Introduction (1A)

Young Won Lim
11/28/16

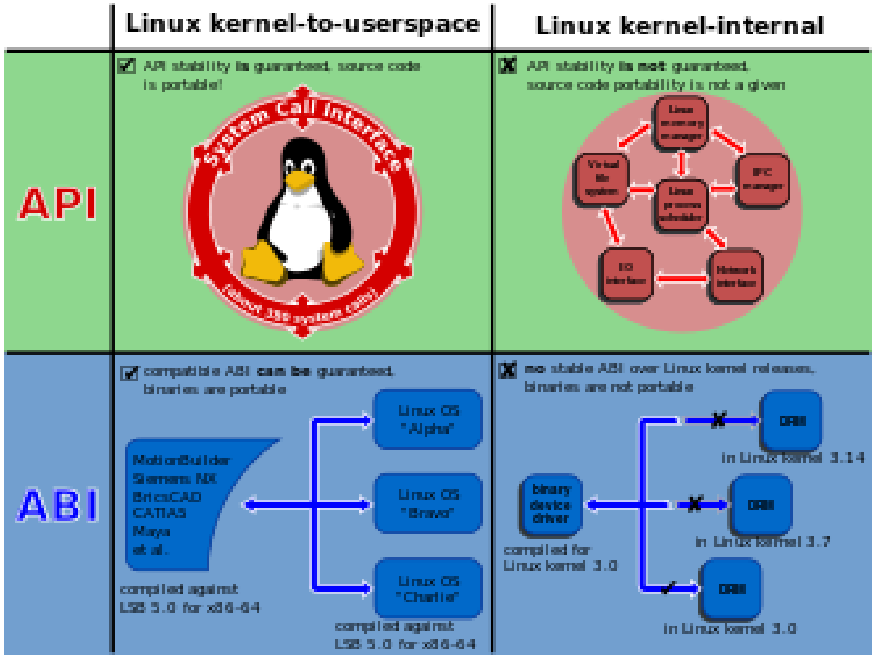
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Linux Kernel Interfaces

The Linux kernel provides several interfaces to user-space applications
that are used for different purposes and
that have different properties by design.

There are two types of
application programming interface (API)
in the Linux kernel that are not to be confused:
the "kernel–user space" API and
the "kernel internal" API.


Linux kernel-to-userspace
Linux kernel-internal
API
☑ API stability is guaranteed, source code is portable!
System Call Interface
(about 380 system calls)
☒ API stability is not guaranteed, source code portability is not a given
Linux memory manager
Virtual file system
Linux process scheduler
IPC manager
IO interface
Network interface
ABI
☑ compatible ABI can be guaranteed, binaries are portable
MotionBuilder
Siemens NX
BricsCAD
CATIA5
Maya
et al.
compiled against LSB 5.0 for x86-64
Linux OS "Alpha"
Linux OS "Bravo"
Linux OS "Charlie"
compiled against LSB 5.0 for x86-64
☒ no stable ABI over Linux kernel releases, binaries are not portable
binary device driver
compiled for Linux kernel 3.0
DRM
in Linux kernel 3.14
DRM
in Linux kernel 3.7
DRM
in Linux kernel 3.0

https://en.wikipedia.org/wiki/Linux_kernel_interfaces#Linux_API

Linux API

The Linux API is

- the kernel–user space API
- allows programs in user space
  - to access system resources
  - and services of the Linux kernel.

composed out of

- the System Call Interface of the Linux kernel
- and the subroutines in the GNU C Library (glibc).

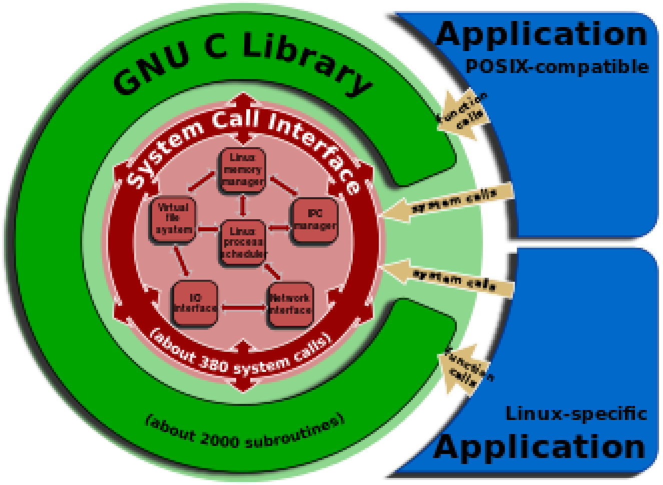
https://en.wikipedia.org/wiki/Linux_kernel_interfaces#Linux_API

Linux API

System Call Interface is the denomination for the entirety of all implemented and available system calls in a kernel

The GNU C Library is a wrapper around the system calls of the Linux kernel;

the combination of the Linux kernel System Call Interface and glibc is what builds the Linux API.



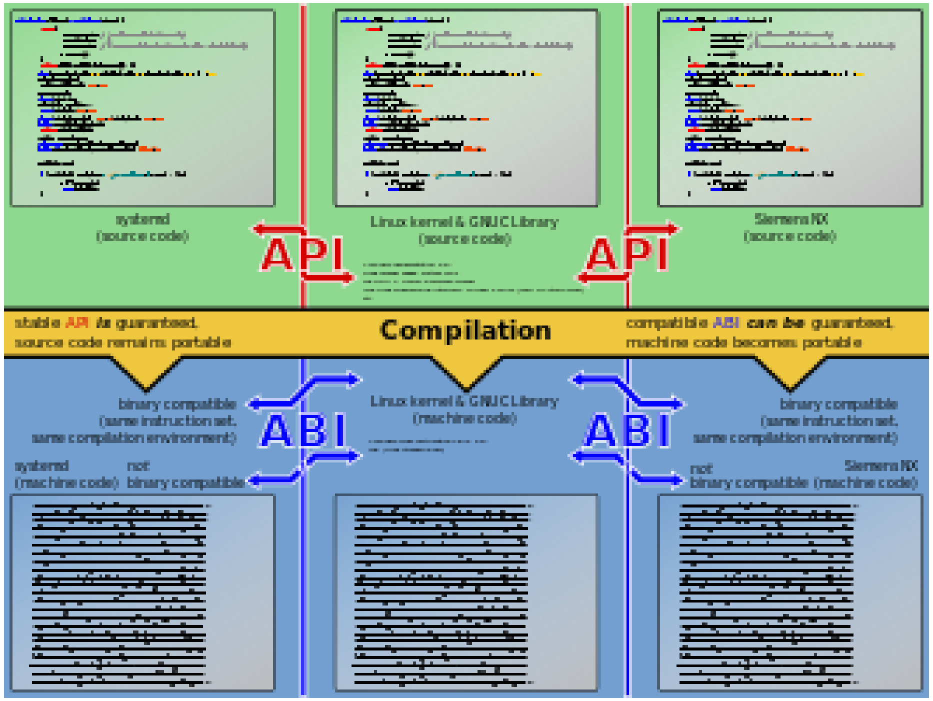
https://en.wikipedia.org/wiki/Linux_kernel_interfaces#Linux_API

Linux ABI

The term Linux ABI refers to a kernel–user space ABI.

The Application binary interface refers to the compiled binaries, in machine code.

Any such ABI is therefore bound to the instruction set.



https://en.wikipedia.org/wiki/Linux_kernel_interfaces#Linux_API

System Call

In computing, a system call is the programmatic way
in which a computer program requests a service
from the kernel of the operating system it is executed on.

This may include
hardware-related services
(for example, accessing a hard disk drive),
creation and execution of new processes,
and communication with integral kernel services
such as process scheduling.

System calls provide an essential interface
between a process and the operating system.

https://en.wikipedia.org/wiki/System_call

System Call Implementation

Implementing system calls requires
a control transfer from user space to kernel space,
which involves some sort of architecture-specific feature.

A typical way to implement this is
to use a software interrupt or trap.

Interrupts transfer control to the operating system kernel so software simply needs to set up some register with the system call number needed, and execute the software interrupt.

https://en.wikipedia.org/wiki/System_call

References

[1] http://minix1.woodhull.com/current/2.0.4/
[2]