POSIX-Compatible Application-Defined Scheduling in MaRTE OS

By: Mario Aldea Rivas and Michael González Harbour
Departamento de Electrónica y Computadores. Universidad de Cantabria

Objectives
- Compatible with current POSIX scheduling policies
- Isolate critical parts from failures in the schedulers
- Single-processor or Multi-processor
- Application-defined protocols for mutexes
- Filtering of Events

Model
- User Space
  - Regular Threads
  - App. Scheduled Threads
  - App. Scheduled Threads

Scheduler Space
- Application Scheduler
  - Sched. Events Queues

Scheduler Thread Body

while (1) {
    posix_appsched_execute_actions (&sched_actions, &timeout, &current_time, &sched_event);
    switch (sched_event.event_code) {
        case POSIX_APPSCHED_NEW:
            add_new_thread (sched_event.thread);
            break;
        case POSIX_APPSCHED_READY:
            make_ready (sched_event.thread);
            break;
        case ...
            ...}
}

Advantages compared to application-level implementations
- Extensive information provided by the kernel
- Efficiency

Advantages compared to other implementations
- POSIX compatible
- Integrated Resource management
- Application-Scheduler bugs cannot affect the rest of the system