

Unit 12 - Week 11

Course outline

How does an NPTEL online course work?

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Week 9

Week 10

Week 11

● Storage Devices, Filesystem Interfaces

● File System Implementation

● File System Operation

○ Quiz : Assignment 11

○ Week 11 Feedback Form

Week 12

Assignment Solution

Download Videos

Text Transcripts

Assignment 11

The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.

Due on 2020-04-15, 23:59 IST.

1) Consider a magnetic disk doing 3600 RPM with average seek time of 12 ms and supporting transfer rate of 3.5 MB/s. Also Each sector is 512 bytes and disk controller overhead for each I/O operation is 5.5 ms. The average time to read a single sector in milliseconds is ____

Hint

No, the answer is incorrect.
Score: 0

Accepted Answers:
(Type: Numeric) 26

1 point

2) Which of the following secondary storage technology does not support random accesses?

1 point

- Magnetic disk
- Tape
- Flash disk
- None of the above

No, the answer is incorrect.
Score: 0

Accepted Answers:
Tape

3) Which of the following is represented by an inode?

1 point

- Files
- Directories
- Pipes
- Superblock
- Boot Sector

No, the answer is incorrect.
Score: 0

Accepted Answers:
Files
Directories

4) There is well known variant of SCAN disk scheduling algorithm known as C-SCAN in which the head starts from one end, completes all the request on its way to the other end and immediately returns to the starting position without serving request in return journey. How does C-SCAN compares with SCAN?

1 point

- Fairer waiting time with same overall throughput
- Fairer waiting time with better overall throughput
- Same waiting time with better overall throughput
- Same waiting time with same overall throughput

No, the answer is incorrect.
Score: 0

Accepted Answers:
Fairer waiting time with same overall throughput

5) Which of the following are true for disk fragmentation?

1 point

- Contiguous allocation strategy suffers from external fragmentation
- Linked-list allocation strategy suffers from external fragmentation
- On increasing block size the maximum size of the file that can be stored increases
- On increasing block size the internal fragmentation increases

No, the answer is incorrect.
Score: 0

Accepted Answers:
Contiguous allocation strategy suffers from external fragmentation
On increasing block size the maximum size of the file that can be stored increases
On increasing block size the internal fragmentation increases

6) Which of the following are true for linked list allocation?

1 point

- There is no limit on maximum size of the file apart from constraints due to disk size.
- Sequential block accesses of a file are slow
- Accessing nth block of the file incurs O(n) time
- In case of FAT, accessing nth block of the file takes O(1) time
- FAT limits the data loss to the corrupt sector
- Subsequent data block accesses of a file can be speedup by keeping FAT in memory and harnessing better disk performance due to large number of outstanding I/Os submitted to controller

No, the answer is incorrect.
Score: 0

Accepted Answers:
There is no limit on maximum size of the file apart from constraints due to disk size.
Sequential block accesses of a file are slow
Accessing nth block of the file incurs O(n) time
In case of FAT, accessing nth block of the file takes O(1) time
FAT limits the data loss to the corrupt sector
Subsequent data block accesses of a file can be speedup by keeping FAT in memory and harnessing better disk performance due to large number of outstanding I/Os submitted to controller

7) Consider an indexed file allocation where inode 7 direct, 1 singly indirect, 1 doubly indirect and 1 triply indirect data block pointers. Also consider the block size to be 512 bytes and that each pointer in all these index files takes 4 bytes. What is the maximum size of the file in bytes supported by this filesystem?

No, the answer is incorrect.
Score: 0

Accepted Answers:
(Type: Numeric) 108219952

1 point

8) In case of disk cache write back policy is better that write through policy. Which of the following statements are true for write back policy?

1 point

- The better performance can be attributed to the better amortised performance of disk when it has multiple outstanding I/O requests
- It saves all but last write from taking place, in case of multiple writes to the same block as disk accesses are orders of magnitude slower than memory
- It is always efficient to use write back policy even in case of multiple systems connected to the disk

No, the answer is incorrect.
Score: 0

Accepted Answers:
The better performance can be attributed to the better amortised performance of disk when it has multiple outstanding I/O requests
It saves all but last write from taking place, in case of multiple writes to the same block as disk accesses are orders of magnitude slower than memory