



University  
of Glasgow

**SAMPLE EXAMINATION**  
(Duration: 2 hours)

**DEGREES OF MSci, MEng, BEng, BSc, MA and MA (Social Sciences)**

**ADVANCED OPERATING SYSTEMS (M)**

**(Answer 3 out of 4 questions)**

**This examination paper is worth a total of 60 marks**

**You must not leave the examination room within the first hour or the last half-hour of the examination.**

1. You are given a system of independent periodic tasks to be scheduled on a single processor in a pre-emptive manner:  $T = \{T_i\}$  for  $i = 1 \dots n$  where  $T_i = (\phi_i, p_i, e_i, D_i)$  for each  $i$ .
- (a) Assume  $p_i = D_i$  for each  $i$ . What is the maximum schedulable utilization of the Earliest Deadline First algorithm for this system? Is this a necessary and sufficient condition?  
[3]
  - (b) Assume that  $p_i \geq D_i$  for each  $i$ . What is the maximum schedulable utilization of the Earliest Deadline First algorithm for this system? Is this a necessary and sufficient condition?  
[3]
  - (c) Assume that  $p_i = D_i$  for each  $i$ . What is the maximum schedulable utilization of the Rate Monotonic algorithm for this system? Is this a necessary and sufficient condition?  
[3]
  - (d) Assume that  $p_i \leq D_i$  for each  $i$ . Under what conditions will the maximum schedulable utilization of the Rate Monotonic algorithm for this system be identical to that stated in response to part (a) above?  
[3]
  - (e) You are provided with the following system definition (all tasks are independent, and are scheduled pre-emptively on a single processor system):  $T_1 = (0, 2, 0.4, 2)$ ,  $T_2 = (1, 4, 1, 4)$ , and  $T_3 = (0, 5, 1.5, 5)$ 
    - (i) Can these tasks be scheduled using the Earliest Deadline First algorithm? Explain your answer.
    - (ii) Can these tasks be scheduled using the Rate Monotonic algorithm? Explain your answer.

The parameters of the system are changed, such that  $T_3 = (0, 8, 4, 8)$ .

    - (iii) Can this new system be scheduled using the Earliest Deadline First algorithm? Explain your answer.
    - (iv) Can this new system be scheduled using the Rate Monotonic Algorithm? Explain your answer.

[2+2+2+2]

2. You have been hired into a software engineering firm to replace a design engineer that has recently left. His legacy is the design of a real time embedded system. Your first task upon arrival is to critically review the design, since the implementation phase is to start within two weeks.

- (a) The system consists primarily of  $N$  independent periodic tasks that must meet their deadlines, along with random aperiodic jobs involved in the user interface of the system. It is desirable to minimize the average response times of the aperiodic jobs. All tasks execute on a single processor, and are pre-emptively scheduled. Your predecessor had decided to use a simple deferrable server. The total utilization of the periodic jobs is  $U_p$ , and the maximum utilization permitted while still being able to meet all deadlines is  $U_{max} > U_p$ . Your predecessor has specified that the deferrable server size should be  $U_{max} - U_p$ .

Do you agree with his choice? If so, indicate why; if not, how would you change the design?

[5]

- (b) Various reasons led to using an earliest deadline first scheduling algorithm for the periodic tasks and the server. Several of the periodic tasks compete for exclusive access to a shared resource, and it is essential that the system not deadlock. Your predecessor designed the system to use the priority inheritance protocol to minimize blocking due to resource contention. Do you agree with his choice? If so, indicate why; if not, how would you change the design?

[5]

- (c) We usually assume that the context switch time is negligible when determining if a system can be scheduled. Your predecessor has instrumented a prototype of the running system, and has determined the maximum context switch time for jobs in execution to be  $T_{CS}$ . He has, therefore, increased the execution time for each of the periodic tasks (including the bandwidth-preserving server) by  $2 \times T_{CS}$ .

Do you agree with this approach? If so, indicate why; if not, how would you approach this problem differently?

[5]

- (d) Each job in periodic task  $T_i$  queries an array of sensors; each query to each sensor takes  $\sim 1\text{ms}$  to complete, and there are  $N$  sensors to be queried serially; after issuing the query, the job self-suspends until the I/O completes. Your predecessor has accounted for this situation by increasing the execution time  $e_i$  for jobs in the task by  $N \times T_{CS}$ .

Do you agree with this approach? If so, indicate why; if not, how would you approach this problem differently?

[5]

3. (a) A simple form of automatic heap management is *reference counting*. Briefly explain how a reference counting system manages memory allocated to objects, and explain under what situation it can fail to reclaim objects. [5]
- (b) Describe one advantage of reference counting over other forms of automatic heap management? [2]
- (c) *Tracing garbage collection algorithms* are often used as alternative to reference counting for automatic heap management. Briefly outline the concepts behind the operation of tracing garbage collection. [4]
- (d) A way of implementing tracing garbage collection is to use a *copying collector*. Outline what is a copying garbage collector, and explain how a copying collector implemented using *semispaces* works. [9]

4. We discussed the MacOS X I/O Kit as an example of how operating system kernels can be improved through the use of modern object-oriented software engineering languages and practices, to improve the reliability of device drivers. The I/O kit allows the use of a limited subset of C++ in the MacOS X kernel, and organises device drivers into a hierarchy, pushing functionality that is common to several devices up to super-classes from which particular drivers are derived. Outline the advantages and disadvantages of this the I/O kit model compared to the C-based device drivers used in Linux. Discuss the trade-off in using object-oriented languages for kernel development in general – is the use of such languages a net win compared to writing kernels in C?

[20]