

# Re: How to develop a random number generation device

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-09/msg02336.html>

---

- *From:* Nobody <nobody@xxxxxxxxxxxx>
  - *Date:* Thu, 13 Sep 2007 01:07:04 +0100
- 

On Wed, 12 Sep 2007 23:46:24 +0200, David Brown wrote:

If you try to run a 5-year old Linux binary on a current distribution, you'll probably find that a lot of the interfaces on which it depends have either disappeared or have changed in an incompatible manner. Lack of a stable ABI is a simple fact of life on Linux.

Binary compatibility for the user-level API is extremely important for Linux, and the API is very stable – new API's and system calls are added, but existing ones are seldom changed or removed, and never without very good reason.

That's only really true for "core" libraries, e.g. libc. Once you get up to higher levels (e.g. X and GUI toolkits), the ABIs can change quite frequently.

Anything which uses C++ tends to be especially fragile, partly due to extensive use of structures (objects), partly due to the amount of inlined code (which often relies upon structure layout and won't change along with non-inlined functions in the library).

Most five year old Linux binaries will run fine on the same architecture as they were build for (assuming you have the right libraries)

IOW, a five year old binary will run on a five year old OS distribution. If you want to run it on a newer distribution, you need to start chasing down all of the five-years-ago versions of the libraries that it uses.

– the only API calls that have changed are very seldom used.

## Re: How to develop a random number generation device

It isn't so much the syntax of the calls (even a fairly slack programmer can figure out that changing the argument list will change the API), but the structures which they use (and, less frequently, their semantics).

.