

Reg. No. : 

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**Question Paper Code : 21353**

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2015.

Sixth/Seventh Semester

Computer Science and Engineering

CS 2028/CS 605/10144 CSE 22/CS 1005 — UNIX INTERNALS

(Common to Information Technology)

(Regulations 2008/2010)

(Common to PTCS 2028 – Unix Internals for B.E. (Part-Time) Fifth Semester  
Computer Science and Engineering – Regulations 2009)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Give any two characteristics of a UNIX file system.
2. Mention the difference between user and kernel modes.
3. What is the difference between delayed write and asynchronous write?
4. List any four fields present in the superblock.
5. What are pipes?
6. What is the use of the kill system call?
7. What comprise the user-level context of a process?
8. When does a process move from kernel running state to sleeping state?
9. Give any two functions of a line discipline.
10. Mention the use of the copy on write bit in the page table.

LIB

28/1/16 AN

PART B — (5 × 16 = 80 marks)

11. (a) Explain the major data structures of the file subsystem and the process subsystem of the UNIX kernel. (16)

Or

- (b) With a block diagram, explain in detail the architecture of the UNIX system kernel. (16)

12. (a) (i) Explain the structure of a buffer pool. (4)  
(ii) Explain the scenarios that kernel may follow in *getblk* algorithm while allocating a buffer for a disk block. (12)

Or

- (b) (i) Explain the *namei* algorithm that converts a path name to an inode number. (8)

- (ii) Explain the *ifree* algorithm for freeing an inode. (8)

13. (a) Explain the implementation of the following system calls:

- (i) open (8)

- (ii) write. (8)

Or

- (b) Explain the implementation of the *link* system call. Discuss about the different deadlock scenarios that may occur during the *link* call (16)

14. (a) (i) Explain the algorithm for handling interrupts. (8)

- (ii) When does the kernel detach a region from a process? Explain the algorithm that the kernel follows while detaching a region from a process. (8)

Or

- (b) (i) What are signals? How are they handled? (8)

- (ii) Explain the implementation of the *exit* system call. (8)

15. (a) How does the kernel manage space on the swap device? When does the kernel swap a process out? Explain how swapping out of processes is handled by the kernel. (16)

Or

- (b) (i) Explain the algorithm for closing a device. (8)

- (ii) What is a clist? Explain the operations done on clists and cblocks. (8)