



MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

I SEMESTER M.TECH.(DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING) END SEMESTER EXAMINATIONS, NOV 24

SUBJECT: ADVANCED SYSTEM SOFTWARE (CSE - 5114)

REVISED CREDIT SYSTEM

(20/11/2024)

Time: 9:30 am - 12:30 pm

MAX.MARKS: 50

INSTRUCTIONS TO CANDIDATES:-

- Answer **ALL** the questions.
- Missing data may be suitable assumed.

1A.	<p>Figure 1 belongs to which descriptor? Illustrate the fields present in the descriptor and describe the fields related to segment selector.</p> <table border="1"><tr><td>BASE (24-31)</td><td>G</td><td></td><td>O</td><td></td><td>LIMIT (16-19)</td><td>1</td><td>DPL</td><td>S = 0</td><td>TYPE</td><td>BASE (16-23)</td></tr><tr><td colspan="7">BASE (0 -15)</td><td colspan="4">LIMIT (0 - 15)</td></tr></table> <p>Figure 1</p>	BASE (24-31)	G		O		LIMIT (16-19)	1	DPL	S = 0	TYPE	BASE (16-23)	BASE (0 -15)							LIMIT (0 - 15)				5M
BASE (24-31)	G		O		LIMIT (16-19)	1	DPL	S = 0	TYPE	BASE (16-23)														
BASE (0 -15)							LIMIT (0 - 15)																	
1B.	<p>Write a function for a process wishing to wait for a specific condition. The state of the process can be TASK_UNINTERRUPTIBLE or TASK_INTERRUPTIBLE.</p>	3M																						
1C.	<p>Depict and discuss the kernel architecture and list and describe the advantages of modules in the kernel architecture.</p>	2M																						
2A.	<p>a.) Consider an example of an SMP kernel which uses standard instructions. Consider a semaphore implementation, where the down() function decrements and tests the count field of the semaphore with a simple decl assembly language instruction. Analyze, what if two processes running on two different CPU's simultaneously execute the decl instruction on the same semaphore.</p> <p>b.) Illustrate the two fields of a process descriptor, representing the relationships between the process and a processor.</p>	4M																						
2B.	<p>How can the effectiveness of a runnable process be evaluated?</p>	2M																						

2C.	Illustrate the relationships among the application program that invokes a system call, the corresponding wrapper routine, the system call handler, and the system call service routine. Also describe, the set of macros invoked by the kernel threads in wrapper routines.	4M
3A.	Write the function and parameters invoked when the slab allocator creates new slabs, that relies on the buddy system algorithm to obtain a group of free contiguous page frames.	4M
3B.	Illustrate with a neat diagram, virtual file system which acts as an abstraction layer between the application program and the file system implementations. Also, describe the main file system classes supported by virtual file systems.	4M
3C.	How linear addresses are assigned to noncontiguous memory areas? Justify.	2M
4A.	Outline the role of the device driver in kernel environment. Illustrate with a neat diagram, the function calls and function pointers used in the module to add new functionality to a running kernel.	5M
4B.	Design a strategy for implementing special memory pools for high-volume objects in a kernel, considering factors such as pool size, object alignment, and performance trade-offs.	3M
4C.	Propose a strategy for implementing custom log levels in printk() for a new kernel feature, considering the need for both critical error reporting and less severe diagnostic messages.	2M
5A.	Analyze the potential challenges in making a driver portable across different Linux platforms.	5M
5B.	Outline the concept of logic programming model.	3M
5C.	Distinguish between parallel and distributed computing.	2M