

kdbus

IPC for the modern world

Greg Kroah-Hartman
gregkh@linuxfoundation.org

github.com/gregkh/presentation-kdbus

THE **LINUX** PROGRAMMING INTERFACE

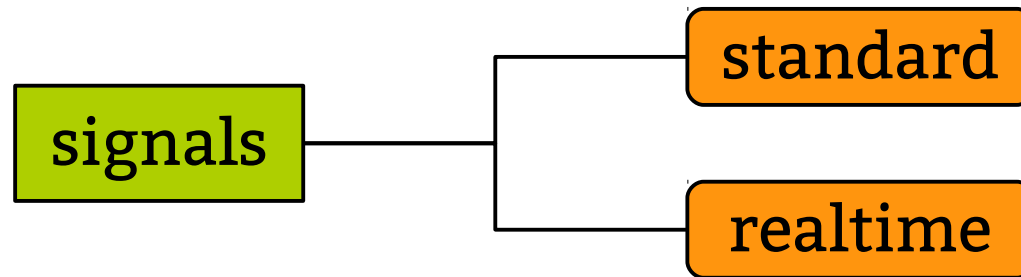
A Linux and UNIX® System Programming Handbook

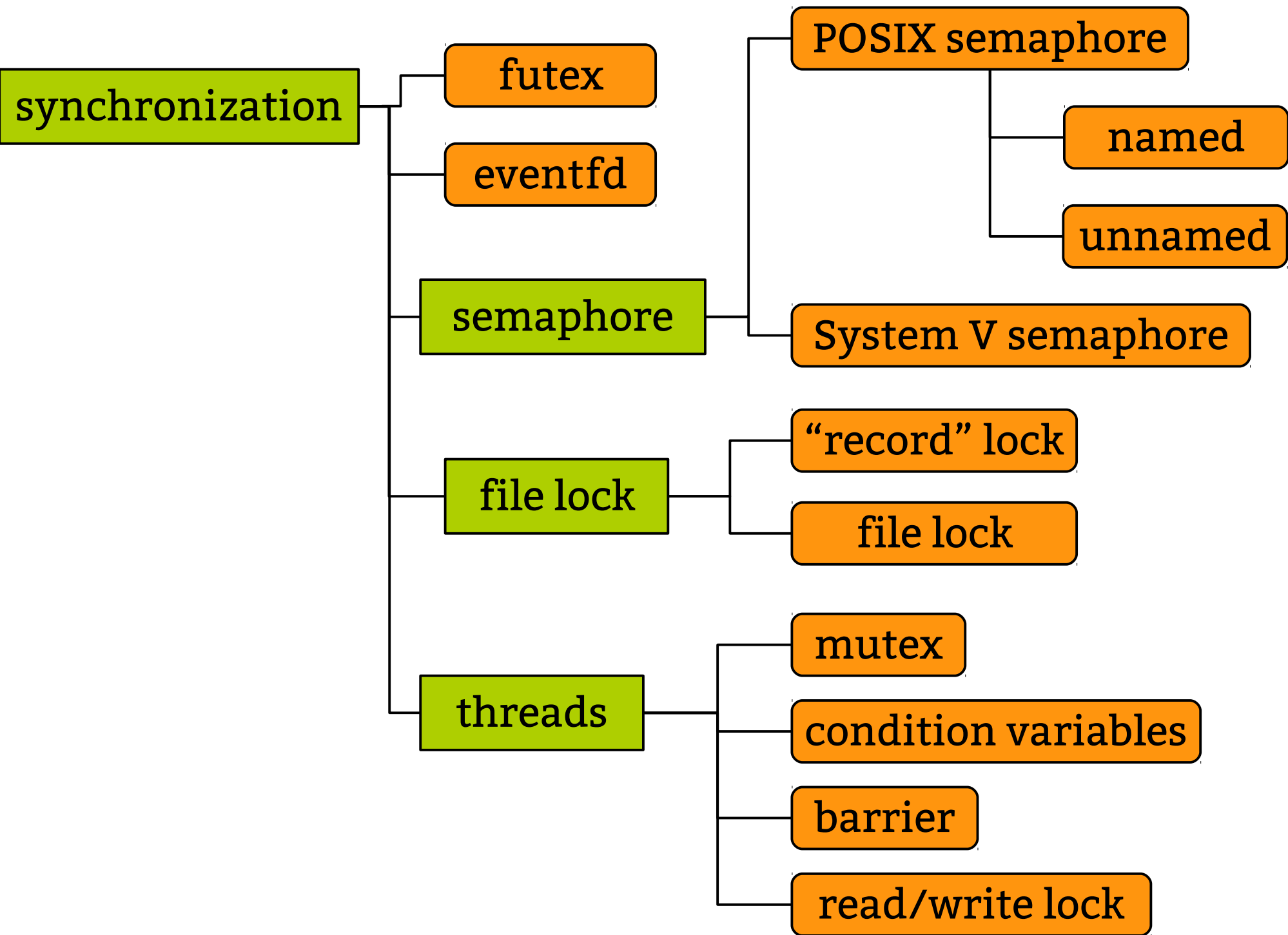
MICHAEL KERRISK

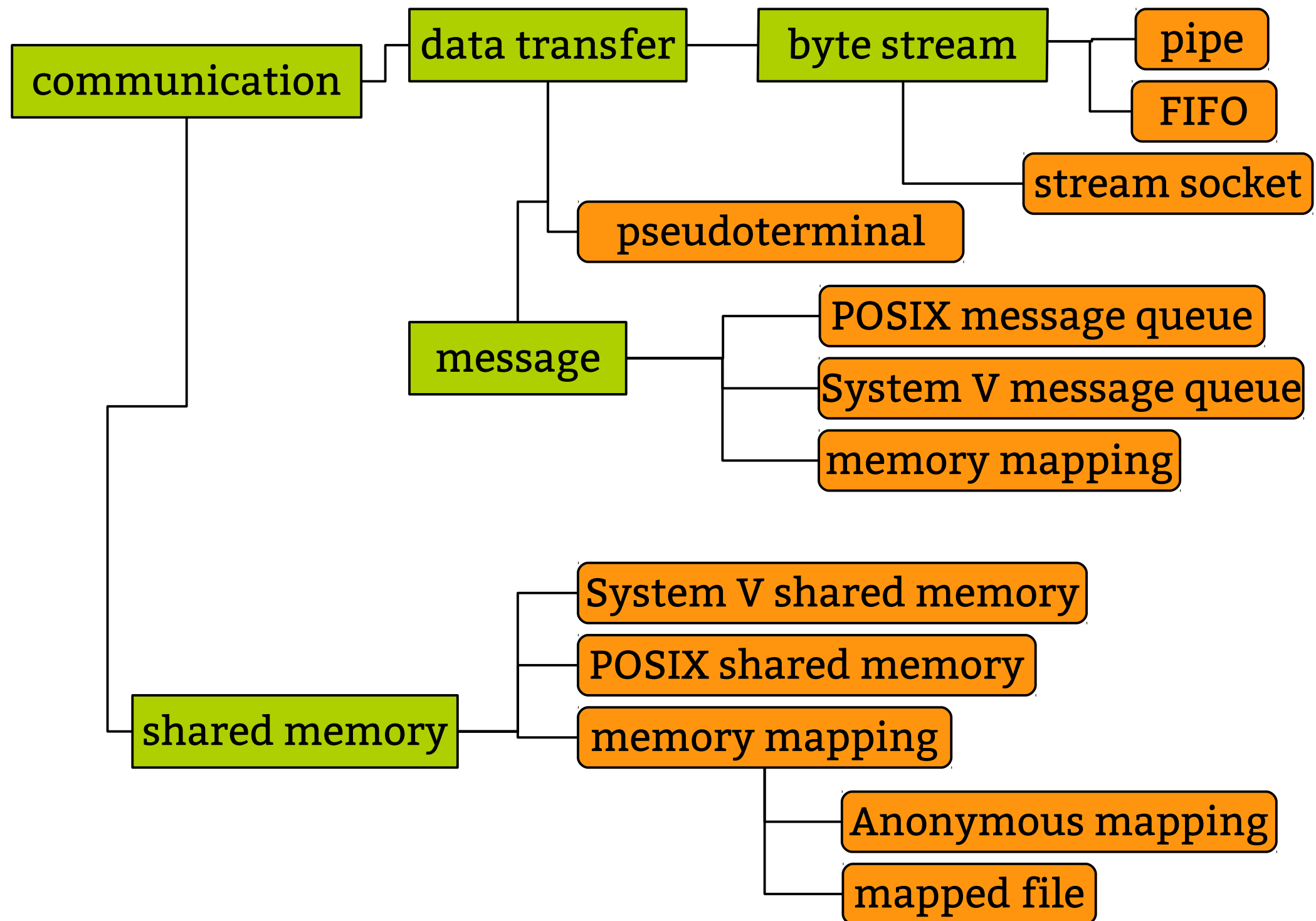


Interprocess Communication

- signal
- synchronization
- communication







Android

- ashmem
- pmem
- binder

ashmem

- POSIX shared memory for the lazy
- Uses virtual memory
- Can discard segments under pressure
- Unknown future

pmem

- shares memory between kernel and user
- uses physically contiguous memory
- GPUs
- Unknown future

binder

- IPC bus for Android system
- Like D-Bus, but “different”
- Came from system without SysV types
- Works on object / message level
- Needs large userspace library
- NEVER use outside an Android system

binder

- File descriptor passing
- Used for Intents and application separation
- Good for small messages
- Not for streams of data
- NEVER use outside an Android system

QNX message passing

- Tight coupling to microkernel
- Send message and control, to another process
- Used to build complex messages and objects
- Linux implementation: SIMPL

kbus

- <http://code.google.com/p/kbus>
 - Tony Ibbs and Ross Younger
- Simple message passing
- No file descriptors
- Multi-copy implementation
- Good for process separation
- Python and C language bindings

D-Bus

- Userspace message bus
- Handles process lifecycles
- Application and System busses
- Subscribe to messages on bus
- Heavily used in Linux systems
- Built on top of existing IPC base types
 - Works on BSDs and other operating systems
- Not optimized for speed

AF_BUS

- IVI people wanted faster D-Bus
- Collabora created, GENIVI sponsored
- Kernel network protocol, removes 2 system calls per message
- Faster than D-Bus daemon
- Rejected by upstream kernel developers

Faster D-Bus

- systemd rewrote libdbus
 - 360% faster than D-Bus

If you want a faster D-Bus,
rewrite the daemon / library,
don't mess with the kernel.

kdbus

- message based, single and multicast
- No extra kernel wakeups
- Filtering in-kernel (bloom filters)
- No blocking calls
- Naming database in kernel
- Reliable, in-order guarantee

kdbus

- Process separation
- 1 copy message passing
- File descriptor passing
- Data streams might be possible

kdbus

- Namespace aware
- Audit aware
- LSM aware (SELinux will work properly)

kdbus

- Will be used for GNOME portals
 - application separation / intents
- Will replace D-Bus in systemd “soon”
 - work in progress
- Fast???
- Will be merged, “when it is ready”
- Hopefully can replace binder, no guarantees

kdbus

<https://github.com/gregkh/kdbus>

- Kay Sievers
- Daniel Mack
- Tejun Heo
- Lennart Poettering
- Greg Kroah-Hartman