Android's C library: bionic

Elliott Hughes <enh@google.com>

Why?

Why not glibc?

Why not a BSD libc?

Why not musl?

Why not Zoidberg?

What?

Large parts of (Free|Net|Open)BSD libc +
script-generated system call stubs +
home grown Linux-specific code if needed.

Supports arm, arm64, mips, mips64, x86, x86-64.

What's unusual?

No separate libpthread/libresolv.

No __isthreaded.

No support for old kernels.

No legacy syscalls used (on any architecture).

Non-_r functions usually thread safe.

Lots of unit tests!

All pic/pie.

What's in the headers?

Some BSD.

Some home-grown based on POSIX specs.

All the uapi headers under <linux/…>.

<features.h> mostly broken.

Spotty C99/C11 support.

No particular version of POSIX.

Missing Linux kernel APIs we'd use

- set_tid_address(2)/clone(2) tid maintenance flags work great. No paired unmap+exit.
- Getting randomness used to be awkward. Tso's getrandom(2) looks good.
- How to atomically read /proc/<pid>/maps?
- Syscall to query a single mmap region's info?
- MINHERIT?