# Real-Time Systems

## Project No. 9

### Scheduling and Processing of Critical Tasks in Multiprocessor Systems

**Ms. Maleki**

In a modern car, there are different units such as engine control unit (high critical level) and infotainment system (low critical level). These systems must share resources and perform well in other cases while ensuring the safety of critical functions. As a result, the two-level critical-mixed system is a type of critical-mixed systems that has only two critical levels and includes LC and HC periodic tasks, where LC tasks have one worst execution time and HC tasks have two worst execution times. The system starts working in the normal mode of small execution, and if one of the HC tasks rejects its execution time and is not yet completed, the system enters the overflow mode (Overrun).

As the designer of this system, you must map the desired tasks to the cores by an innovative algorithm and schedule the tasks under the Multi-Processor Stack Resource Policy protocol or MPSR and by the EDF-VD scheduling algorithm. This innovative algorithm uses two parameters of resource congestion and kernel efficiency for mapping. Resource congestion means how productive each task is when accessing a specific resource. To map each task, all the resources it has access to are considered and the congestion of these resources is calculated on each core. The priority is to choose the core that has the highest congestion of all resources.Otherwise, the core with the most congestion for one or more resources is selected (tasks must be queued according to the congestion parameters, and if the remaining productivity of the core is less than the productivity of the task being processed, the next core is selected in the queue). Now, if the conditions are the same for two or more cores, the core with the highest productivity is selected (WFD). To evaluate this method, you should generate a set of synthetic tasks using the Uunifast algorithm and compare the following desired parameters using both the heuristic mapping algorithm provided by you and the WFD, and record and report the results.

## Project Phases:

1. In the first phase of the implementation of the task generation algorithm (along with the generated task report), Tol Allocation of resources, allocation of critical parts to tasks, determination of the levels of all tasks (based on SRP protocol) and mapping capability charts as output results be reported

**Diagram: Mapping for 0.25 and 0.5 efficiency for each state.**

- The average mapping capability of 400 tasks with the ratio of the number of HC to LC tasks equal to 1 in 10 times of generating tasks (different tasks in each execution), choosing the number of resources randomly from the interval [2-6] and the number of cores equal to 2/4/8.

**Diagram: Scheduleability in both system modes**

- The average scheduling of 400 tasks with the ratio of the number of HC to LC tasks equal to 1 in 10 execution times (different tasks in each execution time), choosing the number of resources randomly from the interval [2-6] and the number of cores equal to 2/4/8 and productivity 0.25 per core.

## Project Output:

Phase One: Implementation of the algorithm to generate tasks (with generated tasks report), production and resource mapping, assigning critical level to tasks, determining all levels of preemptive tasks based on the protocol (SRP), and the diagrams of mapping ability to be reported as results.

Final Phase: The scheduling must be fully implemented, and all the diagrams must be documented and reported.