

This question paper contains 2 printed pages

Your Roll No

6259

M. Sc. Computer Science / II Sem.

MCS – 202 – Advanced Operating Systems

(Admissions of 2009 and onwards)

Time 2 hours

J
Maximum Marks 50

(Write your Roll No on the top immediately on receipt of this question paper)
Attempt all questions Parts of a question should be answered together

- Q1 a Differentiate between interrupts and exceptions (2)
- b Give the disk layout of Unix file system, describing each component (5)
- c What is the function of *brlse* algorithm? Why is it essential to raise processor execution level in the *brlse*? (3)
- Q2 a Consider the part of the *iget* algorithm given below
- ```

while (not done){
 if (inode in inode cache) {
 if (inode locked) {
 sleep (event inode becomes unlocked),
 continue, /* loop back to while */
 }
 }
}

```
- What is the need for a process to start the loop from the beginning again (refer to comment above in the code), if inode is already there in the inode cache? (3)
- b Using system calls other than the *read()* system call, write a function that returns the size of the file whose name is passed to it as an argument (4)
- c The kernel maintains an exhaustive list of free disk blocks but not of the free inodes on the disk Give reasons (3)
- Q3 a What is wrong if *mknod()* system call is used to create a directory? (2)
- b Consider the following program segment.
- ```

main(int ac, char *av[]){
    if (ac != 2){
        cout << "Usage " << av[0] << " <directory name>",
        exit(1);
    }
    if (chdir(av[1]) == -1)
        cout << av[1] << " is not a directory".
}

```
- Suppose the name of the executable file for the above program is *chToDir* For the following sequence of commands
- ```

/usr/Progs> chToDir OSProgs
XXX>

```

Assuming that *OSProgs* is a valid directory name that exists in the *Progs* directory and the prompt displays the current directory name. What will be displayed at XXX and why? (3)

c Explain with the help of an example, how the order of writes to the disk can minimize the corruption of file system in the event of system failure (5)

Q4 a Explain what happens when a process wants to write data to a pipe that does not have enough space for the data to be written (4)

b The kernel does not allow the *mount* to succeed if the inode reference count of the directory at which mounting is to take place is greater than one. Give an example to show that if this is not followed then it may result in an error/security lapse (3)

c Given the command as *ls -l myFile*, the permissions of *myFile* are shown as *rws-x-x*. What is the significance of 's' in the permissions? What will happen when someone executes this file as *\$myFile*? (3)

Q5 a Describe the three components of the context of a process. (6)

b Briefly explain what is per process region table and how is it related to the region table? (4)

HowToExam.com