CS 537: Introduction to Operating Systems
Fall 2024: Final

This exam is closed book, but you may use 1 sheet of notes.

No calculators may be used. All cell phones must be turned off and put away.

You have 1 hour and 30 minutes to complete this exam.

Write all of your answers on the accu-scan form with a #2 pencil:

- CANVAS LAST NAME - fill in your last (family) name starting at the leftmost column
- CANVAS FIRST NAME - fill in the first five letters of your first (given) name
- IDENTIFICATION NUMBER - This is your UW Student WisCard ID number
- ABC of SPECIAL CODES - Write your lecture number as a three digit value:
  - 001 – TuTh 9:30-10:45 (Louis)
  - 002 – TuTh 11:00-12:15 (Shivaram)

These exam questions, your sheet of notes, and the scantron must be returned at the end of the exam, but we will not grade anything in this booklet. You will not be getting your note sheet back.

If any question is ambiguous or confusing, use the words on the page to answer the question. You may also bring it to the attention of a proctor, but the proctors are instructed to not give out guidance on answering questions. We may make clarifying announcements to the room as a whole. If you find any error in a question, bring it to the attention of a proctor.

The following might help you with some calculations:

- 0x100 = 2^8 = 256
- 2^{10} = 1024
- 2^{12} = 4096
- 2^{10} bytes = 1KB
- 2^{20} bytes = 1MB

This exam has 54 questions. Each question has the same number of points.

Good luck!

True (A) / False (B)

1. Direct Memory Access (DMA) orchestrates the transfer of data between main memory and devices without much CPU intervention.
2. One advantage of increasing a disk's rotational speed is reducing the seek time of I/O operations.
3. One of the advantages of the SSTF approach to disk scheduling is that it avoids starvation.
4. One reason the FTLs of most SSDs are log-structured is to reduce write amplification.
5. Overall, SSDs with TLC cells achieve higher performance and are more expensive than SSDs with SLC cells.
6. One danger of hard links is dangling references to files.
7. AFS uses access control lists per directory.
8. In an FFS-like filesystem, multiple inodes may point to the same file data.
9. To successfully perform the operation `open("/foo/bar")` in the Very Simple FS requires 5 blocks to be read (assuming each inode is in its own block and each directory is in a single block).
10. In the FFS filesystem, the system attempts to spread files within the same directory across different groups.
11. In data journaling, a transaction consists of a begin block, blocks for all updates to the FS, and an end block.
12. After a crash, FSCK fixes the on-disk state of the filesystem to match its state before the last update to disk began.
13. When appending a single new block onto the end of a file, a crash occurs after the data block has been written but before updating the file's inode or the data bitmap. This leaves the FS in an inconsistent state.
14. In journaling, to make sure the write of a transaction end is atomic, one should make it a single 512-byte block.
15. Checkpointing is the process of creating a journal transaction before making any changes to the rest of the file system.
16. The motivation for developing LFS was that prior filesystems like FFS performed only sequential access to disk drives.
17. All the blocks in a given segment will be garbage or live in LFS.
18. LFS uses two checkpoint regions for redundancy in case of a crash during an update to one of the regions.
19. Virtual machines are useful for consolidating multiple OSes onto the same hardware.
20. Using para-virtualization can slow down memory translation for a guest OS.
21. The Popek Goldberg theorem says that a VMM can be constructed if all the privileged instructions are sensitive.
22. Hardware that meets the Popek Goldberg criteria can use the trap-and-emulate model for virtualization.
23. Using a single queue for multiprocessor scheduling avoids locking overheads.

24. If there are two processes, P1 with nice value -10 and P2 with nice value +10, then P1 is given a higher priority while scheduling.
25. One way to solve the load imbalance problem in MQMS is work stealing.
26. Linux CFS picks the process with the smallest `vruntime` when scheduling.
27. When the ULE scheduler is selecting the next thread to run, it alternates between selecting from the interactive queue and the batch queue.
28. Suppose that in a TCP communication that uses sequence numbers, the last message that the receiver process has seen has a sequence number of K. The next message that it receives can have a sequence number of K + 2.
29. In the RAID1v mode of the P6 filesystem, if copies of a data block are different, then the data block of the disk with the highest index should be returned.
30. The purpose of modifying the original struct for P6 filesystem's superblock is to incorporate pointers to the different filesystem data structures (e.g. bitmap pointer, inode pointer, etc.).

Multiple Choice

Devices

31. What techniques are employed to lower CPU overhead for I/O operations?

A. Interrupt service routines
B. Privileged I/O instructions
C. Direct Memory Access
D. CPU directed I/O instructions
E. Both A and C

32. Suppose a disk has 36 tracks numbered from 0 to 35. Currently the disk arm is at track 20, and there is a queue of disk access requests for tracks 2, 4, 14, 20, 22, 23, 24 and 35. If Shortest Seek Time First (SSTF) disk scheduling policy is being used, the request for track 14 is serviced after servicing how many requests?

A. 1
B. 4
C. 6
D. 2
E. None of the above

33. What will be the expected total I/O time for a sequential read of a 5 MB file from a disk with the following characteristics:

    Capacity: 1TB
    RPM: 10,000
    Average Seek Time: 5ms
    Max Transfer Rate: 100 MB/s
    Single read size: 4KB

A. 61 ms
B. 50 ms
C. 58 ms
D. 55 ms
E. None of the above

SSDs

You use a log-based SSD to hold your data. The SSD has 3 blocks with 10 pages per block. Each page holds a single character. The SSD has been used for a while and the current state of the SSD is shown below. The diagram for the SSD contains the following items:

FTL mapping logical pages to physical pages
block number
page number (2 lines)
state of each page (valid, Erased, or invalid)
data stored at each page
an indicator(+) if a page is currently live (has an entry in the FTL)

Current State of SSD:

```
FTL     1:  6   2: 10   5:  2   7:  1
        8: 20   9:  4  12: 16  13:  9
       15: 12  17: 17  18:  5  29: 13
       19: 11  20: 14  21: 15  23:  3
       24: 18  25:  0  27:  8  28:  7

Block 0          1          2
Page  0000000000 1111111111 2222222222
      0123456789 0123456789 0123456789
State vvvvvvvvvv vvvvvvvvvv vEEEEEEEEE
Data  Ih7tj3UYbK 34quFTEPXg H
Live  ++++++++++ +++++++++  +
```

Operations that the OS may perform are:

write(Logical Address, Data) – storing the data at the logical address
read(Logical Address) – reading the data from the logical address
trim(Logical Address) – deleting the data from the logical address

34. What will be returned by the OS operation read(24)?
A. Error
B. 18
C. F
D. X
E. 4

35. What single OS operation would result in the state of the SSD changing to the following:

```
FTL      1:  6   2: 10   5:  2   7:  1
         8: 20   9:  4  12: 16  13:  9
        15: 21  17: 17  18:  5  29: 13
        19: 11  20: 14  21: 15  23:  3
        24: 18  25:  0  27:  8  28:  7

Block 0          1          2
Page  0000000000 1111111111 2222222222
      0123456789 0123456789 0123456789
State vvvvvvvvvv vvvvvvvvvv vvEEEEEEEE
Data  Ih7tj3UYbK 34quFTEPXg Hg
Live  ++++++++++ ++ ++++++  ++
```

A. trim(15)
B. write(19, g)
C. write(21, g)
D. write(15, g)
E. None of the above

Simple File Systems and FS API

36. A difference between hard links and symbolic links is that

A. Symbolic links cannot be made to directories
B. Hard links cannot be made to directories
C. Hard links cannot be made across file systems
D. Symbolic links increase the reference count in the inode
E. Both B and C

37. After the following commands are executed by `user` inside the empty directory `/foo/bar`, what will the `ls -al` command output? Assume `user` has full access to the directory. Note that the user, group, file size, and datetime have been omitted.

```
echo “hello” > foo.txt
chmod 764 foo.txt
ln -s foo.txt ../bar/bar.txt
mkdir bar
ln foo.txt /foo/bar/bar/bar.txt
mv foo.txt foo2.txt
rm bar.txt
rmdir bar
unlink bar
```

```
A. drwxrwxr-x 3 .
   drwxrwxr-x 3 ..
   drwxrwxr-x 2 bar
   -rwxrw-r-- 2 foo2.txt

B. drwxrwxr-x 3 .
   drwxrwxr-x 3 ..
   drwxrwxr-x 2 bar
   -rwxrw-r-- 3 foo2.txt

C. drwxrwxr-x 3 .
   drwxrwxr-x 3 ..
   drwxrwxr-x 2 bar
   -rw-rw-rw- 2 foo2.txt

D. drwxrwxr-x 3 .
   drwxrwxr-x 3 ..
   -rwxrw-r-- 2 foo2.txt

E. None of the above
```

In a simplified file system the following operations are implemented:

mkdir()	- creates a directory
creat()	- creates an empty file
open(), write(), close()	- appends a block to a file
link()	- creates a hard link to a file
unlink()	- unlinks a file

The on-disk file system is represented by the following:

inode bitmap
inode table (each inode surrounded by square brackets)
data bitmap
data blocks (each block surrounded by square brackets)

The inode structure contains the type (d - directory, f - regular file), a pointer to a single data block (a), and a reference count (r). A directory data block contains (filename,inode number) pairs.

The current state of the file system is as follows:

```
inode bitmap  11111100
inodes [d a:0 r:4][d a:1 r:2][f a:-1 r:1][f a:-1 r:2][d a:2
r:2]
[f a:-1 r:1][][]
data bitmap   11100000
data [(.,0) (..,0) (g,1) (q,2) (u,3) (x,3) (t,4)][(.,1)
(..,0)(c,5)] [(.,4) (..,0)][][][][][]
```

38. What operation will result in the file system state changing to the following:

```
inode bitmap  11111100
inodes [d a:0 r:4][d a:1 r:2][f a:-1 r:1][f a:-1 r:1][d a:2
r:2]
[f a:-1 r:1][][]
data bitmap   11100000
data [(.,0) (..,0) (g,1) (q,2) (u,3) (t,4)][(.,1) (..,0) (c,5)]
[(.,4) (..,0)][][][][][]
```

A. rm("/x")
B. creat("/u")
C. mkdir("/g/c")
D. unlink("/x")
E. None of the above

39. What is the inconsistency in this file system?

```
inode bitmap 1000011000010001
inodes       [d a:0 r:3] [] [] [] [] [f a:13 r:1] [d a:6 r:2]
[] [] [] [] [f a:4 r:2] [] [] [] []
data bitmap  1000101000000100
data         [(.,0) (..,0) (o,6) (m,5) (h,11) (f,11)] [] [] []
[k] [] [(.,6) (..,0)] [] [] [] [] [] [] [x] [] []
```

A. Inode 11 type is incorrect
B. Inode 6 directory entries inconsistent
C. Inode bitmap corrupt bit 15
D. Data bitmap corrupt bit 4
E. No inconsistency

40. What is the inconsistency in this file system?

```
inode bitmap 1011000000000001
inodes       [d a:0 r:3] [] [d a:13 r:3] [d a:9 r:2] [] [] []
[] [] [] [] [] [] [] [] [f a:15 r:3]
data bitmap  1000000001000101
data         [(.,0) (..,0) (f,15) (x,15) (r,2)] [] [] [] [] []
[] [] [] [(.,3) (..,2)] [] [] [] [(.,2) (..,0) (s,3)] [] [w]
```

A. Inode 15 reference count incorrect
B. Inode 2 directory entries inconsistent
C. Inode bitmap corrupt bit 2
D. Data bitmap corrupt bit 9
E. No inconsistency

41. What is the inconsistency in this file system?

```
inode bitmap 1000100010010000
inodes       [d a:0 r:3] [] [] [] [d a:1 r:2] [] [] [] [d a:8
r:3] [] [] [d a:2 r:2] [] [] [] []
data bitmap  1110000010000000
data         [(.,0) (..,0) (v,4) (m,4) (o,8)] [z] [(.,11)
(..,8)] [] [] [] [] [] [(.,8) (..,0) (f,11)] [] [] [] [] [] []
[]
```

A. Inode 8 reference count incorrect
B. Inode 4 type is incorrect
C. Inode bitmap corrupt bit 4
D. Data bitmap corrupt bit 7
E. No inconsistency

Journaling

Imagine you have an FFS-like file system that is creating a new empty file in an existing directory and must update 4 blocks: the directory inode, the directory data block, the file inode, and the inode bitmap. Assume the directory inode and the file inode are in different on-disk blocks. Assume we've added a basic implementation of full-data journaling to our FFS-like file system and perform the same file create operation as above. Assume a transaction header block and a transaction commit block. Assume each block is written synchronously (i.e., a barrier is performed after every write and blocks are pushed out of the disk cache). If the system crashes after the following number of blocks have been synchronously written to disk, what will happen after the system reboots? (If the number of disk writes exceeds those needed, assume they are unrelated.) What happens if a crash occurs after only updating the following block(s)?

42. 5 disk writes
A. No transactions replayed during recovery; file system in old state
B. No transactions replayed during recovery; file system in new state
C. Transaction replayed during recovery; file system in old state
D. Transaction replayed during recovery; file system in new state
E. Transaction replayed during recovery; file system in unknown state

43. 6 disk writes
A. No transactions replayed during recovery; file system in old state
B. No transactions replayed during recovery; file system in new state
C. Transaction replayed during recovery; file system in old state
D. Transaction replayed during recovery; file system in new state
E. Transaction replayed during recovery; file system in unknown state

Log Structured File Systems

You are given a stream of writes that appear on a disk that runs a Log Structured File System. However some of the entries from the segment are now missing and you are asked to investigate what those missing entries might be.

Assumptions: You can also assume that a single inode takes up an entire block (for simplicity). The LFS inode map is called the “imap” below and of course is also updated as needed. The inodes in this filesystem contain:

(a) size field which counts the number of blocks present in the file/directory
(b) ptr field which contains a list of pointers to data blocks
(c) type field which is ‘d’ for directories and ‘f’ for regular files

44. You see a segment with the following set of writes

```
block 100: [("." 0), (".." 0), ("foo" 1)]
block 101: [size=1,ptr=100,type=d]
block 102: [size=0,ptr=-,type=f]
block 103: [imap: 0->XXX,1->102]
```

What could be the contents of XXX in the above segment?

A. 100
B. 101
C. 102
D. 103
E. None of the above

45. You see a segment with the following set of writes

```
block 100: [("." 0), (".." 0), ("foo" 1)]
block 101: [size=1,ptr=100,type=d]
block 102: [(“.” 1), (“..” 0)]
block 103: [size=1,ptr=YYY,type=d]
block 104: [imap: 0->XXX,1->103]
```

What could be the contents of YYY in the above segment?

A. 100
B. 101
C. 102
D. 103
E. None of the above

46. You see a segment with the following set of writes

```
block 104: [SOME DATA]
block 105: [SOME DATA]
block 106: [size=2,ptr=104,ptr=105,type=f]
block 107: [imap: 0->101,1->106]
block 108: [SOME DATA]
block 109: [SOME DATA]
block 110: [size=4,ptr=104,ptr=105,ptr=108,ptr=109,type=f]
block 111: [imap: 0->101,1->110]
```

How many blocks contain garbage data after the following set of writes?

A. 1
B. 2
C. 3
D. 4
E. 5

47. You see a segment with the following set of writes

```
block 104: [SOME DATA]
block 105: [SOME DATA]
block 106: [size=2,ptr=104,ptr=105,type=f]
block 107: [imap: 0->101,1->106]
block 108: [SOME DATA]
block 109: [SOME DATA]
block 110: [size=2,ptr=108,ptr=109,type=f]
block 111: [imap: 0->101,1->110]
```

How many blocks contain garbage data after the following set of writes?

A. 1
B. 2
C. 3
D. 4
E. 5

48. Assuming we start with no state in memory, and have a LFS with that looks like the following: how many disk I/Os do we need to perform to read the contents of the directory “/”?

```
block 0: Checkpoint region: imap[0..2] -> 3
block 1: [("." 0), (".." 0)]
block 2: [size=1,ptr=1,type=d]
block 3: [imap: 0->2]
```

A. 1
B. 2
C. 3
D. 4
E. 5

49. Assuming we have the imap cached in memory and have a LFS with that looks like the following: how many disk I/Os do we need to perform to read the contents of the directory “/”?

```
block 0: Checkpoint region: imap[0..2] -> 3
block 1: [("." 0), (".." 0)]
block 2: [size=1,ptr=1,type=d]
block 3: [imap: 0->2]
```

A. 1
B. 2
C. 3
D. 4
E. 5

Advanced Topics and Project 6

50. Which of the following system-level virtual machines does not have an underlying host operating system?

A. LIghtweight VM
B. Machine Simulator
C. Type-1 Hypervisor
D. Type-2 Hypervisor
E. None of the above

51. Which of the following schedulers are NOT examples of MQMS?

A. O(1)
B. CFS
C. ULE
D. BFS
E. None of the above

52. What of the following may happen if we set the TCP timeout period at a very small value (e.g. 1 millisecond) ?

A. Wasted CPU cycles on the sender side
B. Messages will be received out of order
C. Application performance will always improve
D. Sender might waits too long to retry messages
E. None of the above

53. Differences in the P6 filesystem and FFS from the textbook and class include

A. Inodes are not packed close to each other
B. New files are initially empty (i.e. no data blocks assigned)
C. Reads may span multiple data blocks
D. Deleting a file does not include freeing its inode
E. None of the above

54. Which of the P6 RAID modes could increase write performance over a non-RAID filesystem by using concurrency?

A. RAID 0
B. RAID 1
C. RAID 1v
D. Both B and C
E. None of the above

This is the end of the exam. Thanks for taking Operating Systems.