Reinder J. Bril, r.j.bril@tue.nl
TU/e Informatica, System Architecture and Networking

Real-Time Architectures 2003/2004

Summary

Reinder J. Bril

Reinder J. Bril, r.j.bril@tue.nl
TU/e Informatica, System Architecture and Networking

Summary

- Response times for FPPS
  - Without jitter
  - With release jitter
  - Completion (or end) jitter
  - Example
- Resource reservation

Reinder J. Bril, r.j.bril@tue.nl
TU/e Informatica, System Architecture and Networking

Response times for FPPS

Worst-case response times	Best-case response time
$WR_j = \sup_{\varphi,i} R_{j,i}$	$BR_j = \inf_{\varphi,i} R_{j,i}$
Critical instant	*Optimal* instant
$x = C_j + \sum_{i<j} \left\lceil \frac{x}{T_i} \right\rceil C_i$	$x = C_j + \sum_{i<j} \left(\left\lceil \frac{x}{T_i} \right\rceil - 1 \right) C_i$
smallest positive solution highest lower bound	*largest* positive solution lowest upper bound
solved by means of an iterative procedure	
can also be determined by means of a time line	

Reinder J. Bril, r.j.bril@tue.nl
TU/e Informatica, System Architecture and Networking

Response times with release jitter *AJ*

Worst-case response times	Best-case response time
$WR_j = \sup_{\varphi,i} R_{j,i}$	$BR_j = \inf_{\varphi,i} R_{j,i}$
Critical instant*	*Optimal* instant*
$x = C_j + \sum_{i<j} \left\lceil \frac{x + AJ_i}{T_i} \right\rceil C_i$	$x = C_j + \sum_{i<j} \left(\left\lceil \frac{x - AJ_i}{T_i} \right\rceil - 1 \right)^+ C_i$
smallest positive solution highest lower bound	*largest* positive solution lowest upper bound
solved by means of an iterative procedure	
can also be determined by means of a time line	

Reinder J. Bril, r.j.bril@tue.nl
TU/e Informatica, System Architecture and Networking

Completion (or end) jitter *EJ*

- EJ_j of a task τ_j *excluding* release jitter:

$$EJ_j = \sup_{\varphi,i,k}(R_{j,i} - R_{j,k}) \le WR_j - BR_j$$

- EJ_j of a task τ_j *including* release jitter:

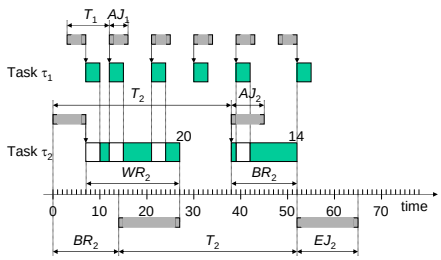
$$EJ_j \le AJ_j + WR_j^* - BR_j^*$$

  – Where the * denotes: including release jitter

Reinder J. Bril, r.j.bril@tue.nl
TU/e Informatica, System Architecture and Networking

Completion and release jitter

- Example:



Resource Reservation

- Definition:
  - Resource reservation is a technique (or *mechanism*) at the level of an RTOS, providing a *virtual* platform.
- Example: (*T*, *B*), *T* = 20 ms, *B* = 5 ms.
- Multiple resources:
  - (co-) processor(s), memory, bus, network, …
- Motivation:
  - Priorities used to/for:
    - Guarantee deadlines (FPS and DPS);
    - Assign relative importance (FPS only);
    - Co-operation protocols + concurrency control.