# P-OS is Innovative

Adaptive Operating System

P-OS follows innovative ideas on Operating System Design.

P-OS implements Control Theory Based Operating System Components e.g Task Schedulers, Memory Management Unites.

In this way, P-OS provides adaptive and fair services using feedbacks in closed loops and uses well known controllers (PID) for fast and smooth system response.
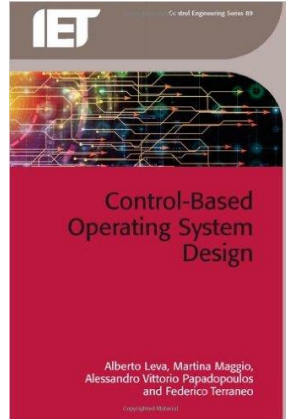
Simulation Support

P-OS supports HW simulations to develop embedded code on your PC like as a Desktop Application. It breaks HW dependency and HW limitations during development. Run your embedded code wherever you want and use unlimited resources of Desktop Application Development.

P-OS is Stable

Stability is must for P-OS and validation phase starts with the projects itself.

P-OS has infrastructures for Continuous Integration and uses CI tools which highly integrated with project environment. Automatized CI infrastructure checks all code modifications from units (Unit Tests, Code Covarage, Static Analysis) to whole project (Project Builds, Integration Tests). Automized Code Reviews are also part of validation.

P-OS is Well-Designed

P-OS just follows general and well-known Computer Science approaches which generally ignored in Embedded Systems. Embedded Systems may have limited resources but it does not mean high level approaches can not be applied.

P-OS is designed Layered and Modular. All relations between modules are definite. You do not see any HW dependent code or static dependecy in kernel.

P-OS allows you to add different types of components or modules by implementing common interfaces like Object Oriented Programming.

P-OS allows you to use Kernel or Drivers seperately. You can use only Drivers in your single thread projects.

P-OS is Highly Optimized

P-OS is highly optimized for limited embedded systems.

Kernel code size is very small so you can use it in your limited CPUs (e.g. Cortex M0). It also supports static resource (e.g. task) creation to save dynamic resource allocation overhead.

P-OS is also highly optimized for Low Power so you can easily use it in power-constraint or battery supplied systems..