

Software: Book Squire, isolcpus, GNU Health and rpminspect

By *Roy Schestowitz*

Created *09/10/2019 - 1:34am*

Submitted by Roy Schestowitz on Wednesday 9th of October 2019 01:34:31 AM Filed under [Software](#) [1]

- [Book Squire Is Ten Years Old](#) [2]

I choose Python for the first version. I got the logging in, navigating of the Library site and the scraping of account data working as a script. Then decided to built it into an application running under the then new Google App Engine platform. That worked for a while just fine. Over time I added a database to store user information and an email notifications feature with nightly reports delivered when accounts had notable events worth mentioning.

After working on a few Django applications I decided to move Book Squire to Django and host it on a VPS. Here it stayed for many years working well except for the random updates made to the Library site which broke the parsing of the pages.

Eventually, the Library upgraded there system in a significant way. Actually made it somewhat user friendly. Still it didn't support multiple cards and you had to click around a bit so Book Squire was reworked and it continued on.

For my latest update to Book Squire I've rewritten it in Clojure. The latest version is much cleaner internally and suspect the maintenance going forward will be easier. The old Python code did suffer overtime as refactoring was never justified enough because it just worked.

- [Matt Fleming: isolcpus is deprecated, kinda](#) [3]

A problem that a lot of sysadmins and developers have is, how do you run a single task on a CPU without it being interrupted? It's a common scenario for real-time and virtualised workloads where any interruption to your task could cause unacceptable latency.

For example, let's say you've got a virtual machine running with 4 vCPUs, and you want to make sure those vCPU tasks don't get preempted by other tasks since that would introduce delays into your audio transcoding app.

Running each of those vCPU tasks on its own host CPU seems like the way to go. All you need to do is choose 4 host CPUs and make sure no other tasks run on them.

How do you do that?

- [GNU Health HMIS 3.6 Release Candidate 1 is out !](#) [4]

We are pleased to announce the initial release candidate for the upcoming GNU Health HMIS server !

- [rpminspect-0.7 released, bug fixes and a new integration test suite](#) [5]

rpminspect-0.7 has been released. The main things in this release are a new integration test suite and many bug fixes. There is one new user feature and that's the `-t` or `--threshold` option.

The `-t` option lets you control the result code that triggers a non-zero exit code from rpminspect. By default, this is set to `VERIFY`. But you could set it to `BAD` or `INFO` or any other valid result code in the program. The result code specified by this option means that any result in rpminspect at that code or higher will trigger a non-zero return code. Combined with the `-T` option, this can be a useful tool for some types of CI system integration.

[Software](#)

Source URL: <http://www.tuxmachines.org/node/129055>

Links:

- [1] <http://www.tuxmachines.org/taxonomy/term/38>
- [2] <http://blog.bradlucas.com/posts/2019-10-07-book-squire-is-ten-years-old/>
- [3] <http://www.codeblueprint.co.uk/2019/10/08/isolcpus-is-deprecated-kind.html>
- [4] http://savannah.gnu.org/forum/forum.php?forum_id=9568
- [5] <http://blog.burdell.org/2019/10/rpminspect-07-released-bug-fixes-and.html>