

Unit-1,2**Introduction to UNIX operating system and Shell Programming****Answer in short**

1. What is shell?
2. What do you mean by multitasking operating system?
3. Differentiate between multitasking & multiprogramming.
4. What is Process?
5. Give name of commonly used shell?
6. What is purpose of shell?
7. What is system call?
8. What is mean by multiuser os?
9. Which command is used to give permission to files?
10. What is mean by Portability?
11. Why the UNIX is more portable than other operating systems?
12. What are three levels of security in UNIX?
13. List the four major system calls used for process control.
14. Can a user belongs to more than one group?
15. Can we use octal code for changing permission of group?
16. Write a note on output for the command: `cat f1 f2 > f3`.
17. What is the significant of file `/etc/passwd`?
18. List any two features of UNIX system.
19. Enter this command sequence: `ls | wc -l`. What do you think the output represents.
20. What is the system default permission for directory file?

Answer following questions in detail

1. Describe the feature of UNIX operating system.
2. Describe UNIX architecture in detail.
3. Explain UNIX file hierarchy.
4. Explain system structure.
5. Describe various types of UNIX files.
6. List and explain types of UNIX shell.
7. Describe each field of command `ls -l` command.
8. What are the three levels of security in UNIX? Briefly describe each one.
9. What does a operating system do when it needs to read a file?
10. Why the shell is called a command interpreter?
11. Describe the block diagram of system kernel.
12. Which command is used to change permission of directory and file? Explain how to change permission using symbolic code.
13. Define a wildcard. Why do we use wildcards?
14. Explain any three wildcard with example.
15. What do you mean by hard link and soft link? Explain with example.
16. Explain pipe feature of shell.
17. What are system calls and what role do they play in the system?
18. Differentiate user mode & kernel mode.
19. Describe different tasks performed by the kernel.
20. What is the difference between the `cmp` and `diff` command?

Multiple Choice Questions

- Which command is used to display the name of the file and directories that consist of at least two characters:
 - ls??*
 - ls
 - ls *
 - ls ?*
- Which command is for setting default permission of files and directories :
 - chmod
 - default
 - umask
 - chstat
- Which command is used to know currently login users?
 - ls
 - who
 - list
 - all
- What among the below can be considered as core of the system?
 - Kernel
 - Shell
 - Operating system
 - None of the given
- Which of the following command used to display hidden files?
 - ls
 - ls -l
 - ls -a
 - ls -h
- Which of the following statement describes shell in UNIX?
 - Shell is operating system
 - Shell is an interface to the UNIX system
 - Shell is a kernel
 - Shell is an application program
- Which of the following command is used to move in parent directory from the current directory?
 - cd <parent directory name>
 - cd
 - cd..
 - cd ..
- How can you append the output of a command to a file?
 - command < file
 - command > file
 - command << file
 - command >> file
- Under what circumstances the kernel provide context switch?
 - When a process puts itself to sleep.
 - When it exits.
 - When it returns from a system call user mode but not a eligible process to run.
 - All of the above

10. Which among is not the feature of UNIX operating system?
 - a. Multiuser system
 - b. Multitasking system
 - c. Pattern matching
 - d. None of the above.
11. Which directory is responsible for storing commands?
 - a. /etc
 - b. /home
 - c. /bin
 - d. /usr/bin
12. Which of the following represent octal permission 515?
 - a. rwxr-xrwx
 - b. r-x--xr-x
 - c. -wx--xrwx
 - d. rwxrwxrwx
13. Which of the following is used to count the number of files in current directories?
 - a. ls-l | wc-l
 - b. ls-l | wc-c
 - c. ls | wc-w
 - d. ls | wc-l
14. Which of the following command is used to check permission?
 - a. chmod
 - b. ln
 - c. ls
 - d. ls-l
15. Which of the following is the descriptor for standard output?
 - a. 0
 - b. 1
 - c. 2
 - d. 3

State your choice: True or False.

1. The Unix operating system was further developed to Unix.
2. Two UNIX systems may or may not use the same system call.
3. Unix enables a user to run only one process at a time.
4. UNIX commands must be in lower case and must not have extensions.
5. Kernel directly interacts with shell.
6. Shell using kernel interacts with hardware.
7. Command interpreter is used by shell to translate commands in kernel understanding Language.
8. The ls command displays all file name and directories.
9. Several commands can be attached using the pipe operator.
10. The default number of lines into which the split command splits a file is 100 lines.
11. The ps command shows the process running in the system.
12. The * symbol is used to save the output coming on the screen in a file.
13. According to Dennis Ritchie, UNIX written in higher level language was major disadvantage for reducing speed in computing.

14. UNIX is a command based system.
15. The system wide default permission for a directory is 666.

Place the correct word in the given blank

1. The _____ is a command interpreter in UNIX.
2. The _____ is a core of the operating system.
3. The _____ interacts with the hardware and the _____ interacts with the user.
4. _____ is used to communicate with the kernel.
5. To look manual pages of system call use command _____.
6. The kernel is represented by the file _____.
7. The directory which doesn't have any parent directory is _____.
8. The _____ command displays all file names in a directory.
9. System administrator related commands are located in _____ directory.
10. System's configuration files are available in _____ directory.
11. Standard header files are located in _____ directory.
12. All device files are located in _____ directory.
13. The commands in UNIX are either _____ or _____.
14. To know command is internal or external use _____ command.
15. The command used by all users is located in _____ and _____.
16. To capture a number of filenames with a simple pattern _____ symbol is used.
17. A process in UNIX system can execute in two modes _____ or _____.
18. Executable system utilities are found in _____.
19. Internal commands are built into the _____.
20. C shell is founded by _____.
21. To change file permission use command _____.
22. _____ is the user ID used for superuser
23. _____, _____ and _____ are three primary functions for process control.
24. In chmod command, _____ operator is used to grant and _____ to deny permission.
25. In chmod command, symbolic code _____ is used to grant/deny permission to owner of the file.
26. The _____ variable stores number of positional parameter.
27. To create a new user _____ command is used and to delete user _____ command is used.
28. The _____ command is used to extract columns from the file.
29. A _____ matches to any single character in regular expression.
30. A _____ is used to match beginning of the line.

Unit – 3 The UNIX file system

Answer in short

1. Which kernel data structure stores the information about permissions to access a file?
2. Define inode.
3. List the data structures and their relationship when a regular file is opened.
4. List the fields of super block.
5. How current byte offset is maintained by the file table?
6. What is the purpose of using lseek() system call?
7. What is free list?
8. List system call that uses file descriptor as argument.
9. Why to check errno variable immediately after executing system call, if fails.
10. What is the significant of flag O_CREAT in open system call?
11. Write a system call to create a file with permission 666.
12. What is atomic operation?
13. List parameters that are in in-core inode not in disk i-node.
14. Why superblock maintain partial list of free i-node and disk block?
15. Why i-node start from 1 not from 0?
16. Does each entry in the user file descriptor table have a separate file table associated with it?
17. List the information stored in inode block
18. Which system call is used to obtain the attributes of a particular file?
19. What does directory entry consist of?
20. What do you mean by file descriptors?
21. What happened to buffer when it assigned to any process?
22. How kernel retrieves free buffer from the free list?

Answer following questions in detail

1. Describe structure of UNIX file system.
2. Explain in detail about the structure of regular file.
3. State the importance of file table in UNIX system. What information is stored in it.
4. Explain the system call used for creating the special files.
5. Describes the content of super block.
6. Draw a diagram representing kernel data structures after executing open().
7. Differentiate between open() and creat() system call with example.
8. Demonstrate usage of read () & write() system call for copying content of file1 to file2.
9. Explain how i-node is assigned to a new file.
10. Explain how data block is assigned to a file.
11. Explain data structure associated with the open files maintain by the kernel.
12. In which situation value of in-core i-node reference count is more than one.
13. Briefly explain file creation concept.
14. State the purpose of in-core inode entries in detail which are not available in (disk) inode.
15. Which system call is used to read inode entries? Explain in detail with example.
16. Explain algorithm for converting the path name to inode.
17. Specify how the open () system call is used to emulate the function performed by the shell's (i)> (ii)>> symbols.
18. Describe any one algorithm from following in detail.
ialloc, iput, bmap, ifree, alloc, iget
19. Explain namei algorithm.

21. What is a link? Explain the types of links. How are links created?
22. Explain the following system call.
 - i. open ii. fcntl iii. link iv. stat v. write vi. read vii. close

Multiple Choice Questions

1. Which of the following information does not stored in inode?
 - a. File size
 - b. File name
 - c. File owner
 - d. File group
2. Directory entry consist of
 - a. File type, file name and file size
 - b. File type, file name and i-node
 - c. File type and i-node
 - d. File name and i-node
3. What system call read a byte at a time from the standard input?
 - a. read(0,buff,0);
 - b. read(1,buff,1);
 - c. read(2,buff,1);
 - d. read(0,buff,1);
4. Which of the following file descriptor represent standard output?
 - a. 0
 - b. 1
 - c. 2
 - d. None of above
5. Which system call is used to read content of the file?
 - a. get
 - b. fread
 - c. getch()
 - d. read()
6. When kernel retrieves block, its buffer is found in hash queue and is free, which of the following sequence occurs? (multiple choice)
 - a. the buffer removed from hash queue
 - b. the buffer marked as busy
 - c. the buffer removed from the free list
 - d. the buffer added into hash queue
7. When kernel retrieves block, its buffer is not available in hash queue, which of the following sequence occurred?
 - a. Process waits for buffer is available in hash queue
 - b. A free block is allocated from the head of free list
 - c. A free block is allocated from the tail of free list
 - d. None of above
8. UNIX stores list of free blocks in _____.
 - a. I-node block
 - b. Super block
 - c. Data block
 - d. Both b and c
9. List of free inodes are maintained in _____.
 - a. Boot block
 - b. Super block
 - c. I-node block

- d. Data block
10. Which of the following information is not stored in superblock?
 - a. Size of filesystem
 - b. Size of i-node table
 - c. List of free blocks
 - d. List of allocated i-nodes
 11. Which of the following is stored in in-core inode but not in disk i-node?
 - a. File size
 - b. File permission and type
 - c. Link count
 - d. Reference count
 12. Which of the following is maintained by file table?
 - a. a pointer to in-core inode table
 - b. the offset of next read or write in the file
 - c. access rights (r/w) allowed to the opening process.
 - d. reference count
 - e. all of above
 13. Which of the following system call is used to create a new file?
 - a. open()
 - b. creat()
 - c. both open() and creat() can be used
 - d. none of above
 14. Which of the following system call is used to delete a file?
 - a. delete()
 - b. close()
 - c. unlink()
 - d. link()
 15. A process may query status of a file by using _____ system call.
 - a. open()
 - b. read()
 - c. status()
 - d. stat()/fstat()

State your choice: True or False

1. Binary file is called ordinary file in UNIX.
2. In UNIX all devices are implemented as files.
3. Is it possible to read and write a file even after it is deleted.
4. open() accepts three arguments when it creates a file.
5. creat() does not take file descriptor as an argument.
6. STDIN is a variable of standard input file descriptor.
7. Inode number is a unique file identification number.
8. Super block stores the summarised information of a file system.
9. Inode block stores information of individual file system.
10. Data block stores the free list of file system.
11. Entry of directory is of 16 bytes.
12. Algorithm ialloc is responsible to assign inode number to newly created file.
13. Algorithm ifree is used to free the data blocks used by the file.
14. Block size is unique in all file system available on the system.
15. Free list is maintained to assign data blocks when new file is created.
16. No file can expand beyond the size of data block.

17. umount is a system call used for mounting file system.
18. Dup() is used to make a duplicate copy of a file.

Place the correct word in the given blank

1. The kernel arranges all the buffers in the pull in _____ data structure, so that it can be searched efficiently.
2. A unique number assign to a file is known as _____.
3. Directory entry consist of _____ and _____.
4. A _____ system call is used to duplicate a file descriptor.
5. An unlink() is used to _____ a file.
6. The buffer header uses _____ and _____ field to uniquely identify the buffer in buffer Pull.
7. _____ system call used to move file offset pointer to a specified point.
8. Use _____ flag to create a file with open system call.
9. Open file descriptor 0 represent _____, 1 represent _____ and 2 represent _____.
10. _____ system call is used to delete files.
11. _____ system call is used to read i-node entry using filename.
12. _____ algorithm is for buffer allocation.
13. _____ block contains bootstrap program.
14. Super block stores the _____ indicating that the super block has been modified.
15. Algorithm alloc assigns _____ to a file.
16. The write system call accepts _____, _____ & _____ as arguments to itself.
17. Open file descriptor 0 represent _____, 1 represent _____ and 2 represent _____.
18. _____ system call is used to read i-node entry using filename.
19. _____ system call is used to change file permission.
20. Directory files have at-least _____ links. (Zero, One, Two, Three)
21. The _____ interact with the hardware and the _____ interact with the user.
22. _____ block known as manager of file system.

Unit 4 – The UNIX Process Management

Answer in short

1. Which system call is used to create process?
2. What are various IDs associated with a process?
3. What is context switching?
4. List different process states?
5. List the system calls used for process management.
6. What is zombie process?
7. What is the return value of fork() system call in child process and in parent process?
8. How parent process can wait for death of child process?
9. How parent process can retrieve exit status of child process?
10. Define Orphan Process.
11. What is the difference between orphan process and zombie process?
12. How do you run separate program in child process?
13. What is a daemon process?
14. Give significance of PID & PPID.
15. How do you display all processes running on your system?
16. Name five important process attributes that are inherited by the child from its parent.
17. Differentiate between user mode & kernel mode.
18. Define sleep and wakeup.
19. Define signal.
20. Write the I/O parameters saved in the U area.
21. Define init process.

Answer following questions in detail

1. Write notes on exec functions.
2. How a new process in UNIX is created? What is the new process called? Give the syntax of the corresponding system call. What operations are performed by the kernel for this system call?
3. What is race condition?
4. Write a note on UNIX process.
5. Write a program in C or C++ to emulate the UNIX ls -l command.
6. Write a program in C to obtain process attributes.
7. Write notes on
i) exit ii) wait iii)waitpid
8. Describe the state transition that a process undergoes during its lifetime .
9. Which memory structures are responsible for storing the data related to process which are only accessible to the running process itself? Give your explanation by making use of diagram.
10. List internal steps after execution of fork().
11. State the purpose of using Wait(). Explain wait() with different example of getting status of child.List steps performed after executing exit().
12. Which steps are performed in UNIX operating system after occurrence of interrupt while executing a process in kernel mode?
13. Discuss booting sequence.
14. Explain usage of exec() in detail.
15. What are signals? Name a way of generating a signal from the keyboard. Why should we use kill with signal names rather than their numbers?
16. Is there any process in the system whose PID is 0 (i.e. Zero)? Give significance of it.
17. Explain in detail about context of the process in UNIX system.

Multiple Choice Questions

1. State of a process changes from “run” to “blocked/wait” when:
 - a. Waiting for disk read occurs
 - b. Time slice expires
 - c. Waiting for CPU occurs
 - d. All of the above

2. Which statements concerning process IDs are valid?
 - a. Shell and Subshell have the same process ID.
 - b. The init process is always assigned process ID 1.
 - c. Following a fork system call, father and son have the same process ID.
 - d. None of above is true

3. init runlevel for system shutdown is:
 - a. 3
 - b. 2
 - c. 1
 - d. 0

4. Which statements concerning process IDs are valid?
 - a. Process IDs are stored in the respective i-nodes.
 - b. It is assured that no two concurrently running processes have the same process ID.
 - c. The process IDs are only unique with respect to each user.
 - d. The process ID of the interactive shell may be returned with the command echo \$?.

5. Pid of init process is _____.
 - a. 0
 - b. 1
 - c. 32767
 - d. None of the above.

6. How do you get parent process identification number?
 - a. Waitpid
 - b. getpid()
 - c. getppid()
 - d. parentid()

7. The process which terminates before the parent process exits becomes_____.
 - a.Zombie
 - b.Orphan
 - c.Child
 - d.None of the above

8. If fork() system call returns -1, then it means_____.
 - a.No new child process is created
 - b.The child process is an orphan
 - c.The child process is in Zombie
 - d.None of the above.

9. Fork returns _____ to parent process on success.
 - a.0
 - b.child process id
 - c.parent process id
 - d.none

10. How many times printf() will be executed in the below mentioned program?

```
main() {  
    int i;  
    for (i = 0; i < 4; i++)  
        fork();  
    printf("my pid = %d\n", getpid());  
}
```

a.4
b.8
c.16
d.32

Place the correct word in the given blank

1. _____ system call is used to create process from the existing process.
2. The system call _____ execute a new program in a process.
3. Process attributes are maintained in _____ by kernel.
4. To get value of environment variable in the process _____ system call is used.
5. A parent process can wait for his child process termination using _____ system call.
6. No process can directly moves to _____ mode without getting placed in a _____ queue.
7. _____ system call used to bias the existing priority of a process.