

E&CE 354 S11 RTX P2 Description

Part A: Due at 4:30pm June 07th 2011

Part B: Due at 8:30am June 27th 2011

Summary of RTX P2 Requirements

1 Scheduling Strategy

Four user priority levels plus an additional “hidden” priority level for the Null process, preemption is *optional*, no time slicing, FIFO discipline at each priority level.

2 RTX Primitives

Primitives as defined in sections 4.1, 4.2, 4.3 and 4.5 of the E&CE 354 Project Description for Spring 2011 term with preemption being *optional*.

3 RTX Processes

Null Process as defined in sections 5.1.1. You need to include 6 test processes as described in section 5.3.4 in your RTX process table. You are not required to implement any testing cases for now. Use dummy test processes code posted on the lab website for RTX P2 submission. Source code of some sample testing processes with simple testing case implementation would be posted on the lab web site to help you develop and debug your code.

4 RTX Initialization

Section 6 as defined in the E&CE 354 Project Description for Spring 2011 term.

5 Error Detection and Recovery

At minimum, the RTX kernel must detect one type of error: an attempt to *send_message* to or *set_process_priority* of a non-existent process_id. The primitive will return an error code (a non-zero integer value). No error recovery is required. It may be assumed that the application processes can deal with this situation.

6 Documentation

Document the design of memory management, processor management, inter-process communication and Null Process. The documentation should include pseudo code to describe the high level design. You can build up your documentation based on the documentation you submitted in RTX P1.

Deliverables and Demo

Part A

This part focuses on the process initialization, context switching and scheduling

- a) Complete the RTX initialization part for both system processes and application processes. For now you only have one system process which is the null process and six application processes which are the dummy testing processes.
- b) Implement the *release_processor* primitive as defined in section 4.2 in the project description
- c) Implement the *set_process_priority* and *get_process_priority* primitives as defined in section 4.5 in the project description with preemption being *optional*.
- d) Implement the null process as defined in Section 5.1.1 of the project description.
- e) Document the design of items in a) – d) and include the pseudo code of implemented primitives.

Zip source code and documentation for items a) - e) and name it p2a_Gid.zip (refer to Section 3.3 of the project description) and submit it to the course book system. Reserve a demo time in the Course Book System.

NOTE: If Part A is not functioning, you will not be able to continue to Part B.

Part B

This part focuses on the inter-process communications and memory management.

- a) Complete the RTX initialization (refer to Section 6 in the project description) part for memory management
- b) Complete the primitives of *request_memory_block* and *release_memory_block* as defined in Section 4.1 the E&CE 354 project description. Preemption is **optional** for now.
- c) Implement *send_message* and *receive_message* primitives as defined in Section 4.3 of the E&CE 354 project description. Preemption is **optional** for now.
- d) Document the design of items a)-c) and corresponding pseudo code.

Zip all the source code and documentation of items in a) - d) and name the file p2b_Gid.zip (refer to Section 3.3 of the project description). Submit the zip file to course book system. Reserve a demo time in the Course Book System.

The RTX P2 implementations (both Part A and Part B) will be demonstrated to project TAs in E2-2363. **All group members** are required to attend the demos of the RTX P2. Each group has about 20 minutes to demonstrate each part of the P2. Use the Course Book System to reserve a demo time for each part of RTX P2. An announcement will be made to the class when the Course Book System is ready to accept RTX P2A and P2B demo reservations.