Problem Addressed

Most embedded operating systems lack the hardware features of a complete operating system in order to keep the OS size smaller. This results in many shortcomings in embedded platforms. Some microcontrollers provide privilege levels and MMU, but lack virtual memory. Other OS provide simpler implementations of memory isolation and dynamic memory management. Also, they follow monolithic design of OS, forcing the OS and applications to be an atomic unit, making multiprogramming even harder. Also, statically allocating memory brings in limitations in concurrency, reducing the performance of the OS.

Key Idea

The paper mentions the above problems and presents Tock which dynamically allocates memory and provides fault isolation. Tock kernel is totally written in Rust, making it type-safe, memory efficient and having better performance. Tock architecture of two units: capsules and processes. Capsule are written completely in Rust programming language, and each capsule is isolated from the other. Various features of Tock have been implemented inside each capsule. Processes allows concurrent execution of programs as they are hardware isolated. Processes in Tock are similar to those in other OS. Processes can be preempted and the round-robin scheduling technique is used for scheduling processes and context switching.

Strengths

* Unlike most other embedded operating systems, Tock supports all the 5 key features that are required by embedded systems : memory efficiency, dependability, concurrency, updating applications at runtime and fault isolation.
* The threat model of Tock provides better security. There are 4 stakeholders in the model, each identified with a degree of trustworthiness.
* Grants help in isolating a process failure to eliminate resource exhaustion. Grants are responsible for splitting the kernel heap across the processes. If a process uses all of its grant memory, it fails, but it does not affect other processes. Grant also makes sure that resources held by processes are released once the process has died.

Weakness

* Capsule isolation implementation brings in an overhead as each capsule has to be referenced and its dependencies stored. This is in contrast to monolithic architecture which does not have capsules.
* The kernel of Tock is developed on Rust, and as we have seen earlier in SPIN OS, we cannot always rely on a programming language for type safety and security.

Thoughts on the paper

The paper was able to present a new approach for OS in embedded systems, which tries to accomodate the usual features which a traditional OS has, but is removed from embedded OS for minimizing the size of OS. Tock is able to provide dynamic memory allocation and fault isolation. Using Rust programming language for kernel, it is able to provide type safety. It can be said that Tock is a first step in bringing the completeness in embedded OS which was earlier missing.