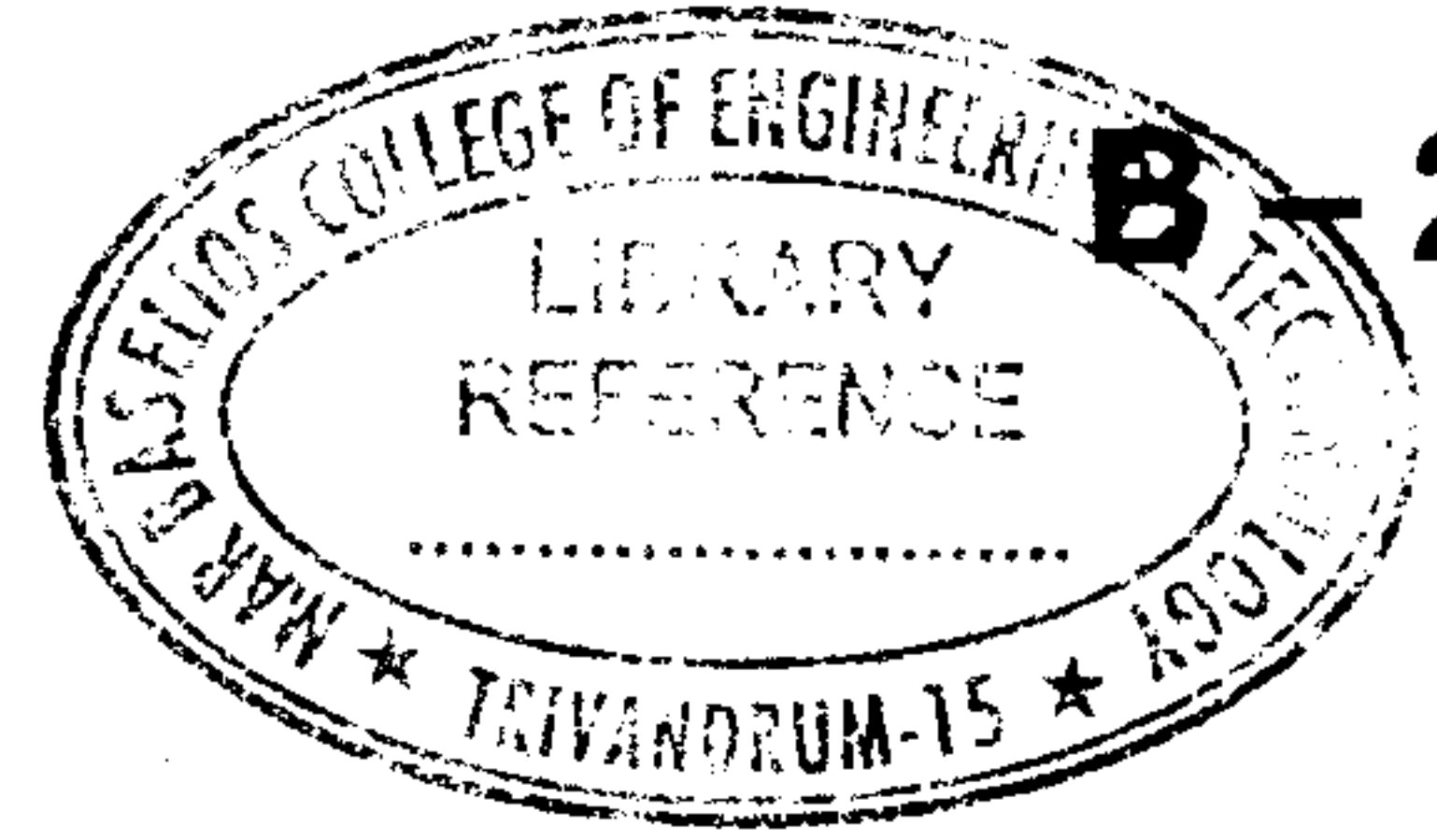




(Pages : 2)



**B - 2926**

Reg. No. : .....

Name : .....

**Second Semester M.Tech. Degree Examination, December 2016  
(2013 Scheme)  
Computer Science and Engineering  
RCC 2001 : OPERATING SYSTEM DESIGN**

Time : 3 Hours

Max. Marks : 60

**Instructions:** 1) Answer *any two* questions from *each* Module.  
2) *All* questions carry *equal* marks.

**MODULE – I**

1. a) What is the difference between a monolithic kernel and a microkernel ? Which of these designs are followed by the Linux kernel ? What are its main features ? 4
- b) What is Complete Fair Scheduling ? What is the reason for imposing minimum granularity in this scheduling scheme ? 6
2. a) Which system calls are associated with process creation ? What optimization makes process creation faster in Linux ? 5
- b) Differentiate between user preemption and kernel preemption. When do they normally occur ? 5
3. a) What scheduling policy will be followed by the Linux kernel for scheduling a MATLAB program and an mp3 player ? Why ? 5
- b) What are the steps to be followed to implement a new system call ? How will you access it from user space ? 5

**MODULE – II**

4. a) What is the reason behind splitting Interrupt Service Routines to top and bottom halves ? 5
- b) What are reader-writer semaphores ? Why do we need them ? 5

P.T.O.



- 5. a) How are shared interrupts handled in Linux ? 5
- b) Differentiate between Big Kernel Lock, semaphores and spin locks. 5
- 6. a) Can we use a semaphore in a bottom half implemented using softing ?  
Justify your answer. 5
- b) Explain the following. How are they different in sequencing events ? 5
  - 1) Sequential Locks.
  - 2) Completion variables.

MODULE – III

- 7. a) Explain the design of the slab layer. 5
- b) What is a block device ? How are they accessed ? 5
- 8. a) Explain the concept of device models, kernel objects and sysfs. 6
- b) What is high memory ? How are they mapped into kernel address space ? 4
- 9. a) Explain Deadline I/O scheduler and anticipatory I/O scheduler. 5
- b) What are kernel threads ? How is its address space different from that of a user thread ? 5

