use [fusuma](#) to enable three finger gestures on the trackpad. If you have ever worked on a MacBook you'll know that the multi touch trackpads enable useful gestures to switch desktops (or workspaces in i3wm). It works really well!

Storage

For [managing my photos using Shotwell](#) I wanted some serious storage on my tablet. Although I have the top model Surface Go 2 with the 256GB SSD drive, I used the MicroSD card slot to extend my storage capacity by another 512GB. The MicroSD card is slower than the integrated SSD, but for files that you don't read every day/hour/minute it is fast enough.



Extending storage on a Surface Go 2 is possible using a MicroSD card

My Debian GNU/Linux installation had no trouble recognising the MicroSD card. I encrypted the entire MicroSD card to prevent my photos from leaking if I would ever lose my card. You should (always) consider this, take appropriate measures to protect and backup your data!

Connectivity WiFi/4G/LTE

After installing the ‘non-free’ wireless firmware in Debian the WiFi works well on Surface Go 2. Getting the LTE/4G to work was a lot harder! For some reason the WWAN connection did not start, even though I inserted a working SIM-card in the Surface Go. For some mysterious reason the modem did not start correctly, 4G/LTE did not work!

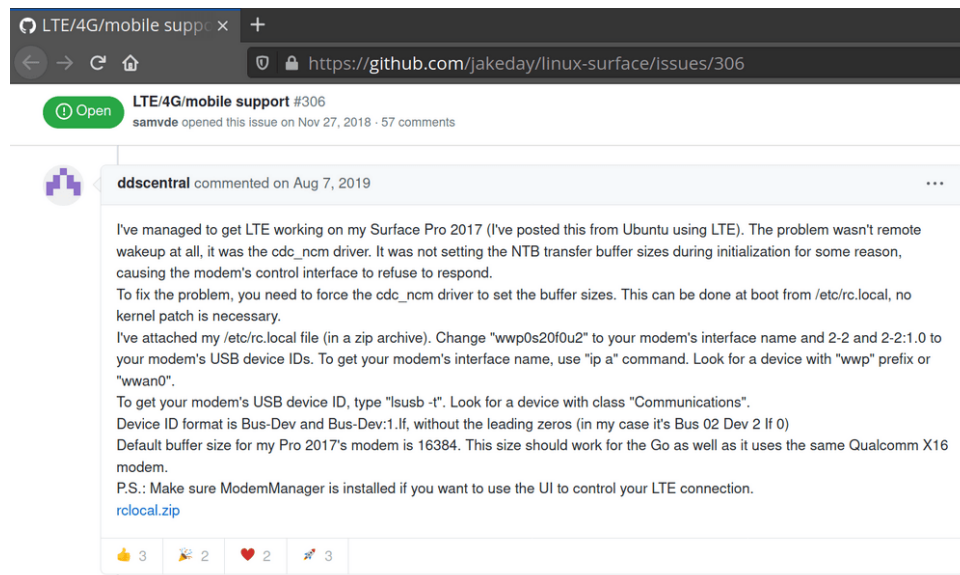


Installing the SIM in the Surface Go 2 tablet

One of the great things of iPad is that it shares the connectivity logic from iPhone.

It automatically manages different wireless networks and mobile broadband connections. I wanted this to work on my Linux tablet, too!

When I looking for a clue to get 4G/LTE to work, I looked into reports on Surface Pro. The Surface Go 2 is technically very similar to the larger Pro. I figured that I could learn from peoples efforts with mobile broadband on the Surface Pro. After many hours of searching and trying different things, I found [this solution to get 4G to work on Surface](https://github.com/jakeday/linux-surface/issues/306):



Resetting the buffers of the Surface broadband modem allowed it to work with Debian GNU/Linux!

You can use the Network Manager (and Modem Manager) to automate connectivity management under Linux. By default it will prefer WiFi connections over mobile broadband. You can have it automatically connect over 4G/LTE if there is no WiFi available. It now works very well in combination with the Surface Go 2 hardware!



Network Manager (nm-applet) automatically switches between WiFi and mobile broadband connections

Suspend/Hibernate/Resume

Another thing no iPad user ever thinks of, is how great Apple got things right when it comes to suspend and resume. When you have an iPad you can have it in standby mode for many days, yet it is always very quick to continue where you have left once you pick it up again.

To make my Surface with GNU/Linux do the same I needed a way to quickly suspend and resume. In Linux / PC world there are basically four power modes:

- **on:** apps are running. System is fully powered on, consumes a lot of power
- **suspend (to RAM):** app states are preserved in RAM memory. System is still powered on, but every component is turned down as much as possible, still consumes power (+/- 10% battery drain per hour in my case with Surface Go 2)
- **suspend (to disk) / hibernate:** app states are preserved on disk. System is powered off and does not consume any power
- **off:** apps are closed. System is powered off, no power is consumed

In order to “instantly resume where you have left” you need to preserve app states. The fastest (and near instant) resumes are possible when the app state is stored in RAM memory. The problem with RAM memory is that it needs current (power!) to maintain it's value. Once power is lost, app states are lost.

The alternative is to suspend to disk (or hibernate), where the state is stored on the permanent storage device in the computer. It will allow the computer to turn off completely. The problem with suspending to disk is that resuming is slower and will require re-entering encryption passphrases for full disk encryption.

The good news is that modern GNU/Linux comes with systemd that enables you to configure how the computer handles suspend and resume. You can configure your tablet to first suspend to RAM (enabling instant resume) and then, after a period of time, suspend to disk.

```
# See systemd-sleep.conf(5) for details

[Sleep]
AllowSuspend=yes
AllowHibernation=yes
AllowSuspendThenHibernate=yes
#AllowHybridSleep=yes
#SuspendMode=
#SuspendState=mem standby freeze
#HibernateMode=platform shutdown
#HibernateState=disk
#HybridSleepMode=suspend platform shutdown
#HybridSleepState=disk
#HibernateDelaySec=86400      # 24 hour
HibernateDelaySec=7200
```

The systemd setting “AllowSuspendThenHibernate” in sleep.conf enables you to define a timeout between suspending to RAM and suspending to disk (hibernation)

On a normal working day, I wake up (very) early to work on my most challenging projects from home. Then I [ride my bike to work](#) and continue working from there. I timed my commute and figured that if my tablet would stay on for at least two hours, it would allow me to instantly resume work. If I do not continue working on my tablet within two hours, it will save battery power by suspending to disk. This way I have the best experience for my particular situation.

```
# See logind.conf(5) for details.

[Login]
HandleLidSwitch=suspend-then-hibernate
HandleLidSwitchExternalPower=suspend-then-hibernate
```

Configure the LidSwitch (in systemd’s login.conf) to automatically suspend and resume when you close the Surface type cover, I have configured it to suspend-then-hibernate using a timeout

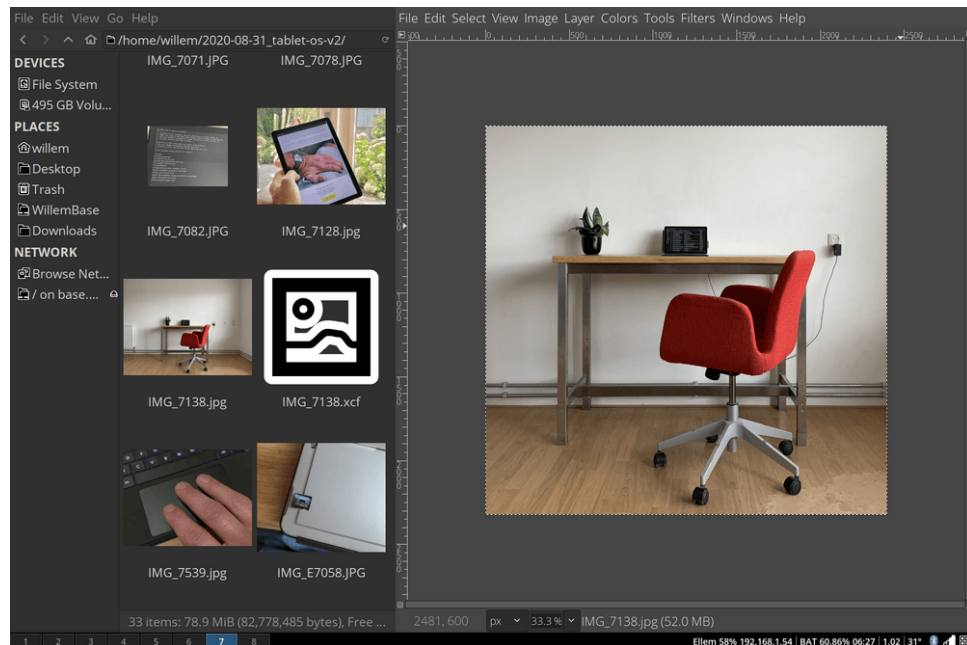
Uniform appearance

Unlike the tightly controlled iPad, software on GNU/Linux is created using many different approaches to user interface design. There is no such thing as the [User Interface Guidelines](#) that Apple dictates for iOS and iPadOS. This is a problem as it causes software to look and feel inconsistent.

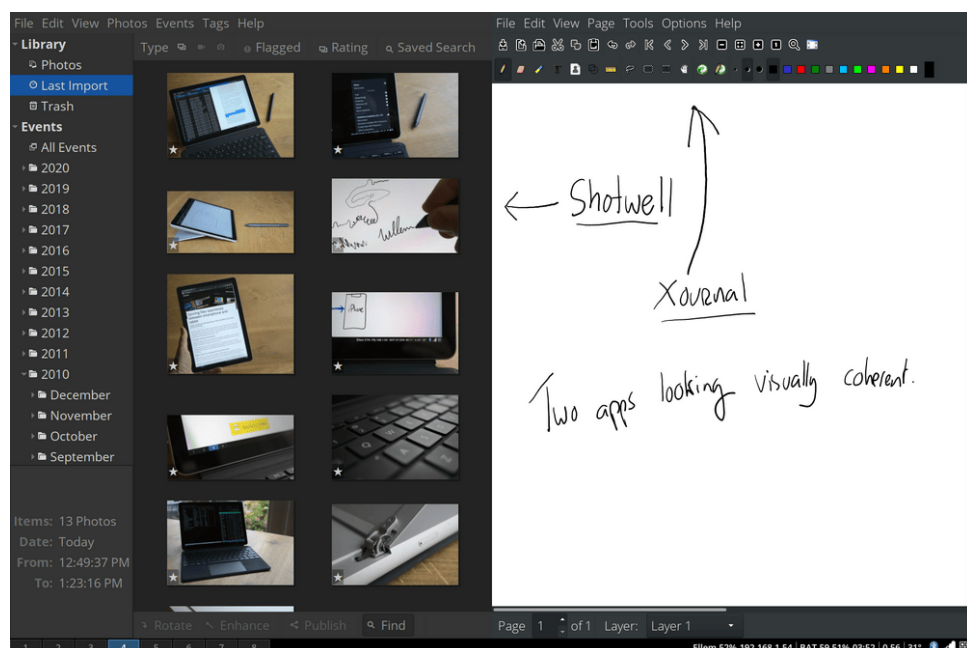
A consistent look and feel is important to enable a frictionless environment and user experience. Therefore I made many small adjustments to colour, fonts, icons and toolbars

to make my apps all look and feel similar. Although it is not perfectly harmonious, it no longer feels as if I am working with “Frankenstein OS”.

I prefer a dark theme as it is easier on my eyes, but you should decide for yourself. Use tools like **lxappearance** to make adjustments for multiple apps. But, you’ll probably need to make adjustment on individual app basis, too. I use the **adwaita-dark** theme and (monochromatic) **HighContrast** icons, with the **Open Sans Light** font.



Thunar file explorer (left) and GIMP (right) in visual harmony - create a uniform look by matching colours, icons and fonts



Shotwell (left) and Xournal (right) - two very different apps running side by side, looking visually coherent



The kickstand is very sturdy as it uses two strong metal hinges



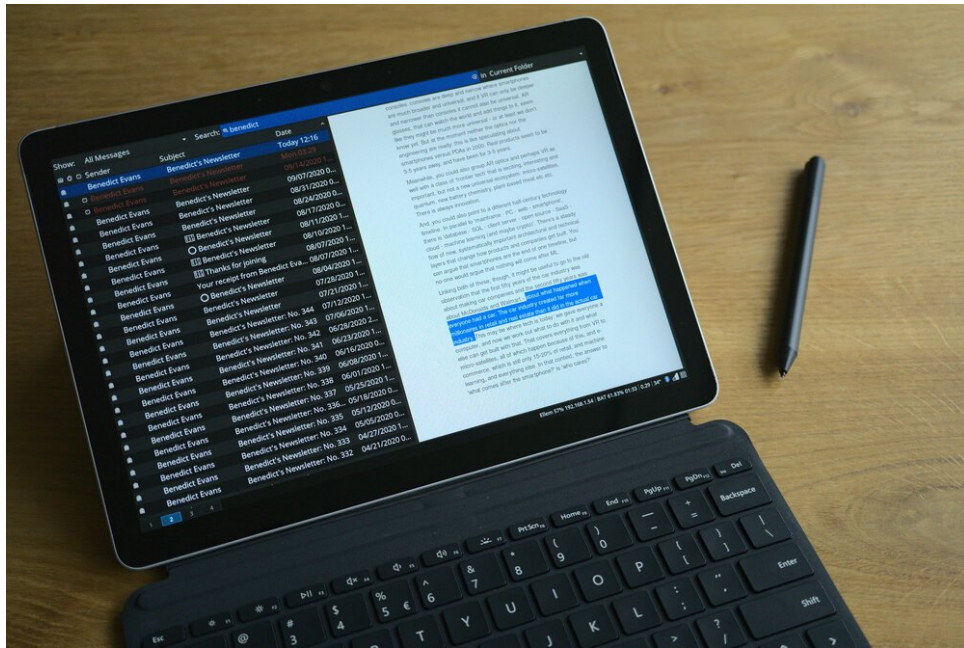
The type cover is charm to type on with real (backlit) keys, including ESC and a row of function keys



The back of the Surface has no extruding camera lens which allows the tablet to lay perfectly flat on a table - as it should!



It's a tiny computer, but it is incredibly powerful and versatile!



Reading mails in Evolution works well!



It is light enough to handle with one hand - you can read the gyroscope sensor to automatically rotate the display - useful when reading an (interesting) article



The Surface Go 2 is a serious tool, capable of powerful things when you connect to the cloud (using a terminal, MOSH and tmux)

Conclusion

Building my own tablet OS is a real challenge. My first attempt was not good enough to replace my iPad. This second attempt however is different. The time and effort I put into refining the user experience results in a much better system. I now use it as my main computer on a daily basis.

Although I always appreciated the work that Apple has done on iPad, it remains limited by its locked down software. Especially now, when Apple has decided to [double down on its AppStore control](#). The only way to control the future is to create it yourself: I believe the time is right to become captain on my own ship!

Update January 2021

[After months of intense use I have decided to abandon my own tablet OS](#)