

Embedded Systems II, Assignment 5

Mälardalen University

Emil Broberg
School of Innovation, Design and Engineering
Mälardalen University, Västerås, Sweden
Email: ebg20007@student.mdu.se

QUESTION 1

a)

Cache coherency refers to the property of consistently storing data in multiple caches that replicate the same data. In a multi-core or multiprocessor system with a shared cache, each core or processor has a copy of the same data. When one core or processor modifies the data, the other cores or processors need to be notified of the change and update their copies of the data. If all processors have a consistent view of the data at each memory location and at each point in time, then the system is said to be cache coherent.

b)

A shared memory can be beneficial in a multi-core system. It allows different cores to access a common pool of memory, facilitating communication by reading and writing data to shared locations. However, it can also be problematic if the system gets too large. If the system gets too large issues can arise such as larger overhead because of cores waiting for access to the shared memory and communication via shared memory. If no proper synchronization is used, the system might become incoherent and also lead to race conditions.

c)

The two main scheduling approaches are partitioned and global scheduling.

Partitioned: In partitioned scheduling the tasks and processors are partitioned into groups. This means that each processor gets a set of tasks that wait in a queue dedicated to that processor. This makes each processor operate independently which increases simplicity of the system.

- **Strengths:**

- **Simplicity:** Each processor operates independently and scheduling can be done locally.
- **Predictability:** Since each processor operates independently and scheduling is done locally, the system can be designed in a predictable way.
- **Low overhead:** Since each processor independently manages its own tasks, there is minimal overhead associated with coordination and communication between processors.

- **Weak points:**

- **Load imbalance:** If the tasks are not evenly distributed between the processors, some processors might be idle while others are overloaded.

Global: In global scheduling, all processors share a common queue of tasks. A global scheduler makes decisions about task allocation, considering the entire set of tasks and the status of each processor. Tasks can be distributed between processors as needed.

- **Strengths:**

- **Adaptive:** The tasks are assigned dynamically and the system can adapt to changes in the workload. In an open system, a task can be added or removed from the system dynamically and the global scheduler can adapt to this.
- **Less context switching:** Since the tasks are assigned dynamically, there is less context switching between tasks. This reduces the overhead of the system. Tasks can only get preempted once all processors are working on a task.

- **Weak points:**

- **Complexity:** The global scheduler needs to consider the entire set of tasks and the status of each processor when making scheduling decisions. This increases the complexity of the system.
- **RM and EDF not optimal:** RM and EDF might not be optimal in a global scheduling system. Implementing global scheduling can be more complex compared to partitioned scheduling. Coordinating tasks across processors will require a more complex scheduling design.

- **Overhead:** The global scheduler needs to communicate with all precessors to make scheduling decisions. This increases the overhead of the system.

- I. QUESTION 2
- II. QUESTION 3
- III. QUESTION 4
- IV. QUESTION 5
- V. QUESTION 6
- VI. QUESTION 7
- VII. QUESTION 8