INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY BANGALORE
Mid Term Examination - Term II (2023-24)

EG301 Operating Systems

Time: 2 Hours
Max. Marks: 60

PART A (MCQ)
Marks 20 X 1 = 20

NOTE: Attempt all subparts of PART A in one place on the first page or two of the answer sheet.
In each case select the most appropriate single answer among the choices given

1. A process P opens a file named "*data*" to read and calls `dup()` once. Another process Q also open the same file to read. Assuming that only P and Q have the file currently open and the call to dup is complete, the number of entries in the open file table corresponding to the file will currently be:
   a. 1
   b. 2
   c. 3
   d. 4
2. In a filesystem like ext2 a student says that in one directory a file called "one" and in another directory a file called "two" have the same inode number. Which of the following is true:
   a. one must be a softlink to the other
   b. one must be a hardlink to the other
   c. the student's observation is faulty since inode numbers must be unique
   d. one was created from the other using the Linux "mv" command.
3. The inode in an ext2 type filesystem contains a lot of meta-information about a given file. Which of these is *not* stored in the inode for a given file:
   a. file name
   b. file size
   c. file type
   d. file permissions
4. In the process of creating a filesystem on Linux starting from a disk which of the following statements is true:
   a. mkfs and fdisk are both used to create a filesystem on a given partition
   b. mkfs is used to create partitions and on a partition fdisk creates a filesystem
   c. fdisk is used to create partitions and on a partition mkfs creates a filesystem
   d. mkfs can be used to mount a filesystem onto an empty directory
5. Which of these is true:
   a. List of blocks allocated to a file is available from the blocks bitmap
   b. Inodes bitmap indicates which inodes are free
   c. Inode table is stored inside the MBR
   d. blocks bitmap is stored in the MBR
6. If a file is opened twice in the same program using the open() system call giving file descriptors fd1 and fd2, then:
   a. reading a character usng fd1, followed by reading a character using fd2 gives the same character in both cases.
   b. using close(fd2) will then cause fd1 to be not usable.
   c. using dup(fd1) results in three file descriptors pointing to the same open file table entry.
   d. after reading a character using fd1, using dup(fd1) will cause a new file descriptor to be created and this new file descriptor will be set to read from the beginning of the file.
7. Which of the following is true:
   a. VFS uses a generic inode table in the RAM which is the same as that on the disk
   b. VFS is a filesystem on the disk similar to ext2
   c. VFS only contains directory information from the disk
   d. VFS implements the user system call interface to file operations such as open() and read()
8. IO caching of data from disks helps to
   a. reduce amount of IO to the disk
   b. reduce space occupied on the disk when the file is saved
   c. reduces the fragmentation of files saved on the disk
   d. reduce the speed of transfer of data blocks between disk and RAM

9. The purpose of a *sync( )* system call is
    a. help different process to synchronize with each other
    b. increase number of data blocks cached from a disk file
    c. case a process to wake up from sleep
    d. ensure file changes are written to the hard disk
10. When a fork() is called after a file is open, which of these is true:
    a. Both processes share the same file descriptor table
    b. They have different but identical copies of the file descriptor table, but use the same open files table entries
    c. They have different but identical copies of the file descriptor table and also duplicated entries in the open files table
    d. The file corresponding to each file descriptor is automatically closed in the parent and then opened in the child.
11. A process P calls fork() at time t=0. At time t=1 the child calls exit(10). At time t=2 the parent P calls wait(). Which of the following holds:
    a. The wait call returns immediately since the child has exited already.
    b. The wait call keeps waiting.
    c. The wait call returns an error/failure, because the child has already exited, so nothing to wait for.
    d. The wait call returns no value, because the child has already exited.
12. When the system call read() is called from a user's process which of the following happen to the CPU:
    a. The CPU goes from kernel mode to user mode
    b. The CPU will likely go to sleep because the system call often results in disk IO
    c. The CPU will start executing the handler for the read system call
    d. The CPU will not change user/kernel mode settings
13. The following is true about the exec() call:
    a. exec() closes all open file descriptors
    b. exec() creates a new process and deletes the old process
    c. exec() creates a new process which is a replica of the old process except for the process id
    d. if a call to exec() returns any value, then it has failed
14. When a parent process exits before one of its children on Linux, then
    a. the child is automatically terminated
    b. the child is said to be orphaned and its parent PID will change
    c. the child goes into a zombie state
    d. the child can use wait to wait for the exit of the parent

15. A C program calls fprintf() to display a string on standard output. Which of the following is true:
    a. fprintf() is a system call and does not make any library call.
    b. fprintf() is a system call, but can also result in the system call write().
    c. fprintf() is a C library call and may result in the system call write().
    d. fprintf() can be used to output to a file on the disk, but not to the standard output.

16. A program has a global integer variable v initialized to 0 and a function f(). The variable v is incremented by 1 each time f() is called. The process when running calls f() once and then calls fork(). the return value of fork is stored in another variable r. Which of the following holds:
    a. Just after fork, the expression (r==v) is true in both parent and child.
    b. Just after fork, the expression (r==v) is true in the child.
    c. Just after fork, in the parent r is non-zero and v is 0.
    d. Just after fork, in the child r is 0 and v is 1.
17. A process moved from blocked to ready state, which of the following is *NOT* a good reason to explain this happening:
    a. The process was blocked on an input and the data became available
    b. Another process, which was on the CPU, executed a divide by zero.
    c. The process was waiting for a child to exit and the child exited.
    d. The process was executing a sleep and the time was up

18. Which of these is true for the round robin scheduling algorithm:
    a. Smaller time slot is always a better thing, it improves overall usage of the CPU
    b. Smaller time slots means more overall time spent switching context
    c. Larger time slots improve the response time
    d. Larger time slots increase the average turn around time.
19. Three processes A, B and C with running times $R_A < R_B < R_C$ arrive in a system. We compute T_1 the Average TAT with all jobs arriving together and using SJF scheduling. We also compute T_2 the TAT average computed assuming they arrive in the order A followed by B followed by C, with the CPU never being idle in between. Then which of these is true:
    a. $T_1 < T_2$
    b. $T_2 < T_1$
    c. $T_1 = T_2$
    d. The relationship depends on the actual running time values R_A, R_B, R_C
20. A student writes a program with the `main()` body consisting of just one line:
    `{for(int i=0;i<10;sleep(1)) printf("%d ",i++);}` .The **`strace`** of the execution would show:
    a. 0-9 are printed by calling the **write** system call 10 times, the output appears one every second.
    b. 0-9 are printed by calling the **write** system call 10 times, but the output appears all together after 10 seconds.
    c. 0-9 are printed by calling the **write** printf only once.
    d. 0-9 are printed by calling the **write** system call just once after 10 seconds.

PART B (Descriptive) Marks 4x10=40

1. What are the preconditions required in a scheduling system for the occurrence priority inversion? Construct one example to demonstrate priority inversion. Enumerate the sequence of executions/arrivals and indicate at what point priority inversion occurs.
2. Assume that the function **setup()** sets up three variables (a) **fname** (a pointer to a string with a file name like "/tmp/blah"), **command** (a pointer to a string with a linux command like "cat") and **args** (the arguments list to be used when calling execvp()). **setup()** is assumed given, **don't write it**. With this, write a C program (with only a main() function) to use execvp and run the given **command** so that when it executes, it's standard input is *redirected* to come from the file given by **fname**). (*Note: Exact syntax is not important, but the code flow is.*)
3. We wish to compute the maximum size of a file on our file system. Assume the inode has **10 direct** block pointers, **1 indirect** block pointer, **1 double indirect** block pointer and **1 triple indirect** block pointer. Assume *each level of indirection has 10 pointers*. If a block of data is 2KiB, what is the maximum size of a file on this filesystem (in KiB). Show your computation in detail.
4. This is regarding process scheduling:
    a. What are the five different states of a process (including user mode and kernel mode running process) in an operating system.
    b. Draw a picture showing directed arrows between states indicating possible transitions.
    c. What is the run queue and the blocked queues in relation to this picture.
    d. Consider these events: (1) IO completion for a blocked process, (2) user mode process makes a system call, (3) return from system call, (4) blocking IO, (5) time slice exhausted, (6) process calls exit system call, scheduling a process at the head of the run queue. For each of these related to the transitions indicated above in part (b).