ABSTRACT

JEREMY P. ERICKSON: Managing Tardiness Bounds and Overload in Soft Real-Time Systems
(Under the direction of James H. Anderson)

In some systems, such as future generations of military unmanned aerial vehicles (UAVs), different software running on the same machine will require different types of timing correctness. For example, flight control software has hard real-time (HRT) requirements—if a job (i.e., invocation of a program) completes late, then safety may be compromised, so jobs must be guaranteed to complete within short deadlines. However, mission control software is likely to have soft real-time (SRT) requirements—if a job completes a fraction of a second late, the result is not likely to be catastrophic, but unbounded lateness would continue to be unacceptable.

The global earliest-deadline-first (G-EDF) scheduler has been demonstrated to be useful for scheduling software with SRT requirements on a multiprocessor, and the multicore mixed-criticality (MC²) framework that uses G-EDF for SRT scheduling has been proposed to safely mix HRT and SRT work on multicore UAV platforms. However, this prior work has limitations that are addressed by this dissertation.

G-EDF is attractive for SRT systems because it allows the system to be fully utilized with reasonable overheads. Furthermore, previous analysis of G-EDF can provide “lateness bounds” on the amount of time between a job’s deadline and its completion time. However, smaller lateness bounds would be preferable, and some tasks (i.e., programs) may be more sensitive to lateness than others. In this dissertation, we explore the broader category of G-EDF-like (GEL) schedulers that have identical implementation and overhead characteristics to G-EDF. We show that by choosing GEL schedulers other than G-EDF, better lateness characteristics can be achieved. Furthermore, we show that certain modifications can further improve lateness bounds while maintaining reasonable overheads. Specifically, successive jobs from the same task can be permitted to run in parallel with each other, or jobs can be split into smaller pieces by the operating system (OS).
Previous analysis of MC² has always used less pessimistic assumptions about execution times when analyzing SRT work than when analyzing HRT work. It is possible that these assumptions can be violated, creating an overload that causes SRT guarantees to be violated. Furthermore, even in the expected case that such violations are transient, the system is not guaranteed to return to its normal operation. In this dissertation, we also provide a mechanism that can be used to provide such recovery.