

Programming: Google's Fuchsia, Go (Golang) and Python

By *Roy Schestowitz*

Created *01/06/2019 - 1:03pm*

Submitted by Roy Schestowitz on Saturday 1st of June 2019 01:03:02 PM Filed under [Development](#) [1] [Google](#) [2]

- [Fuchsia Friday: Android, Linux apps, and Fuchsia's close relationship w/ Chrome OS](#)[3]

Following along with the development of Google's Fuchsia OS, it has become clear that it will be capable of running both Linux and Android apps. Chrome OS can also do both of these things, and that's no coincidence, as the Fuchsia team has opted to use some of Chrome OS's developments for their own benefit.

- [Golang Gets Cheaper Context Switching](#) [4]

As good news considering how much longer it takes to perform a full context switch on Intel CPUs due to various vulnerability mitigations, the Go programming language run-time now has the ability for performing cheaper context switches.

Landing in GCC 10 Git with the Golang code is a less involved context switching implementation for Linux x86_64 systems with the libgo run-time library.

- [Build an XML sitemap of XML sitemaps](#) [5]

Suppose that you have so many thousands of pages that you can't just create a single /sitemap.xml file that has all the URLs (aka) listed. Then you need to make a /sitemaps.xml that points to the other sitemap files. And if you're in the thousands, you'll need to gzip these files.

The blog post demonstrates how Song Search generates a sitemap file that points to 63 sitemap- $\{M\}$ - $\{N\}$.xml.gz files which spans about 1,000,000 URLs. The context here is Python and the getting of the data is from Django. Python is pretty key here but if you have something other than Django, you can squint and mentally replace that with your own data mapper.

- [Generate a random IP address in Python](#) [6]

I have a commenting system where people can type in a comment and optionally their name and email if they like.

In production, where things are real, the IP address that can be collected are all interestingly different. But when testing this manually on my laptop, since the server is running `http://localhost:8000`, the `request.META.get('REMOTE_ADDR')` always becomes `127.0.0.1`. Boring! So I fake it.

[Development Google](#)

Source URL: <http://www.tuxmachines.org/node/124439>

Links:

[1] <http://www.tuxmachines.org/taxonomy/term/145>

[2] <http://www.tuxmachines.org/taxonomy/term/120>

[3] <https://9to5google.com/2019/05/31/fuchsia-friday-android-linux-apps-chrome-os/>

[4] https://www.phoronix.com/scan.php?page=news_item&px=Golang-Cheaper-Context-Switches

[5] <https://www.peterbe.com/plog/build-an-xml-sitemap-of-xml-sitemaps>

[6] <https://www.peterbe.com/plog/generate-a-random-ip-address-in-python>