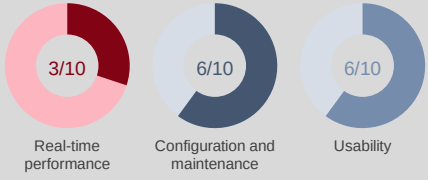
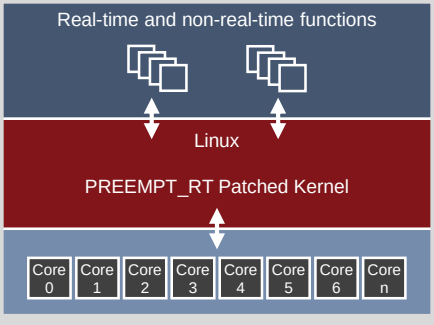
3 Ways to get Linux Ready for Real-Time

The PREEMPT_RT Patch

PREEMPT_RT is the most well-known kernel patch for making the Linux kernel preemptive. Preemption increases determinism and latency.





Real-time characteristics
- Worst case interrupt latency: ~ 10-50 µs.

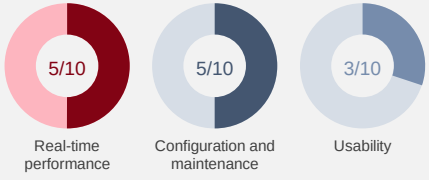
Configuration and maintenance
- Needs patching of the Linux kernel.
- PREEMPT_RT is maintained by the Linux Foundation.

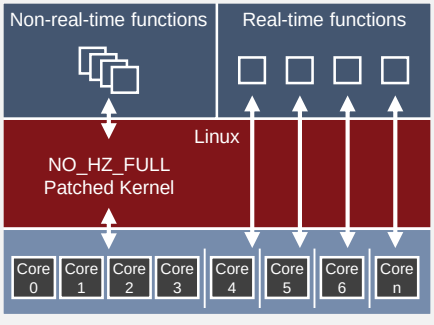
Usability
- Decreases throughput significantly.

Suitable for
- Functions with moderate real-time requirements.

User Space Core Isolation

Pinning single threads to their own cores and isolating them from the Linux kernel allows them to run in a bare-metal like environment, providing very good real-time characteristics.





Real-time characteristics
- Worst case interrupt latency: ~ 5-30 µs.

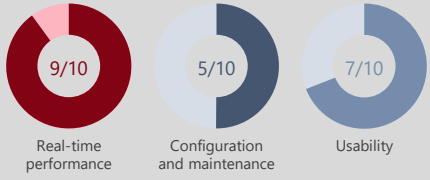
Configuration and maintenance
- Configuration is complex.
- Needs kernel patching.

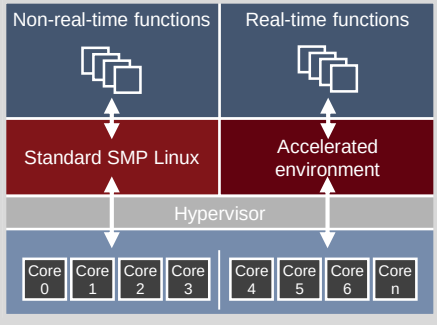
Usability
- Allows only one thread per core.
- System calls from pinned services breaks determinism.
- Provides good real-time characteristics for single-threaded functions pinned to an isolated core.

Suitable for
- Single-threaded real-time polled loop functions.

Hypervisor Acceleration

Partitioning on operating system level using a bare-metal hypervisor provides one real-time domain and one domain for an unmodified Linux kernel with IPC and shared file system.





Real-time characteristics
- Worst case interrupt latency: < 1 µs.

Configuration and maintenance
- Configuration is complex.
- Two different OS SDKs

Usability
- Allows using unmodified SMP Linux.
- System calls and multiple tasks allowed from real-time domain.
- Allows inter-core load balancing for real-time tasks.

Suitable for
- Functions with very high real-time requirements.

ENEA