Wednesday, May 12, 2004

University of Glasgow

DEGREES OF M.Eng., B.Eng., B.Sc., M.A. and M.A. (Social Sciences)

COMPUTING SCIENCE 4H:
REAL TIME AND EMBEDDED SYSTEMS

(Answer 3 out of 4 questions.)

Section A

1. We discussed several priority-driven scheduling algorithms for systems of independent, pre-emptible, periodic tasks on a uniprocessor.

(a) Describe the operation of each of the following types of algorithms. Give one specific example of each algorithmic type, justifying your choice.

(i) fixed-priority

(ii) task-level dynamic-priority (and job-level fixed-priority)

(iii) job-level (and task-level) dynamic-priority

[6]

(b) Cost and packaging requirements for a control system that you are designing, where the system consists of independent, pre-emptible, periodic tasks, require you to operate one of the processors in the system at a utilization of 0.99. All of the tasks have relative deadlines identical to their periods. Which of the specific algorithms that you identified in part (a) above would you choose as the scheduling algorithm for your system? If more than one would work, indicate which you would prefer. Explain your answers.

[4]

(c) You have been provided with the following system definition:

$\mathbf{T} = \{T_1 = (0, 2, 0.5), T_2 = (1, 4, 1.5), T_3 = (0, 5, 1.75)\}$

(i) Is this system schedulable using either of the task-level dynamic-priority algorithms you identified in part (a) above? Explain your answer.

[5]

(ii) Is this system schedulable using the fixed-priority algorithm answered in part (a) above? Explain your answer

[5]

2. Most of the theoretical results in the course text and discussed during lectures are with respect to systems consisting solely of periodic tasks: all jobs are preemptible at all times, jobs never suspend themselves, context switches are free, there are an infinite number of priority levels available, and jobs do not compete for resources. This question is focused on how these "practical" factors affect our ability to validate real embedded systems.

(a) Non-periodic tasks: we discussed several techniques for correctly scheduling non-periodic jobs in priority-driven systems. Describe how each of the following types of server algorithms works (in terms of release, replenishment and consumption), assuming that $T_s = (\Box_s, p_s, e_s, D_s)$:

(i) polling server

[3]

(ii) deferrable server

[3]

(iii) sporadic server

[4]

(b) If the system you were designing and building demanded that you be able to minimize the response times of all aperiodic jobs without destroying the correctness of the schedule with respect to the periodic tasks, which of these types of server would you use and why?

[2]

(c) In a real system, jobs will often contend for access to resources.

(i) What is a priority inversion? How can exclusive access to resources cause priority inversion to occur?

[1]

(ii) Compare and contrast the Basic Priority-Inheritance Protocol and the Basic Priority-Ceiling Protocol in terms of their scheduling, allocation, and priority-inheritance rules.

[6]

(iii) You are constructing a system for the next Mars lander, and a variety of considerations require that you use a fixed-priority scheduling algorithm for it. Some of the jobs in the system will compete for exclusive access to a critical resource, and it is essential that the system not deadlock. Which of these two access control protocols would you use and why?

[1]

Section B

3. **(a)** Describe the real-time scheduling interface provided by the POSIX 1003.1b standard. Your answer should outline the features of the API, and briefly explain the differences between the different scheduling policies.

[5]

(b) Describe how rate monotonic scheduling may be implemented using the features of the POSIX real-time scheduling interface.

[4]

(c) Implementations of the POSIX real-time scheduling interface support only a limited number of priority levels. Explain how, and in what circumstances, performance might be degraded when a rate monotonic schedule is mapped onto a system that supports fewer priority levels than there are tasks.

[5]

(d) Discuss how a future extension to the POSIX real-time scheduling interface could be designed to efficiently support earliest deadline first scheduling. Your answer should include a description of any changes to the real-time scheduling API, and an outline of how the operating system scheduler queues could be organized and manipulated.

[6]

4. **(a)** A packet switched communications system generates and transmits data packets with constant inter-packet spacing. A receiver measures the inter-packet spacing on arrival, and observes some deviation from the expected constant spacing. Assuming the sender is working correctly, give three reasons why the inter-packet spacing may be disrupted, and outline the effect each has on the packet timing.

[6]

(b) Priority queuing can be used in a packet switched network to ensure that certain packet flows receive a guaranteed amount of capacity. What features, in addition to the priority queues, must be added to the network to support this? Why are those features needed?

[4]

(c) Describe the operation of the weighted fair queuing packet scheduling algorithm. Your answer should include:

(i) A diagram showing the flow of packets from an input link, through the packet classifier onto the required queues, and on to the output link; and

(ii) An explanation of how entries on the various queues are manipulated, including a description of the algorithm by which packets are ordered for transmission.

You will need to refer to the "finish number" in your answer, but you do not need to explain how the finish number of a packet is calculated.

[6]

(d) What is the difference in complexity between the scheduling decision for weighted round robin queuing and weighted fair queuing? Describe the cause of this difference in complexity, and outline the impact it might have on the packet forwarding performance of a router.

[4]