CS 111 Final exam

CHU; JONATHAN WEI-HONG

TOTAL POINTS

130 / 150

QUESTION 1

1 Scatter/gather I/O 7 / 10

- **0 pts** Correct
- **10 pts** No answer

✓ **- 3 pts** **Not identifying DMA**

- **3 pts** Not identifying non-contiguity of virtual RAM pages
- **2 pts** not identifying data copying as main issue
- **2 pts** Memory mapped I/O is not a motivation
- **2 pts** Not about accumulating I/O operations.
- **2 pts** Files and inodes not relevant.
- **10 pts** Totally wrong
- **2 pts** Scattering and gathering is over RAM, not I/O device.
- **2 pts** Not related to TLB misses.
- **1 pts** Segments are not necessarily contiguous in physical memory, either.
- **2 pts** Memory mapped I/O != paged virtual memory
- **1 pts** Which mechanisms of a VM system?
- **8 pts** DMA and the paging aspect of VM lead to problems without scatter/gather.
- **2 pts** File system issues irrelevant.
- **4 pts** Scatter/gather typically unrelated to demand paging.
- **2 pts** DMA requires physically contiguous memory.
- **3 pts** Defragmentation has nothing to do with scatter/gather.
- **2 pts** Swapping not relevant.
- **2 pts** Double buffering is irrelevant.
- **3 pts** Poor explanation.
- **2 pts** Fragmentation is not directly related to this issue.
- **9 pts** One tiny bit of correct information
- **1 pts** Internal device memory not relevant.

QUESTION 2

2 Metadata journaling 10 / 10

✓ **- 0 pts** **Correct**

- **10 pts** No answer
- **3 pts** Didn't provide enough discussion about what could happen if we write data blocks after metadata/journal is modified.
- **7 pts** Not very correct.

QUESTION 3

3 URLs and links 10 / 10

✓ **- 0 pts** **Correct**

- **10 pts** No answer
- **4 pts** A URL is more like a soft (symbolic) link
- **3 pts** In both cases, the link is a name describing a traversal through a set of linked data items - files and directories in the case of a soft
link, web pages in the case of a URL.
- **3 pts** There is no guarantee in either case that the data
item named by the URL or soft link actually exists.
- **10 pts** wrong answer
- **1 pts** mixed the concept of domain and URL
- **1 pts** do not explain how a URL works

QUESTION 4

4 Password salting 10 / 10

✓ **- 0 pts** **Correct**

- **10 pts** No answer
- **3 pts** Did not correctly explain in detail the definition of salt
- **4 pts** Did not correctly discuss in detail preserving password secrecy in the context of hashes
- **3 pts** Did not correctly explain dictionary attacks / brute force attacks

QUESTION 5

5 Factors 10 / 10

✓ **- 0 pts Correct**

- 10 pts No answer

- 5 pts A factor is an aspect of the system that you intentionally alter in controlled ways during the evaluation.

- 5 pts Proper choice of factors will allow the experimenter to gain insight into the likely performance outcome of design choices and varying use cases

- 1 pts The reason is not clearly or correctly explained

- 10 pts wrong answer

- 2 pts not proper answer "why"

- 3 pts It's the variables we alter

QUESTION 6

6 File descriptors and capabilities 10 / 10

✓ **- 0 pts Correct**

- 10 pts No answer

- 1 pts OS can easily revoke a file descriptor by removing it from the process control block.

- 3 pts Uniqueness not really a property of either capabilities or file descriptors. Important point is that possession grants access.

- 2 pts Important point is mere possession of each grants access.

- 2 pts Capabilities do not necessarily have any "position" information associated.

- 1 pts Users can also access files by opening them via ACL, so FDs alone don't specify their possible available files.

- 7 pts Both capabilities and file descriptors are about access control, not identification and/or authentication.

- 2 pts Changing the ACL does not invalidate existing file descriptors.

- 2 pts File descriptors are R/W specific.

- 3 pts File descriptors tell us nothing about why someone could access a file, merely that they can.

- 8 pts Insufficient detail.

- 5 pts Important point is that both are access control mechanisms providing security based on mere possession of a data structure.

- 1 pts Capabilities usually do not contain a list. Rather, you have a list of capabilities.

- 7 pts How is a FD like a capability?

- 5 pts Misdefinition of capabilities.

QUESTION 7

7 Dining philosophers 10 / 10

✓ **- 0 pts Correct**

- 10 pts No answer

- 9 pts Wrong answer.

- 3 pts Needs a better explanation. A good example is when all philosophers call getforks() at the same time and all of them get the left fork.

- 3 pts Partial correct.

QUESTION 8

8 Monitors and synchronized methods 4 / 10

- 0 pts Correct

- 10 pts No answer

- 4 pts More detail on granularity.

- 2 pts All synchronized methods in an object share one lock.

- 2 pts OO monitors provided by language, not OS.

- 6 pts Monitors lock entire object for any method, synchronized methods only lock on specified methods.

✓ **- 6 pts Sync methods more fine grained than object monitors, since the latter locks object on ANY method.**

- 10 pts Totally wrong.

- 3 pts Monitors do not prevent inter-object deadlocks.

- 2 pts Monitors lock a class instance, not an entire class.

- 1 pts Java sync methods require identification of the methods. They don't try to determine if the object is modified.

- 3 pts With synchronized methods, non-synchronized methods can be used in parallel.

- **1 pts** Java synchronized methods provide enforced locking.

- **3 pts** Object oriented monitors are often provided in the language, and need not be implemented by the programmer.

QUESTION 9

9 Callbacks in AFSv2 0 / 10

- **0 pts** Correct

- **10 pts** No answer

- **2 pts** Callbacks occur when a file is updated, not to check if the cached copy is still OK.

✓ **- 10 pts Not the purpose of an AFS v3 callback. It's for cache consistency.**

- **5 pts** Callbacks go from server to caching clients when a file is updated.

- **8 pts** More detail required.

- **10 pts** AFS is a file system.

- **5 pts** Callback is to notify caching client of updates at other sites, not to validate that data has been received.

- **5 pts** Why does this have to happen?

- **2 pts** Not just for directories.

- **2 pts** Why would a file's status change without the client knowing about it?

QUESTION 10

10 PK certificates 10 / 10

✓ **- 0 pts Correct**

- **10 pts** No answer

- **2 pts** Did not mention public key of issuer in certificate.

- **2 pts** Did not mention digital signature of trusted 3rd party in certificate

- **2 pts** Did not say that a mutually trusted third party is needed to sign the digital signature

- **4 pts** Did not correctly say that the trusted 3rd party's public key, which matches the 3rd party's private key used to sign the digital signature, is needed to decrypt the digital signature

QUESTION 11

11 Zombie states 10 / 10

✓ **- 0 pts Correct**

- **10 pts** No answer

- **5 pts** A final state indicates that a process has finished executing all of its code. However, it has not yet been cleaned up.

- **5 pts** It allows the parent process to check its exit status and possibly perform other cleanup tasks.

- **10 pts** wrong answer

- **2 pts** all of the memory and resources associated with a zombie process are deallocated

- **2 pts** The parent process checks the exit status

- **5 pts** Parent process waits for child process

QUESTION 12

12 Fairness and scheduling 10 / 10

✓ **- 0 pts Correct**

- **10 pts** No answer

- **5 pts** Performance is a vague term. What precisely do you mean? Your example is unclear.

- **1 pts** Precisely what do you mean by performance here? Fairness itself is one aspect of performance.

- **10 pts** That's not a property.

- **5 pts** Why is continuity desirable?

- **2 pts** Even a fair scheduler would not insist on a blocked process getting an equal time slice.

- **2 pts** Need better description of why.

- **3 pts** Fairness and preemption aren't the same thing. Unfair algorithms can also use preemption.

- **1 pts** You're talking about turnaround time, not response time.

- **2 pts** Your description does not say why throughput is damaged.

- **2 pts** Disk latency not really relevant here.

- **2 pts** That's not throughput. Throughput is the amount of useful work completed in a unit time. You're talking about turnaround time.

QUESTION 13

13 Free list ordering 10 / 10

✓ **- 0 pts Correct**

- **10 pts** No answer

- **8 pts** Incorrect understanding of memory free list.
- **2 pts** Missing details or not a very good explanation for ordering by size.
- **2 pts** Missing details or not a very good explanation for ordering by address.
- **4 pts** Wrong answer for ordering by size.
- **4 pts** Wrong answer for ordering by address.

QUESTION 14

14 Page replacement for looping sequential workloads 10 / 10

✓ - **0 pts** **Correct**

- **10 pts** No answer
- **3 pts** More specifics on the alternate algorithm.
- **4 pts** Clock algorithms approximate LRU, so they aren't likely to do well.
- **1 pts** How could we know this?
- **5 pts** What other algorithm to use?
- **2 pts** How to practically implement your chosen algorithm?
- **3 pts** How will you do lookahead at the end of the loop area? How can you know?
- **1 pts** How to practically order the pages?
- **3 pts** How to choose which chunks to replace? Bad if you choose the LRU chunks.
- **2 pts** How do you know when you've reached the end of the loop and need to move to the head?
- **5 pts** Problem is vast number of page misses.
- **5 pts** This algorithm is no better than LRU, since it guarantees maximum paging.
- **3 pts** Why would you see constant page replacement?
- **3 pts** Which pages do you designate for swapping?

QUESTION 15

15 Load and stress testing 9 / 10

✓ - **0 pts** **Correct**

- **10 pts** No answer
- **4 pts** Did not say that load testing measures system performance under particular loads, usually loads that are expected to occur in actual operation
- **4 pts** Did not say that stress testing is used to understand how a system will perform in unusual circumstances.
- **2 pts** Did not mention that stress testing is most likely to be used in systems that cannot afford to fail.

- 1 Point adjustment

"have to run under less predictable outcomes" is too vague.

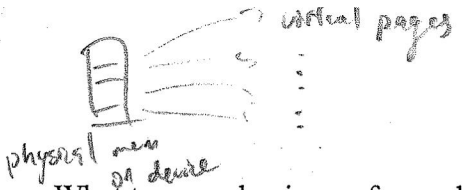
Final Exam
CS 111, Principles of Operating Systems
Winter 2018

Name: Jonathan Chu

Student ID Number: 004832220

This is a closed book, closed note test. Answer all questions.

Each question should be answered in 2-5 sentences. DO NOT simply write everything you remember about the topic of the question. Answer the question that was asked. Extraneous information not related to the answer to the question will not improve your grade and may make it difficult to determine if the pertinent part of your answer is correct. Confine your answers to the space directly below each question. Only text in this space will be graded. No question requires a longer answer than the space provided.



VM, paging

1. What two mechanisms of a modern memory management system lead to the need for scatter/gather I/O? Why do they do so?

Virtual memory and paging lead to the need for scatter/gather I/O between devices and physical memory.

Virtual memory requires the OS to translate contiguous virtual addresses to physical addresses, so data transferred between the device and virtual memory could be contiguous, but it becomes disjointed in physical memory. Paging causes the "scatter/gather" - contiguous virtual addresses are not necessarily contiguous in physical memory. This means that the OS will likely need to take the data from the device and scatter it to different physical pages when writing from device to memory, or vice versa it may need to gather data from multiple physical pages.

2. For a journaling file system that only puts metadata in the journal, the data blocks must be written to the storage device before the journal is written to that device. The process requesting the write is informed of its success once the journal is written to the device. Why is this order of operations important?

Data Write
Journal Write
Journal Commit

It is important the data write happens first so that there are no pointers pointing to invalid locations on disk. If data were written last, the metadata already written would refer to it while it has not yet been written, for a short time.

The write is communicated to be successful after the journal is written because until the journal is written, the data is not persistent. That is, it's possible, if the system fails before the journal write is finished, that the data written to the device doesn't have the appropriate corresponding metadata on the device, and the device may not know of the data's presence until the journal write is done.

3. Does a URL more closely resemble a hard link or a soft (symbolic) link? Why?

A URL more closely resembles a symbolic link because it simply specifies the location, a means of finding a resource with no knowledge of how it is hosted and no guarantee that it is still there. (404)

- In the Unix file system, a hard link contains a pointer to the inode corresponding to a file, and it is accounted for in the inode's reference count.
- A symbolic link only contains the pathname of the file, and its existence is not known by the file's inode. Thus, there is no guarantee the file will be at the specified path since the inode could have been removed.

4. What is the benefit of using password salting? Why does it provide this benefit?

Password salting provides increased privacy by making it more difficult for an attacker to guess the password.
One way this is done is by adding a random number to the end of the password.

One common approach to guessing passwords is a dictionary attack, where the attacker guesses combinations of commonly used strings in passwords, like "123" or "password". This approach reduces the total # of possibilities the attacker has to guess, reducing the time it takes to arrive at the correct password (in the common case).

Password salting protects against dictionary attacks by adding something to the password that probably isn't in the attacker's "dictionary" and would take additional time to guess.

5. In performance evaluation of systems software, what is a factor? Why is the choice of factors important in such evaluations?

A factor is a variable or parameter that affects a running system's performance, by some metric.

The choice of factors is important in performance evaluation because while running in real time, there will be many factors that impact the system's performance, many of which we won't be able to control.

Thus, it is important to test the system while varying factors at different levels to account for the various situations that could arise at runtime.

6. In what way is a file descriptor like a capability?

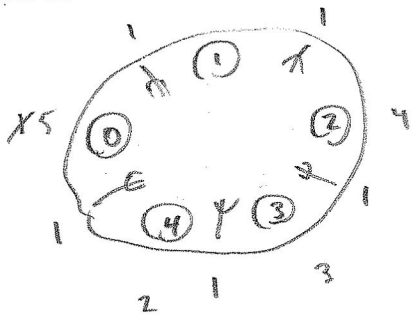
A file descriptor is the means by which Unix keeps track of an open file for a process and is how the process reads from and writes to the file. (for open instance of file) (per-user)

A capability specifies an entity's permission to access a particular object or type of objects and is one approach to access control.

Thus, a file descriptor is like a capability in that it specifies, for a process, that it is currently allowed and able to access (read/write) a particular object: a file.

7. Consider the following proposed solution to the Dining Philosophers problem. Every of the five philosophers is assigned a number 0-4, which is known to the philosopher. The philosophers are seating at a circular table. There is one fork between each pair of philosophers, and each fork has its own semaphore, initialized to 1. `int left(p)` returns the identity of the fork to the left of philosopher p, while `int right(p)` returns the identity of the fork to the right of philosopher p. These functions are non-blocking, since they simply identify the desired fork. A philosopher calls `getforks()` to obtain both forks when he wants to eat, and calls `putforks()` to release both forks when he is finished eating, as defined below:

all at once



```
void getforks() {
sem_wait(forks[left(p)]);
sem_wait(forks[right(p)]);
}

void putforks() {
sem_post(forks[left(p)]);
sem_post(forks[right(p)]);
}
```

Is this a correct solution to the dining philosophers problem? Explain.

This is not a correct solution. The four conditions for deadlock are:

1. mutual exclusion
2. incremental allocation
3. no preemption
4. circular dependency of resources

All four conditions exist.

1. Semaphores provide mutual exclusion
2. The two sem_wait() calls in getforks() are not atomic
3. Philosophers continue to wait for forks; not preempted
4. As they are seated circularly, there is circular dependency.

Consider the situation where all philosophers call getforks() at once, and each philosopher gets the fork to their left. Then we've achieved deadlock, since all philosophers will be waiting for the right fork.

8. What is the difference between synchronization using object-oriented monitors and synchronization using Java synchronized methods?

A monitor includes a mutex or semaphore and a condition variable, making some shared object safe to access.

A Java synchronized method effectively locks an entire function, allowing its code to be run by only one thread at a time.

The difference is that a monitor is more flexible, limiting access to a specified critical section, whereas a Java synchronized method is easier to use but limits access to an entire method, which may cover more code than is necessary.

9. What is the purpose of a callback in AFSv2?

In a file system that uses journaling, the system can fail while a journal entry is being written to disk. Upon system restart, this would be indicated by the presence of a transaction begin but no transaction end.

In this case, the system would likely perform a callback of any writes that had been performed but not logged in the journal. In other words, it would "undo" the changes that had been started but not finished and return the system to its prior state before the transaction began.

10. Describe how a certificate allows us to securely obtain a public key for some other party. What information, in addition to the certificate itself, must we have to be sure of the certificate's validity? Why?

A certificate allows us to securely obtain a public key by encrypting it with another trusted party's private key. Since that other trusted party is the only one with its private key, we can decrypt the certificate using the trusted party's public key, and, if the digital signature provided with the certificate is valid, we can be sure that the public key we received is the one we were looking for.

This implies that we need, in advance, the public key of the trusted party. Often these come with installed software like web browsers.

11. What is the purpose of a final state (also known as a zombie state) for a process?

When a process is killed, it enters a final state before going away so that related processes, like parent/child processes, that may have been waiting for the killed process, can still identify its presence and proceed accordingly. Once all processes that depended upon the killed process are finished, it will leave the zombie state.

Also, a process that was forked from the killed process will still have to refer to the old process' address space to perform copy-on-write.

12. If we use a scheduler algorithm that optimizes fairness, what other desirable property is likely to be damaged? Why?

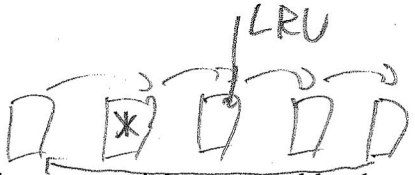
A scheduling algorithm that optimizes fairness, like Round Robin, is likely to hurt turnaround time, the time from a process' arrival to its completion.

This occurs because fair scheduling algorithms tend to give some processor time to each waiting process, rather than focusing on finishing any one process or subset of processes. This means that processes, on average, take longer to complete.

13. Elements in a memory free list could be ordered by size or could be ordered by their address. What is an advantage of ordering them by size? What is an advantage of ordering them by address?

One advantage of ordering them by size is it would be easier to find an element of a particular desired size, as we could use an algorithm like binary search that requires the list to be sorted. It would be very easy to find the largest block, for a worst fit algorithm, or the smallest free block.

One advantage of ordering by address is it would be easier to perform a coalescing algorithm to combine nearby free blocks, since nearby blocks would be adjacent in the free list. Coalescing reduces external fragmentation.



14. A looping sequential page workload runs sequentially through a set of pages of some fixed size, cycling back to the first page once it is finished with the last page. Why might an LRU page replacement algorithm handle this workload poorly? What kind of practical page replacement algorithm would handle it better?

In such a workload, the least recently used page is actually the page that is going to be accessed next, since the workload goes in order circularly. This workload exhibits no temporal locality, which is what an LRU algorithm relies upon.

A most recently used page replacement algorithm would handle this workload better, since in this case, it is the best approximation to "furthest in the future", the ideal page replacement algorithm.

15. What is the difference between load testing and stress testing? When is stress testing most likely to be used?

Load testing works to test a system under different levels of load to see how the system performs on these loads.

Stress testing is less systematic and tests a system under several levels of different factors, checking that the system performs even under unpredictable conditions and that no unforeseen combination of levels breaks the system.

Stress testing is more likely to be used in later stages of testing a system, when most of the main functionalities under expected loads and conditions have been developed and are working.

Stress testing might also be more practical on systems that have to run under less predictable conditions.