

Roll No.

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

B.E / B.Tech (Full Time) DEGREE END SEMESTER EXAMINATIONS, NOV / DEC 2013

COMPUTER SCIENCE & ENGINEERING

Seventh Semester

PTCS 9023 UNIX INTERNALS

(Regulation 2009)

32

Time : 3 Hours

Answer ALL Questions

Max. Marks 100

PART-A (10 x 2 = 20 Marks)

1. Give the difference between a program and a process.
2. List any two advantages of the buffer cache.
3. What is an inode?
4. In algorithm *ifree*, what does the kernel do if it finds the list of inodes in the superblock full?
5. When is the *dup* system call used?
6. What is meant by mounting of a filesystem?
7. What are signals?
8. What comprise the context of a process?
9. When does a process incur a validity page fault?
10. What is a c-list?

Part – B (5 x 16 = 80 marks)

11. (i) Explain the architecture of the UNIX operating system with a neat diagram. (8)
(ii) In algorithm *getblk*, explain the scenarios that the kernel may follow to allocate a buffer for a disk block when the kernel cannot find the block on the hash queue. (8)
12. a) (i) Explain the *ialloc* algorithm that allocates an inode to a new file. (12)
(ii) Give any two reasons for the difference in which free list of inodes and free list of disk blocks are maintained. (4)

OR

b) (i) Explain the *namei* algorithm for converting a path name to an inode. (12)
(ii) List the fields present in the super block. (4)
13. a) Explain the implementation of the following system calls:
(i) *open* (8)
(ii) *read* (8)

OR

b) What are pipes? Discuss in detail the issues involved in reading from pipes and writing into pipes. (16)

14. a) (i) Explain the different states in which a process can be with a process state transition diagram. (8)

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

OR

b) (i) Write short notes on process scheduling. (8)

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

15. a) What is swapping? Explain how the kernel manages swap space and handles swapping. (16)

OR

b) Explain how inter-process communication can be done with message queues. (16)