**Concurrency and scheduling**

**1.You can explain the advantages and drawbacks of concurrency**

**2.You can list at least four potential problems of concurrency (deadlock, livelock, race condition, priority inversion) and elaborate on them**

Deadlocks: One wait for another

Livelocks: One triggers the other (distributed enless loops)

Race conditions: The output is dependent on the sequence or timing of other uncontrollable events

**3.You can explain the terms online and offline scheduling**

|  |  |
| --- | --- |
| **Online-Scheduling** | **Offline-Scheduling** |
| Inflexible (changes) | Flexible (changes) |
| Always maximum use | Use depends on scheduling algorithm |
| Low costs at execution | Higher cost at execution |
|  | Normally priority based |

**4.You can explain static and dynamic priorities**

|  |  |
| --- | --- |
| **Static Priority** | **Dynamic Priority** |
| EDD (Earliest Due Date First) | EDF (Earliest Deadline First) |
| RMS (Rate Monotonic Scheduling) | LRT (Latest Release Time) |
| DMS (Deadline Monotonic Scheduling) |  |

**5.You can perform schedulability tests (see calculations) and interpret the results**

**6.Determination of task priorities for RMS**

**7.You can name at least four approaches to solve concurrency issues (mutex, semaphore, PIP, PCP)**

**8.You can explain the priority inversion problem**

**9.You can explain how the priority inversion problem can be solved using the priority inheritance protocol (PIP)**

**10.You can explain how the priority inversion problem can be solved using the priority ceiling protocol (PCP)**

**Petri Net**

**1.You can name the elements (places, tokens, transitions, edges) of Petri-nets and explain their interactions**

1. **Places (Stellen)**

Places represent states (Zustände).

1. **Transitions(Transitionen)**

Transitions represent actions or event (Ereignisse)

1. **Edges (Kanten)**

Arcs (flow relations) connect places and transitions where

* + An arc is directed either from place to transition or  
    from transition to place
  + An arc never connects a transition with a transition  
    an arc never connects a place with a place

1. **Tokens (Marken)**

Marks can move from place to place via a transition along the related arcs, but only when the predefined switch condition is fulfilled.

Marks are also called tokens.

* A switch of tokens are an atomic action (uninterruptible).
* A switch takes no time (instantaneous)  
  (instantan = sofort und unmittelbar).

**2.You can explain capacities of places**

The capacity of a place is the maximum amount of tokens that a place can cover.

Default capacity is ∞

**3.You can explain multiplicities of edges**

A Petri net graph is a Petri net structure as a bipartite directed multigraph. Usually, in the graphical representation, parallel arcs connecting a place (transition) to a transition (place) are represented by a single directed arc labeled with its multiplicity, or weight k. A circle contains a dot represents a place contains a token.

**4.You can create a reachability graph for a given Petri-net**

The reachability graph of a Petri net is the part of the transition system reachable from the initial state in graph-like notation.

The reachability graph can be calculated as follows:

1. Let X be the set containing just the initial state and let Y be the empty set.

2. Take an element x of X and add this to Y. Calculate all states reachable for x by firing some enabled transition. Each successor state that is not in Y is added to X.

3. If X is empty stop, otherwise goto 2.

**Folie 33, 34**

**5.You can identify deadlocks.**

* **According to Edward Coffman the conditions are:**

1. Mutual exclusion

2. Hold and wait

3. No preemption

4. Circular wait

**6.You can determine the properties of a given Petri net (boundedness, safe/unsafe and dead/alive**

## liveness mean for Petri Nets?

* The Petri Net is a deadlock-free system.
* The concept of liveness is closely related to the deadlock situation, which has been situated extensively in the context of computer operating systems. A Petri net modeling a deadlock-free system must be live.

## safeness mean for Petri Nets?

* A Petri Net is safe if there are no overflows. Meaning there is no possibility that a token is blocked from transitioning because the next place is full.

## reachability mean for Petri Nets?

* An important issue in designing event-driven systems is whether a system can reach a specific state, or exhibit a particular functional behavior.