

Real-Time Operating Systems

Chapter 10 in "Real-Time Systems"
(Author: Hermann Kopetz)

Mar-08

UCI
DREAM Lab



Outline

- Task Management
- Interprocess Communication
 - NBW (Non-Blocking Writer)
 - NBB (Non-Blocking Buffer)
- Time Management
- Error Detection
- A Case Study: ERCOS

Mar-08

UCI
DREAM Lab



Task Management

- Concerned with the provision of the dynamic environment within a host for the initialization, execution, and termination of application tasks.

Mar-08

UCI
DREAM Lab



Task Management of TT Systems

Time	Action	WCET
10	Start T1	12
17	Send M5	
22	Stop T1	
38	Start T3	20
47	Send M3	

The diagram shows a table with three columns: Time, Action, and WCET. The rows contain the following data: (10, Start T1, 12), (17, Send M5,), (22, Stop T1,), (38, Start T3, 20), and (47, Send M3,). An arrow points from the table to a box labeled 'Dispatcher' which contains a clock icon.

Task descriptor list in a TT operating system

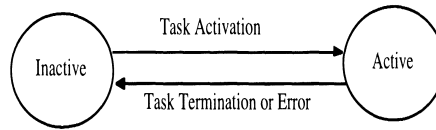
- The temporal control structure of all tasks is established a priori by off-line support tools and encoded in a **Task-Descriptor List (TADL)** that also contains the cyclic schedule for all activities of the node.
- The dispatcher is activated by the synchronized clock tick.
 - It looks at the TADL, and then performs the action that has been planned for this instant.

Mar-08

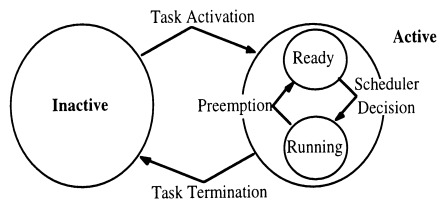
UCI
DREAM Lab



TT Systems with S-Tasks (cont)



State diagram of a non-preemptive S-Task



State diagram of a preemptive S-Task in ET Systems

Mar-08

UCI
DREAM Lab



Task Management of TT Systems

- Too simplistic
- The definition of TT systems is too narrow.

Mar-08

UCI
DREAM Lab



Task Management of ET Systems

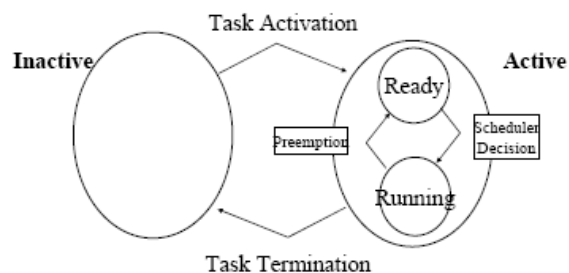
- The sequence of task executions is determined dynamically by the evolving application scenario.
- Whenever a significant event happens, a task is released to the active (ready) state, and the dynamic scheduler is invoked.
- It is up to the scheduler to decide at run-time which one of the ready tasks is selected for the next service by the CPU.

Mar-08

UCI
DREAM Lab



ET Systems with S-Tasks



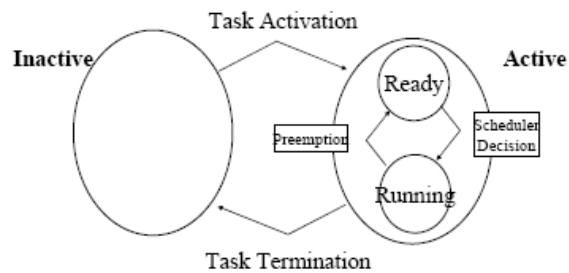
- Significant events causing activation of a task
 - External: Arrival of a msg or an interrupt from the controlled object
 - Internal: Termination of a task or some other condition within a currently executing task
 - Reaching of the clock at a specified point in time

Mar-08

UCI
DREAM Lab



ET Systems with S-Tasks



- **Non-preemptive S-tasks**
 - Can cause poor responsiveness
- **Preemptive S-tasks**
 - Data conflicts between active S-tasks can be avoided if the OS copies all input data required by this task from the global data area and the CNI into a private data area of the task at the time of task activation.

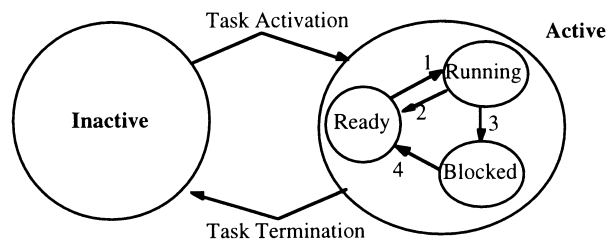
Mar-08

UCI
DREAM Lab



ET Systems with C-Tasks

- **Blocked state** is introduced
- WCET of a C-task cannot be determined independently of the other tasks.



1 Scheduler Decision
2 Task Preemption

3 Task executes WAIT-FOR-EVENT
4 Blocking Event occurs

State diagram of a preemptive C-Tasks with blocking

Mar-08

UCI
DREAM Lab



Modern Systems with C-Tasks

- In many modern complex systems, both TT and ET tasks are needed together.

In ET OS the dynamic resource management is extensive:

- ◆ Dynamic CPU allocation (next lecture)
- ◆ Dynamic memory management
- ◆ Dynamic Buffer allocation and event driven management of the communication activities
- ◆ Explicit synchronisation between tasks including semaphore queue management and deadlock detection.
- ◆ Extensive interrupt management

It is beyond the state of the art to formally analyse the timing of ET operating systems (e.g., OSEK).

Mar-08

UCI
DREAM Lab



Inter-process Communication

- Two types of Messages
 - Event msg
 - State msg
- Common region of data: Related to the state msg mechanism.

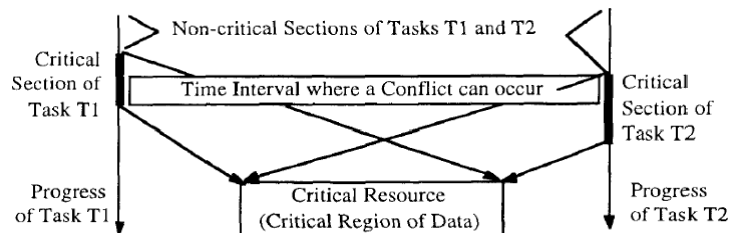


Figure 10.5: Critical task sections and critical data regions.

Mar-08

UCI
DREAM Lab



Inter-process Communication

- Semaphore
- **TT systems**
 - Implicit synchronization (maintaining data consistency without using semaphores) is possible
 - Two tasks with critical sections that access the same region of data can be coordinated a priori such that they never overlap.
- **ET systems**
 - The overhead of semaphore operations can be reduced if every task gets a private copy of the global data at the time of task activation and the OS updates the global data after task termination.

Mar-08

UCI
DREAM Lab



Inter-process Communication - Non-Blocking Write (NBW) Protocol

- A lock-free sync protocol
- Applicability to State msgs was advocated initially.
- Atomic access to CCF (concurrency control field) must be guaranteed by hardware.

Initialization: CCF := 0

Writer:

```
start: CCF_old := CCF;  
      CCF := CCF_old + 1;  
      <write into data structure>  
      CCF := CCF_old + 2;
```

Reader:

```
start: CCF_begin := CCF;  
      If CCF_begin = odd  
      then goto start;  
      <read data structure>  
      CCF_end := CCF;  
      If CCF_end ≠ CCF_begin  
      then goto start;
```

Mar-08

UCI
DREAM Lab



Inter-process Communication - Non-Blocking Write (NBW) Protocol

- An upper bound for the number of read retries exists if the time between write operations is significantly longer than the duration of a write or read operation.
- A significant innovation !
- It can be extended to handle event msgs to some extent !

Mar-08

UCI
DREAM Lab



Time Management

- In many real-time applications, the majority of tasks will be time-triggered, either at a priori known point in time or at dynamically established points in time.
- The OS must provide flexible time management services to simplify the application software.
 - Clock Synchronization
 - Provision of Time Service
 - 'now'
 - time-stamping
 - do at T

Mar-08

UCI
DREAM Lab



Error Detection

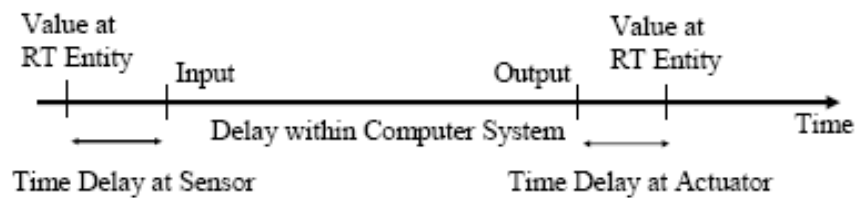
- Monitoring task execution times
- Monitoring inter-arrival periods of interrupts
 - To detect and disable the interrupt line to reduce the probability of erroneous sporadic interrupts
- Double execution of tasks
- Watchdogs
 - heartbeat
 - challenge-response

Mar-08

UCI
DREAM Lab



Timing at an I/O Interface



Mar-08

UCI
DREAM Lab



The Dual Role of Time

A significant event that happens in the environment of a real-time computer can be seen from two different perspectives:

- ◆ It defines the point in time of a value change of a RT entity. The precise knowledge of this point in time is an important input for the later analysis of the consequences of the event (*time as data*)

Example: Downhill skiing

- ◆ It may demand immediate action by the computer system to react as soon as possible to this event (*time as control*).

Example: Emergency stop

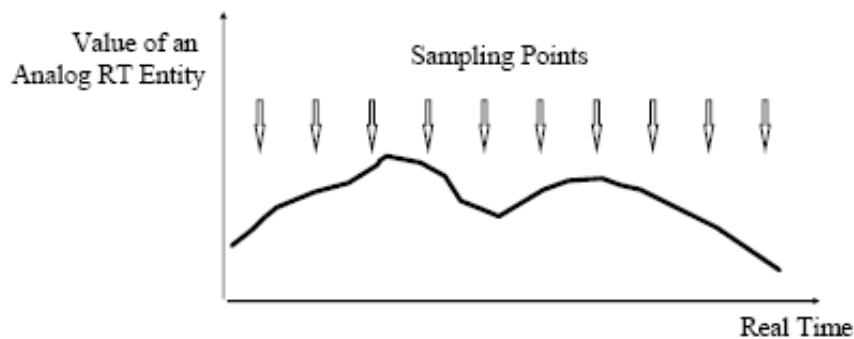
It is much more demanding to implement *time as control* than to implement *time as data*!

Mar-08

UCI
DREAM Lab



Sampling

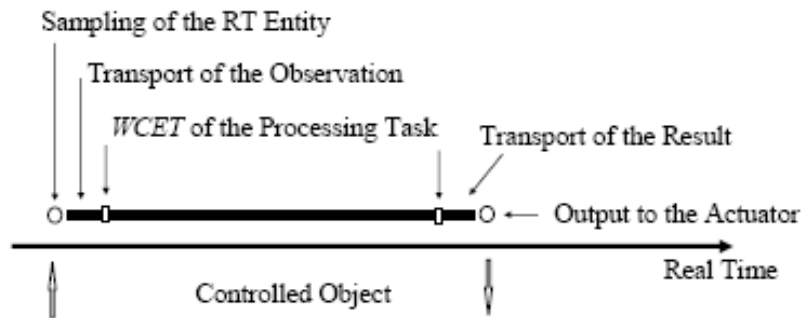


Mar-08

UCI
DREAM Lab



Timing in a Sampled System



Mar-08

UCI
DREAM Lab



Sampling – States vs Events

Sampling refers to the periodic interrogation of the state of a RT entity by a computer.

The lengthening points is called the sampling interval.

The length of the sampling interval is determined by the dynamics of the real-time entity.

States can be observed by sampling.

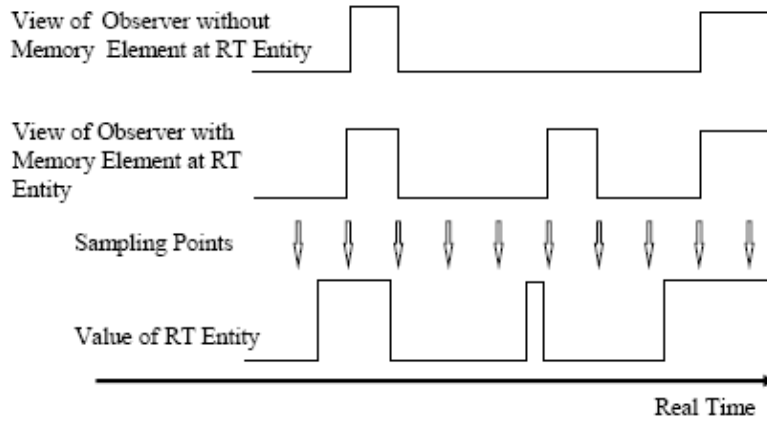
Events cannot be sampled. They have to be stored in an intermediate memory element (ME).

Mar-08

UCI
DREAM Lab



Sampling with and without Memory



Mar-08

UCI
DREAM Lab



Sampling – Position of Memory Element

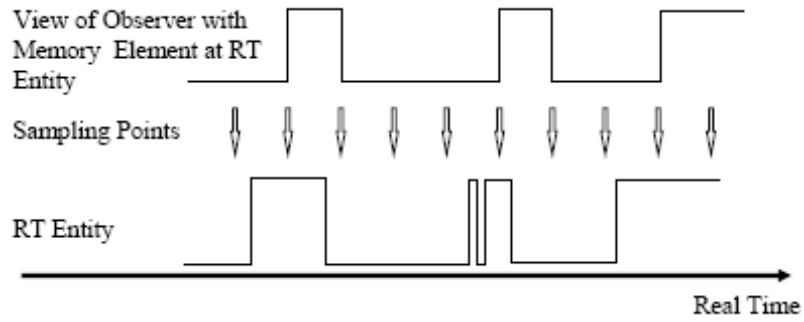


Mar-08

UCI
DREAM Lab



Sampling – Position of Memory Element



Mar-08

UCI
DREAM Lab

