5 Years Integrated M.Sc.(IT) – Semester 6
Question Bank
060010601 – UNIX Internals

Introduction and Overview of UNIX

Answer following in short

1. What is system call?
2. Who are principal designer of UNIX operating system?
3. Write any two points to differentiate between multitasking & multiprogramming.
4. What is kernel in UNIX operating system?
5. List any four features of UNIX system.
6. List the four major system calls used for file system management.
7. Name the data structures used for file system management.
8. Name the data structures used for process management.
9. Write one difference between system call and library function.
10. Write one difference between system call and API.
11. Draw architecture diagram of UNIX operating system.
12. Give any four characteristics of UNIX file system.
13. List any two services of operating system.
14. Draw block diagram of system kernel.
15. What are the responsibilities of file subsystem of UNIX kernel?
16. Justify: "Using large logical blocks increases the effective data transfer rate between disk and memory"
17. State the purpose of boot block.

Long Questions

1. Explain features of UNIX system.
2. Operating systems like UNIX provides services both for program and user. Explain.
3. Explain in detail about UNIX system architecture.
4. Describe the block diagram of system kernel.
5. Write a note on UNIX phenomena the file system has "places" and that processes have "life."
6. Why there are two different mechanisms for accessing hardware control and file subsystem?
7. Describe fie system structure in UNIX operating system.

Multiple Choice Questions

1. Which one among the below translate UNIX commands into action?
   a) Shell b) Kernel c) Command Interpreter d) System call
2. Which one among the below interacts between user and kernel?
   a) Shell b) Kernel c) Both a) and b) d) System call
3. Which one among below directly interacts with the hardware?
   a) Shell b) Kernel c) Compiler d) Both b) and c)
4. Which one among below is false with respect to UNIX operating system?
   a) UNIX considered each and every thing as a file.
   b) Kernel deals directly with user.
   c) File and process are two basic entities that support UNIX system.
   d) Several user can use the system simultaneously.
5. Majority part of UNIX operating system is written using
   a) Assembly language b) C language c) Java language d) C++ language
6. Who is main contributor for development of Linux?
   a) Ken Thomson b) Linux Smith c) Linus Torvalds d) Dennis Richi

7. Which among the below given is not the reason for success of the UNIX system?
   a) System is written in high –level language
   b) It does not use hierarchical file system.
   c) It has not a complex user interface.
   d) It is a multiuser, multiprocessing system
8. What among the below can be considered as core of the system?
   a) Kernel b) Shell c) Operating system d) None of the given
9. Which component among the below translate commands into action in UNIX operating system?
   a) Shell b) Kernel c) Command Interpreter d) System call
10. Which component interacts between user and kernel?
    a) Shell b) Kernel c) Command Interpreter d) System call
11. Under what circumstances the kernel provide context switch?
    a) When a process puts itself to sleep.
    b) When it exits.
    c) When it returns from a system call user mode but not a eligible process to run.
    d) All of the above.
12. Choose the correct one for the statement: All UNIX flavours have one thing in common.
    a) They uses the same system call
    b) They share the same kernel
    c) All UNIX flavour must use same shell
    d) All system call are based on underlying hardware structure.
13. Where does the system call build up?
    a) Kernel b) Shell c) Command interpreter d) Application program

Buffer Cache

Answer following in short

1. What do you mean by throughput of the system?
2. What do you mean by block in UNIX operating system?
3. What is the purpose of buffer cache?
4. Which are two parts of buffer?
5. Justify "The buffer is the in-memory copy of the disk block".
6. Why a disk block can never map into more than one buffer at a time?
7. How does kernel searches for buffer in the buffer cache?
8. What are the values for the status of buffer?
9. What do you mean by delayed write?
10. What do you mean by buffer status busy?
11. Name the mechanism used for managing free list of buffers.
12. To search effectively, how kernel manages buffers? (which data structure used to manage buffers so that search become effective?)
13. Name algorithm used for buffer allocation.
14. What is the purpose of brelse algorithm?
15. Give one point of difference between bread and breada algorithm.
16. List any two advantages of buffer cache.
17. List any one disadvantages of buffer cache.
18. Why kernel always allocate buffers from the head of the list?

Long Questions

1. Describe the structure of buffer pool.
2. How does free buffers are maintained by kernel in buffer cache?

3. Explain the scenarios of buffer retrieval in case of buffer found on the hash queue.
4. Explain the buffer retrieval scenarios in case of buffer not in hash.
5. Explain race condition in buffer retrieval.
6. Write a note on advantage and disadvantages of buffer cache.
7. Why size of buffer's data area must be at-least as big as size of disk block?
8. What is delayed write? Explain in detail.
9. Explain algorithm to release a buffer.
10. Explain breada algorithm. How it is differs from bread.

Multiple Choice Questions

1. To increase the response time and throughput, the kernel minimizes the frequency of disk access by keeping a pool of internal data buffer called
   a. Pooling
   b. Spooling
   c. Buffer cache
   d. Swapping
2. Buffer cache helps to
   a. Store data
   b. Improved read/write performance
   c. Allocate memory
   d. None of the above
3. Which of the following is not the field of buffer header?
   a. device number
   b. block number
   c. status
   d. size
4. How does free list of buffers are maintained?
   a. Using link list
   b. maintaining free list of array
   c. doubly link list
   d. doubly link list with least recently used mechanism
5. Which of the following is used to search buffer of disk block in buffer pool?
   a. device number and inode number
   b. block number and inode number
   c. device number and status
   d. device number and block number
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

File Systems

Short Questions

1. Which information is obtained by using inbuilt macros S_ISREG of stat data structure?
2. Which kernel data structure stores the information about permissions to access a file?
3. Define inode.
4. List the data structures and their relationship when a regular file is opened.
5. What is remember inode? Where does it located?
6. List the fields of super block.
7. How current byte offset is maintained by the file table?
8. What is the purpose of lseek() system call?
9. What is the return value of lseek() system call
10. How a user can move from beginning to end or in the middle of the file?
11. What is file descriptor?
12. What is the difference between user file descriptor table and file table?
13. Does each entry in the user file descriptor table have a separate file table associated with it?
14. List the information stored in inode block.
15. Why ialloc algorithm is required while creating a new file?
16. Which algorithm is used to free the inode used by the file?
17. Which fields are additional in in-core inode than (disk) inode?
18. Why does inode number start from 1 not from 0?
19. What is the purpose of link system call? Give syntax with parameter description.
20. What does directory entry consists of?
21. How kernel searches for free inodes?
22. Give any one situation for buffer status contains invalid data.
23. What do you mean by delay-write status of buffer?
24. Give one difference and one similarity between algorithm bread and breada.
25. What happens when inode is released and free list of inode in super block is full?
26. What do you mean by system call? How it is differs from C library function?
27. What is the return value and type of creat() system call?
28. Where does file name stored?

Long Questions

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

Multiple Choice Questions

1. 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);
2. Which of the following information does not stored in inode?
  a. File size
  b. File name
  c. File owner
  d. File group
3. 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
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

List of free inodes are maintained in _______.
  a. Boot block
  b. Super block
  c. I-node block
  d. Data block

10. Type field of i-node stored zero means _______.
  a. i-node is allocated
  b. i-node is free
  c. i-node is locked
  d. i-node is corrupted
11. 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
12. 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 cout
d. Reference count

13. 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

14. Use of dup(intfd) system call is _______ .
a. Create duplicate file
b. Copy a file descriptor into the first free slot of the user file descriptor table
c. Copy a file content
d. Copy a file table entry in file table maintained by kernel

15. Which of the following system call is used to creat a new file?
a. open()
b. creat()
c. both open() and creat() can be used
d. none of above

16. Which of the following system call is used to delete a file?
a. delete()
b. close()
c. unlink()
d. link()

17. A process may query status of a file by using ________ system call.
a. open()
b. read()
c. status()
d. stat()/fstat()

Process Management

Short Questions

1. Give significance of PID & PPID.
2. Mention the use of the fork().
3. How do you display all processes running on your system?
4. Name five important process attributes that are inherited by the child from its parent.
5. Kernel said to be non-preemptive. Note your comments on previous statement.
6. Define swapping & paging.
7. List any two advantages and disadvantages of kernel data structure.
8. Define pathname component.
9. Differentiate between user mode & kernel mode.
10. What are the 3 levels of UNIX operating system?
11. Define system call interface.
12. How will you kill the last background job without knowing its PID?
13. Define zombie state of process transition.
14. In which situation mode is used in open()?
15. Define signal.
16. What is context of process?
17. How you can avoid creating a process for command execution when it is to be executed from shell prompt?
18. Which programs are executed by spawning a shell? What does the second shell do?
19. Mention the use of lseeksystem call.
20. Write the I/O parameters saved in the U area.
21. Define init process.

Long Questions

1. Describe the state transition that a process undergoes during its lifetime .
2. 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.
3. Why kernel does not prempt process running in kernel mode in UNIX system?
4. List internal steps after execution of fork().
5. Discuss what happens to open files on creation of child process. Illustrate using program and by drawing.
6. State the purpose of using Wait(). Explain wait() with different example of getting status of child.
7. List steps performed after executing exit().
8. Which steps are performed in UNIX operating system after occurrence of interrupt while executing a process in kernel mode?
9. In user mode, a process can directly access hardware? If not, why? If yes, how?
10. Discuss booting sequence.
11. Differentiate between exceptions & interrupts.
12. Explain usage of exec() in detail.
13. 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?
14. Is there any process in the system whose PID is 0 (i.e. Zero)? Give significance of it.
15. Explain in detail about context of the process in UNIX system.

Multiple Choice Questions

1. What is a context switch?
   a. Kernel switches from executing one process to another.
   b. Process switches from kernel mode to user mode.
   c. Process switches from user mode to kernel mode.
   d. None of the above
2. Pid of init process is _______.
   a. 0
   b. 1
   c. 32767
   d. None of the above.
3. What is the default maximum number of processes that can exist in Linux?
   a. 32768
   b. 1024
   c. 4096
   d. Unlimited
4. How do you get parent process identification number?
   a. Waitpid
   b. getpid()
   c. getppid()
   d. parentid()
5. The process which terminates before the parent process exits becomes_______..
   a. Zombie
   b. Orphan
   c. Child
   d. None of the above
6. 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.
7. Fork returns _____ to parent process on success.
   a. 0

b. child process id

c. parent process id

d. none

8. 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

9. What is the output of the below code?

```
void exit_handler1();
void exit_handler2();
int main() {
intpid;
atexit(exit_handler1);
atexit(exit_handler2);
pid = fork();
if(pid == 0) {
_exit(0);
} else {
sleep(2);
exit(0);
}
return 0;
}
```

a. Only child executes the exit_handler 1 and 2.

b. Only parent executes the exit_handler 1 and 2.

c. Both parent and child executes the exit_handler 1 and 2.

d. Neither parent nor child executes the exit_handler 1 and 2.

10. What is output of the following program?

```
int main() {
fork();
fork();
fork();
if (wait(0) == -1)
printf(“leaf child\n”);
}
```

a. “leaf child” will be printed 1 times

b. “leaf child” will be printed 3 times

c. “leaf child” will be printed 4 times

d. “leaf child” will be printed 8 times

11. Which niceness value among the following indicates most favourable scheduling?

a. 0

b. 20

c. 5

d. -19

12. Nice can be used by an ordinary process to_______.

a. increase the priority of a process

b. decrease the priority of a process

c. increase or decrease the priority of a process

d. None of the above.

13. 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

14. 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 $?
15. 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
16. init run-level for system shutdown is_______.
    a. 3
    b. 2
    c. 1
    d. 0

Inter-process Communication

Short Questions

1. What is signal? How it can be sent to other process?
2. What happens to process on the receipt of the signal?
3. What do you mean by anonymous pipe?
4. Why pipe used only for half duplex communication?
5. What is return value of pipe() system call on success?
6. Why does pipe is available to use in related process?
7. Give name of any two signal which cannot be ignored or catch by a process.
8. What do you mean by installing signal handler?
9. What do you mean by re-installing signal handler?
10. When does process checks for arrival of signal?
11. What do you mean by transient pipe?
12. What is named pipe?
13. List System V IPC mechanism?
14. Which command is used to query status of System V IPC?
15. Which system call is used to remove message queue?
16. Which system call is used to access semaphore?
17. What will be return value of getshm() system call?
18. What is the significant of ftalk() call?
19. What if reader process closes the (all)read end of the pipe?

Long Questions

1. Which system call is used to send a signal? Give its syntax and parameter description. What if signal not sent?
2. What happens to process on the receipt of the signal? Explain.
3. Why does process will not react to signal on receipt of signal in case when it occurs more than one time? Discuss its solution.
4. How signal mechanism is used as communication mechanism between processes? Explain.
5. Which signal is sent to the parent process on child termination? Illustrate by giving example.
6. What do you mean by anonymous pipe? Why it called anonymous?
7. Why unnamed pipe used only in related processes?
8. Why pipe is used in half duplex communication? Explain by giving example.

9. What is FIFO? How it is differs from pipe (unnamed pipe)? How it can be created and accessed?
10. Write programs to demonstrate communication between two arbitrary process using FIFO.
11. Explain different scenario in case of inter processes communication done using FIFO.
12. How a parent process can wait for termination of its child process? Explain with an example.
13. How a child process can pass exit status to its parent process? Illustrate using program.
14. Explain system call wait(), waitpid().
15. Explain internal working of exit() system call.
16. Explain block and non block operation in msgrcv() and msgsnd() system call.
17. Describe the common characteristics of System V IPC mechanism.
18. How process can obtain shared memory region for communication? Also explain reading and writing into shared memory with example.
19. What operation can be performed on semaphore? Explain each with example.
20. How semaphore can be used to solve critical section problem?

Multiple Choice Questions

1. Which of the following system call is used to create named pipe?
   a. pipe
   b. open
   c. mknod
   d. fifo
2. Which system call is used by the process to send a signal to other process(es).
   a. signal
   b. msgsnd
   c. kill
   d. sndsignal
3. The kernel handles a signal only when process is in:
   a. User mode
   b. Kernel mode
   c. Both
   d. Sleeping mode
4. Process can ignore signal using:
   a. signal system call
   b. kill system call
   c. trap system call
   d. handle system call
5. Which is true regarding pipes?
   a. half duplex
   b. full duplex
   c. message boundaries are preserved
   d. Unordered data
6. On receipt of signal process can do which of the following action. (Tick all answer)
   a. Exits
   b. Ignore
   c. Executes function
   d. Sends signal
7. Which system call is used to get process id of last process which sends last message on message queue?
   a. msgsnd
   b. msgrcv
   c. msgget
   d. msgctl

8. Which system call is used to remove message queue structure from the kernel?
   a. msgsnd
   b. msgrcv
   c. msgget
   d. msgctl
9. Which system call is used to read data from the shared memory?
   a. read
   b. shmat
   c. shmdt
   d. None of these
10. What information is stored in semaphore structure for each semaphore?
    a. The PID of process that performed last operation
    b. The current value of semaphore
    c. Number of process waiting for resource to become available
    d. All of above
11. Which of the following is not an element of sembuf structure?
    a. The number of the semaphore you wish to deal with
    b. Process Id of current process
    c. The operation to perform (positive, negative, or zero)
    d. Operational flags
12. The system call semctl() can be used to do following
    a. To retrieve shared memory structure
    b. To perform various command on shared memory
    c. To create semaphore
    d. To remove semaphore set from the system
13. The persistency of a FIFO is
    a. process
    b. kernel
    c. file system
    d. none
14. Advantage of FIFO over pipe is
    a. related processes can communicate
    b. unrelated processes can communicate
    c. none
15. What mkfifo() creats?
    a. pipe
    b. unnamed pipe
    c. named pipe
    d. msg queue
16. System V IPC common attributes are
    a. key
    b. id
    c. owner
    d. all the above
17. Which one of the following is not system V IPC ?
    a. Shared Memory
    b. Semaphores
    c. FIFO
    d. Message Queues
18. Which system call is used to create Sys V message Queue.
    a. msgget

b. shemget
c. semget
d. msgctl

19. Which is not the correct option for removing a message queue
    a. ipcrm -Q
    b. ipcrm -q
    c. ipcrm -m
    d. None
20. Message queues are created in
    a. userspace
    b. kernelspace
    c. both
    d. none
21. Command used to check shared memory is
    a. ipcs
    b. ipcs -m
    c. ipcs -s
    d. ipcs -q
22. Which is Fastest IPC?
    a. Message Queue
    b. shared memory
    c. Socket
    d. all
23. What is the persistancy level of Shared memory segments?
    a. signal
    b. process
    c. file system
    d. kernel
24. The structure which keeps the information about shared memory in the kernel is
    a. struct ipc_perm
    b. struct semid_ds
    c. struct shmid_ds
    d. struct msgid_ds
25. One process requires M resource to complete a job. What should be the minimum number of resources available for N processes so that at least one process can continue to execute without blocking/waiting?
    a. M * N
    b. M * N – 1
    c. M * N + 1
    d. M
26. Semaphore P( ) operation usually does the following:
    a. descrements the semaphore count and the process sleeps if needed
    b. increments the semaphore count
    c. wakes up a sleeping process
27. Which call to use to set the resource count of semaphore?
    a. semget( )
    b. semctl( )
    c. sem_setcount( )
    d. sem_set_count( )
28. Which of the following IPC is fastest amongst all?
    a. Message queue
    b. Semaphore

   c. Shared memory
   d. All are equal

Input /Output Management

Short Questions

1. What is a device file?
2. What do you mean by device driver?
3. What do you mean by major number?
4. What do you mean by minor number?
5. What are software devices? Give one example.
6. Which are two types of devices in UNIX system? Give two examples of each.
7. Which are two modes of line discipline?
8. What do you mean by canonical mode of line discipline module?
9. What is Cblock?

Long Questions

1. List three stages of device configuration.
2. Write a note on terminal driver.
3. What are the functions of line discipline?
4. What is Clists? What are its functions?
5. Write note on Cblock/Clist.
6. Write a note on terminal driver in raw mode and terminal driver in canonical mode.
7. Explain system call ioctl().

Multiple Choice Questions

1. Which system call is used to create device file?
   a. creat
   b. open
   c. mknod
   d. pipe
2. Which of the following is used to represent device driver or device type?
   a. No of link
   b. Major number
   c. Minor number
   d. None of above
3. Which of the following device parameter represent instances of device?
   a. Number of link
   b. Major number
   c. Minor number
   d. None of above
4. Which of the following is function of line discipline:
   a. To parse input strings into lines.
   b. To process erase characters.
   c. To echo received characters to the terminal.
   d. All of above.