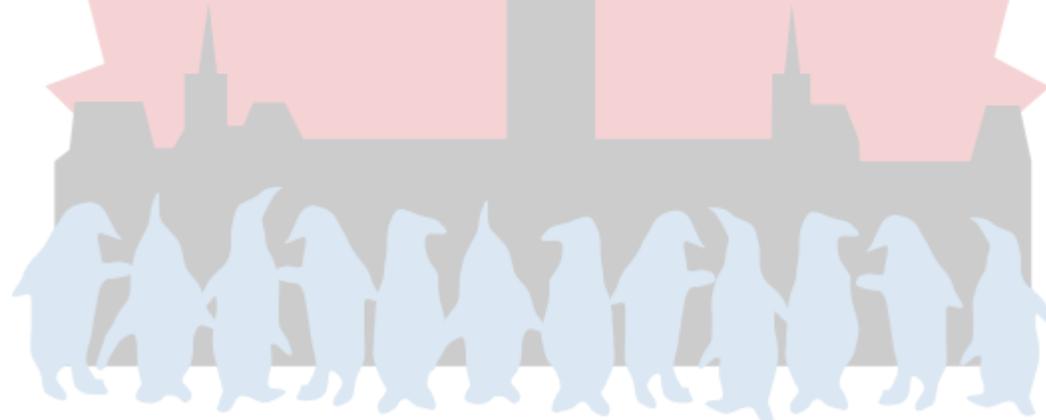


# Beneath the Surface

scott.murphy@arrow-eye.com

2017-05-04

A lot happens beneath the surface



**Ottawa Canada**  
**Linux Users Group**

# About this Talk

This is not going to be a greatly detailed talk. After the previous presentations on the surface units, I acquired a Surface Pro 4 specifically for the purpose of trying something different and this qualifies in many ways. Of course putting Linux on it is the actual goal. I have tried three distributions so far, all Arch based.



**Ottawa Canada**  
Linux Users Group

# Why Arch?

The short story is UEFI boot from the installer and a very up-to-date kernel. I've decided that I'm going to use the latest kernel and if hardware does not work, see if I can find work arounds for the sticking points.

# What hardware did I pick?

The model I picked up is a the quad core i7 with 16GB RAM and a 250GB storage device.

I picked this one because buying the base unit to play with would be annoying if this worked half as well as I hoped. No way to do a real upgrade, so I would be stuck with very minimal hardware and feeling cheated somehow.

There was also a deal where I got the Type Cover (basically keyboard) and a second power adapter. I also got one of the Arc mice. I kind of like it. There is also a family license for Office 365, but I don't have a need for that

# First Impressions

When I powered it up, I had to admit that this is a nice device. Well thought out, buttons where you expect them to be, magnetic clips for everything.

I'm still not sure about the stylus, I'm not much of a touch screen person (at this time).

It looks like they designed a very nice piece of hardware. Time will tell if that is in fact true.

# What does it look like?



Surface Pro 4



**Ottawa Canada**  
Linux Users Group

# Right side view



You can see the groove for popping out the stand, the power/dock connector, the USB 3 port and the mini display port. Hidden away under the stand is a MicroSD slot.



**Ottawa Canada**  
Linux Users Group

# Left side view



You can see the groove for popping out the stand, and the headphone/microphone jack.



**Ottawa Canada**  
Linux Users Group

# OK, so what is the process?

This was amazingly simple. I found a post describing the process which was way more involved than I needed. This person converted the internal storage to ZFS pools and used the MicroSD as the home partition.

This entailed lots of messing around with ZFS commands until the actual install and then pretty much standard.

# Details

# Preliminary Things

- Boot up the device, do all the Microsoft registration stuff, download all of the updates
- You are doing this in case there are firmware updates for the components. Remember that this is a Microsoft device and you will probably only get hardware fixes from Microsoft and that via windows.



**Ottawa Canada**  
Linux Users Group

# Make Room!

- Now using the windows disk tools, shrink the windows partition so you have room to install Linux.
- In my case, I couldn't shrink it by a whole lot, there were unmovable files in the partition. I shrank it as far as I could and then worked with what I had.
- I could have saved myself a little extra work if I had disabled windows swap and the hibernate file. Once those were gone, I could shring it to about half the size.



**Ottawa Canada**  
**Linux Users Group**

# Disable Secure Boot

This is not that hard, boot into the firmware (Power+Volume-Up) and you get into a nice firmware screen. Under Security, disable Secure Boot. Then adjust the boot order to boot from USB first. When you exit, you will reboot.

If you want, you can go through the process of signing your kernel, installing the keys and re-enabling secure boot so you do not have the red banner with the unlocked padlock on it when you power up. I may get to that one of these days, but If I'm updating the kernel image often, then I need to automate the process.

# Boot into Arch

Assuming you put your newly created boot media in the USB port, you are ready to do things.

After a few seconds, you will be greeted with a `#` prompt. You are ready to install Arch. The instructions on the main site will be sufficient for almost all of it. Setting up the UEFI boot will take a little more effort.

# Decisions

Before you optimize for disk space for Linux, remember that you probably want to keep a bootable Windows 10 for the updates. If you do not, then you can remove the recovery partition and other miscellaneous partitions. Keep the first partition, the EFI boot one. You can delete it and rebuild it later if you want.

The idea here are to dump the EFI partition, the Windows partition, and the Windows Recovery partition to an external device. In order to do that, you need to be either running Linux already, or have a powered USB hub so you can have more then one USB device plugged in.

It might even be possible to boot windows from a USB device and that would be even better. Just store it on an external drive – a USB stick is probably a bad idea.

I have not done this yet, but I am considering it.



**Ottawa Canada**  
**Linux Users Group**

# Installation

Assuming you are following the standard install, the only thing that might cause you grief is the partitioning.

This is my partition table right now.

```
# fdisk -l
Disk /dev/nvme0n1: 238.5 GiB, 256060514304 bytes, 500118192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: AB605323-969D-4612-AD01-FF4FF753E3A0

Device            Start      End      Sectors  Size Type
/dev/nvme0n1p1    2048      534527  532480   260M EFI System
/dev/nvme0n1p2    534528    796671  262144   128M Microsoft reserved
/dev/nvme0n1p3    796672    67905535 67108864 32G Microsoft basic data
/dev/nvme0n1p4    498315264 500117503 1802240 880M Windows recovery environment
/dev/nvme0n1p5    67905536 68429823 524288 256M Linux swap
/dev/nvme0n1p6    68429824 498315263 429885440 205G Linux filesystem
```

# Partitoning

Basically, leave the first partition alone, it is the UEFI boot partition. FAT32 based, roughly 512MB in size. Your Linux bootloader will end up in here.

You should add two new partitions, one as swap, the other as the file system. Using your partition tool of choice, create your new partitions. assign 256M (or more) to your swap and the rest to Linux. Make note of which is which. When you are installing, you want to make sure you install to the correct location.



**Ottawa Canada**  
Linux Users Group

# Obtaining PARTUUID Information

You will want this information to create the proper conf files for booting.

```
[root@boomtown ~]# blkid
/dev/nvme0n1: PTUUID="ab605323-969d-4612-ad01-ff4ff753e3a0" PTTYPE="gpt"
/dev/nvme0n1p1: LABEL="SYSTEM" UUID="964C-0D5D" TYPE="vfat" PARTLABEL="EFI system partition" PARTUUID="9707fcea-8f03-4268-8
/dev/nvme0n1p2: PARTLABEL="Microsoft reserved partition" PARTUUID="fa649fe4-55a4-4ee7-a6bc-d857197887ad"
/dev/nvme0n1p3: LABEL="Local Disk" UUID="60B0F056B0F033DE" TYPE="ntfs" PARTLABEL="Basic data partition" PARTUUID="fe70dc1f-
/dev/nvme0n1p4: LABEL="Windows RE tools" UUID="62EEF2BFEEF28A95" TYPE="ntfs" PARTLABEL="Basic data partition" PARTUUID="9bb
/dev/nvme0n1p5: LABEL="swap" UUID="36211f3f-4c6c-40f3-8adf-1be859f14d99" TYPE="swap" PARTLABEL="Linux swap" PARTUUID="87622
/dev/nvme0n1p6: UUID="d41c6d7d-15f3-4b10-ad84-feb6e5312ddf" TYPE="ext4" PARTLABEL="Linux filesystem" PARTUUID="318fa873-ff0
```

It scrolls off the right here, but you are looking for the string after PARTUUID, something like PARTUUID="9707fcea-8f03-4268-89ae-e6430bbe14ca"



**Ottawa Canada**  
Linux Users Group

# UEFI Boot

There is no need to install rEFInd or any of that, the existing system will work. Systemd-boot works well here.

```
bootctl --path=/boot install  
pacman -S intel-ucode
```

Now edit the config file `loader.conf`.

```
# cd /boot/loader  
# ls  
entries loader.conf  
# vim loader.conf
```

make the contents look like this:

```
default arch  
timeout 3
```

# UEFI Boot

Next, edit the arch.conf entry in the entries directory:

```
# cd entries[root@boomtown loader]# cat loader.conf  
# vim arch.conf
```

Make it look like this:

```
title      Arch Linux  
linux     /vmlinuz-linux  
initrd    /intel-ucode.img  
initrd    /initramfs-linux.img  
options  root=PARTUUID=318fa873-ff0a-43e4-920c-75d789054498 rw
```

The PARTUUID info you get from the `blkid` command you ran earlier.

# Boot Menu

When I was done, the boot menu had four entries:

- Arch Linux
- Windows Boot Manager
- EFI Default Loader
- Boot into Firmware Interface

They all worked, so no real difficulties.



**Ottawa Canada**  
Linux Users Group

# I have a login prompt...

Yeah, you do. It is also rather tiny, as you can see.



This is a 2736 x 1824 capture of the login screen.

# Zooming in

```
Arch Linux 4.10.13-1-ARCH (tty1)
boomtown login: _
```

This is the top left corner of the previous image

# Getting a Desktop

Unless you are a CLI junkie and like using a split screen text terminal which you can do if you want to, you will probably want a GUI desktop. We can talk about obscure text terminals later.

The next section is not supposed to start an argument regarding which desktop or greeter is the best, I just picked a couple to play with as low water mark items.

# Desktops and Greeters

As I am just testing at this point, I have made no actual decisions yet.

Items I tried:

- Deepin Desktop
- MATE Desktop
- Budgie Desktop



**Ottawa Canada**  
Linux Users Group

# Installing Deepin

I decided to start with Deepin. The process is rather simple:

```
pacman -Syu  
reboot  
pacman -S xorg xorg-server  
pacman -S deepin  
pacman -S deepin-extra
```

# Enable the Greeter

```
vi /etc/lightdm/lightdm.conf
```

Look for the commented out greeter session and change it to

```
greeter-session=lightdm-deepin-greeter
```

Then enable the service

```
systemctl enable lightdm.service
```

You can reboot now and see your glorious desktop.



**Ottawa Canada**  
Linux Users Group

# What does it look like?



Needs a little personalization



**Ottawa Canada**  
Linux Users Group



# So what doesn't work?

- Touch Screen
- Cameras
- The type cover has issues if you remove it and clip it back on
- No stylus for obvious reasons

## Not Tested

- I'm not using the mouse, as the trackpad is fine for me. I have not tested bluetooth at this time.
- I have not tested hibernate. Given I did not create a large swap file
- I have not tried a microphone and headset yet.
- Booting from MicroSD

# Future items

- Replace the 256GB SSD with a larger one
- Get the cameras to work
- Get the touch-screen to work. It might already, but I have not messed with the kernel internals at this point. It may be there and just need to be enabled. It works on a surface 3
- Test a powered USB hub and different devices
- Test the external display port
- Use it as a laptop replacement for short trips
- See how well it handles an external HDD (spinning rust and SSD)



**Ottawa Canada**  
**Linux Users Group**

# Summary

All said and done, this is usable in its current form. Pretty much everything I'd want to use works with little adjustment.

I can only assume that the PixelSense Touch drivers for the Surface Pro 4 will be included in a near future update, at which time the touch screen will be active. That might be interesting and would deserve a followup.

# References

- [Arch Linux on The Surface Pro 4](#)
- [Building a Custom Linux Environment With Openbox](#)
- [Installing Ubuntu on Surface pro 4](#)
- [Install Deepin Desktop Environment in Arch Linux](#)
- [Install MATE Desktop Environment in Arch Linux](#)
- [Install Budgie Desktop Environment in Arch Linux](#)
- [Replacing the SSD](#)