

# Fun With The 'AWK' Command

Today, I'll show you a real-world use of the venerable AWK command. I use the AWK command fairly regularly.

First, a little about AWK. The AWK command has been around since 1977 and is named after the three people who wrote the original version: Alfred Aho, Peter Weinberger, and Brian Kernighan. It was first officially released in Version 7 Unix (V7) and has been a valuable text processing program ever since.

This is just one instance of how I used it to process some text.

I was going through the statistics at the Music For Us site. Among the statistics is a list of countries from which MFU has had visitors. Given the age of the site, it's a fairly long list.

The format of the text was as follows:

```
[code]1 United States United States 8,645
2 Germany Germany 1,044
3 China China 846
4 France France 672
5 Canada Canada 538[/code]
```

I only cared about the name of the countries, but it was a very long list (more than 100 entries) and I didn't want to edit them out manually. So, it was a pretty easy decision to use AWK to process it.

First, I copied the text into a document called 'countries.txt'. I then opened my terminal and headed to that directory. This was the command I used:

```
[code]awk '{ print $2 }' countries.txt > finished.txt[/code]
```

So, I basically told it to use AWK to process countries.txt, to extract the second column, and to write that data to a file called finished.txt. In this case, I could have easily extracted any of the columns without much difficulty. You simply need to change the \$2 to whichever column you want.

If you'd like a full explanation of the command, you can actually check it out on one of my favorite sites, [explainshell.com](http://explainshell.com). (That link will take you directly to the explanation of this specific command.)

Anyhow, the output was as expected. It looks like:

```
[code]United
Germany
China
France
Canada[/code]
```

(If you want to see the full list, check out the paste [here](#). By the way, you can feel free to use that site for your own paste needs. We have our own free pastebin service.)

I guess the final verdict is that Linux really lets me be lazy! Instead of processing that text manually, I used the AWK command and processed it all in less than a second. I didn't have to sit there and copy/paste. I didn't have to spend the time verifying it. I just had to run one simple command and I had the data I needed, nice and neat.

The thing is, I really only need the basics so I have giant holes in my knowledge – and yet, I know enough to do things like that, making my life easier. There's always so much more to learn, and that's awesome.

---

# How To: Uninstall the Default Music Player in Elementary OS

The default music player in Elementary OS is kinda lame. There are many better choices. Once you've chosen one, this is how you remove the default music player in eOS.

This was asked on a forum that I visit and I took the time to find the answer. I figured that I'd not been able to easily find the answer with a search engine, so I might as well turn it into an article. It's actually pretty simple.

eOS lacks any handy GUI system monitor that I could find, so I installed one. You might as well do the same. With eOS, you're eventually going to want it – though top or htop or even atop do the job just as well. Anyhow, I installed one and found that closing the “io.elementary.music” process closed the music player. Obviously, you can't uninstall that. It's part of some bigger package and eOS does things in unusual ways.

With further digging, I finally noticed an application called 'noise' in the list of running processes. Killing it would kill the music player, just like killing the io.elementary.music process would. Finding this out pleased me.

```
[code]sudo apt remove noise[/code]
```

It was actually pretty neat when I ran that command. It uninstalled the default music player but the dock still had the music player icon showing – and the command made that icon fade into a partially transparent icon. Slick move, eOS. Slick move...

At that point, I just decided I'd reboot. Y'all know how to do that already, but let's do it from the terminal.

```
[code]sudo reboot[/code]
```

Or, if you want, you can put them all together with one grand command that looks like this:

```
[code]sudo apt remove noise -y && sudo reboot[/code]
```

Press the enter button, type in your password, and go grab a snack – but just a quick one as it won't take long before you're back at the login screen without the default music player installed any longer.

---

## Realtek RTL8192EU and Linux: It works!

Probably because I was scouring AskUbuntu and reading a number of complaints about getting their wireless device working, I decided to see exactly how difficult it was.

I have a computer that I use for testing and I mostly access it by VNC. It has an unused wireless dongle and I figured now was a good time to see if I can make it work.

The first step was to turn it on. Believe it or not, it worked. It had a very, very weak signal even though it's within a few feet of the router. So, I cracked open my terminal and entered:

```
inxi -Fc 0
```

Sure enough, I found this:

```
Device-2: Realtek RTL8192EU 802.11b/g/n WLAN Adapter type: USB
```

So, I turned to a search engine and entered:

## *Realtek RTL8192EU +Linux driver*

I scanned the results and noticed that there was fairly recent (mid-2018) GitHub page here. So, I headed there to read what the author had to say.

Rather than playing around, I made sure I'd followed their directions:

```
sudo apt -y install linux-headers-generic build-essential dkms git
```

Except that's not necessarily going to work, so I simply removed the -y.

```
sudo apt install linux-headers-generic build-essential dkms git
```

There. That'll work. I mashed the enter button, entered my password, and I downloaded the .zip file from the top of the page while it installed. I didn't really see any need to build it myself, nor did I want the git hurdles (even though I'd just installed it). So, I just downloaded it to /Downloads and then extracted it into its own folder.

Once that was all done, I entered the newly created directory and ran:

```
sudo ./install_wifi.sh
```

The sudo elevates my permission, the ./ tells it that I want the file in the directory I'm in, and the install\_wifi.sh is the name of the script that will install the drivers automatically. It will take a few minutes, even on modern hardware, but it's not a difficult task. If anything, it's straightforward and intuitive.

I followed that up with a reboot and, sure enough, I was then able to use the driver and my signal strength was

significantly higher. That's it. That's all it took. Does it seem hard? Probably. Once you've done it a few times, you'll be used to it. Working in the terminal is one of the greatest things you can do with Linux. There's no reason to be afraid of it. Just know what the commands are going to do before you go ahead and do them.

---

## Installing NetSurf Browser on Ubuntu 18.04

The NetSurf Browser looked interesting to me. I was using a virtual machine with ArchBang Linux and I'm a sucker for a different browser. I have at least two dozen browsers installed at any one time, even though I regularly only use a few. Anyhow, ArchBang has a default of NetSurf and that piqued my interest.

NetSurf doesn't do anything special. It's just a light browser that claims to have CSS support. When I tried browsing with it, it was quite painful and it displayed sites rather poorly. Still, if you have a very old computer and want to have a browser, this may be the kind of browser you're stuck with. It is technically a browser that's lightweight and seems to have some basic CSS support.

As I said, I'm a sucker for browsers and have a bunch installed. In this case, I downloaded the source code and decided I didn't really feel like building it. So, I did the ol' 'apt-cache search netsurf' and nothing popped up. So, I went online and looked for a .deb that'd make it easy to install. It turns out to be more of a pain in the butt than you might think.

No matter... I've got you covered. Like normal, crack open your terminal with CTRL + ALT + T. Then, enter the following:

```
[code]
cd Downloads
mkdir netsurf
cd netsurf
wget
http://archive.ubuntu.com/ubuntu/pool/universe/n/netsurf/netsu
rf-common_3.6-3.2_all.deb
wget
http://archive.ubuntu.com/ubuntu/pool/universe/n/netsurf/netsu
rf-gtk_3.6-3.2_amd64.deb
wget
http://archive.ubuntu.com/ubuntu/pool/universe/n/netsurf/netsu
rf_3.6-3.2_all.deb
sudo apt install ./netsurf-common_3.6-3.2_all.deb
sudo apt install ./netsurf-gtk_3.6-3.2_amd64.deb
sudo apt install ./netsurf_3.6-3.2_all.deb
[/code]
```

Then, assuming all goes well, you'll have a working NetSurf in your menu, under Internet if you use an Ubuntu flavor that uses menus like that.

As 18.04 is the last official 32 bit Ubuntu, and official flavors, odds are good that many people will be looking for a lightweight browser to go with it. ***This isn't a very good browser – but it's a working browser, and a browser that doesn't take a lot of resources to run.*** Don't expect too much from it and you won't be too let down.

You're probably better off taking the performance hit and running Vivaldi.

---

# How to Sanitize EXIF Data From Your Pictures

## Linux Exif Data and Personal Security

When you take pictures, be it with your cell phone or with your digital camera, the software adds data to the pictures. This data is called the Exif data. If your camera supports it, and if you have it enabled, it will encode all sorts of private data along with your image data.

Needless to say, this is a potential privacy nightmare and people have not only been doxxed this way but people have ended up in jail because of leaving the data in their photos. Now, frankly, if you're taking images that'd get you tossed into jail, I'm pretty much okay with that. For the rest of you, read on...

First, install *exiftool* from your default repositories. If you're using apt, it's quite simple and almost certainly in your default repositories.

```
[code]sudo apt install exiftool[/code]
```

Now, let's make it work. Open your *.bash\_aliases* file, assuming you have one, and add the following line:

```
[code]alias picclean="exiftool -overwrite_original -all= *.png && exiftool -overwrite_original -all= *.jpg && echo $PWD 'images cleaned for privacy sake!'"[/code]
```

Now, reload your aliases with:

```
[code]source ~/.bash_aliases[/code]
```



To use this, open the directory that houses the photos you want to share with your terminal and just enter:

```
[code]picclean[/code]
```

That will clean all your .png and .jpg files. That's also enough information for you to customize it for your system, should you have a different configuration than I.