Roll. No										

B.E./B.Tech. (Full Time) DEGREE END SEMESTER EXAMINATIONS – NOV./DEC.2012
INFORMATION TECHNOLOGY
V Semester (Regulation : 2008)
CS 9023-UNIX INTERNALS

Time : 3 Hrs **Max: 100 Marks**

ANSWER ALL QUESTIONS

PART – A (10 X 2 = 20 Marks)

1. List one system call that interacts with the file system and one system call that interacts with the process subsystem.
2. Differentiate between a program and a process.
3. Define the context of a process.
4. Why should the kernel raise the processor priority level to block out interrupts before removing a buffer from the free list?
5. Differentiate the action taken by the kernel when there are no free buffers and when there are no free inodes. Justify.
6. Why do inode numbers start from 1 and not 0?
7. Where does the kernel store information about a process?
8. How does a process synchronize its execution with that of its children?
9. Does the allocation scheme for a swap device differ from the allocation scheme for file systems? Justify.
10. What are device drivers?

PART – B (5 x 16 = 80 Marks)

11. Explain the structure of the **Buffer pool**. Discuss with diagrams any **three** scenarios the Unix kernel may follow while allocating a buffer for a disk block.

12. (a) (i) How is a file represented internally in Unix? Explain the algorithm that allocates an in-core inode. Give 2 system calls which would invoke this kernel algorithm. (10)

(ii) Assume that an Operating System like UNIX has the following structure for the table of contents in its inode. What is the maximum size of a file that is supported by that OS? 4 entries for direct blocks, 2 for single indirect and 1 double indirect blocks. The block size is 4K bytes. A block number is addressable by a 32 bit integer. (6)

OR

(b) (i) What information is stored in the super block? Explain the algorithm **ialloc** that assigns a disk inode. Give 2 system calls which would invoke this kernel algorithm. (10)

(ii) Differentiate between the free inode list and the free block list maintained by the kernel in the super block of the file system. Justify why this difference is needed. (6)

13. (a) Explain the algorithm which implements the READ system call. Mention clearly the kernel data structures used and how its fields are used / updated.

OR

(b) Explain the algorithm which implements the MOUNT system call. Mention clearly the kernel data structures used and how its fields are used / updated.

14. (a) (i) Explain with a diagram, the various state transitions that a process undergoes during its lifetime in a Unix system. (8)

(ii) Explain any **ONE** algorithm that manipulates regions mentioning input(s), output(s), purpose and system call(s) that uses it. (8)

OR

(b) (i) Explain how the fork system call is implemented in Unix. (10)

(ii) What happens when the following program is run? Explain. (6)

```
#include <signal.h>
  main()
 {

        extern catcher();
        signal(SIGCLD, catcher);
        if (fork()==0) exit();
        pause();
}
catcher()
{
        printf("parent caught sig \n");
        signal(SIGCLD, catcher);
}
```

15. (a) (i) Explain how the swapper process works. (6)

(ii)Explain the data structures used in the demand paging memory management system of Unix. (10)

OR

(b) (i) Explain how the Shell is implemented in Unix. (8)

(ii) Write a short note on Inter process communication in Unix. (8)