

CS 111 Final exam

AGI; KUBILAY YUCEL

TOTAL POINTS

141 / 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 7 / 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 10 / 10

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 10 / 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 **7 / 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 **10 / 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.

Final Exam
CS 111, Principles of Operating Systems
Winter 2018

Name: Kubilay Agi

Student ID Number: 304784519

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.

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

One mechanism that leads to the use of scatter/gather is ~~paging~~.

Data that is scattered all over physical memory needs to be combined to be processed (gather) and then redistributed to wherever the produced data fits into memory (scatter). This is because in modern memory management, process' memory is split up into pages and placed all over physical memory to avoid fragmentation. Scatter/gather needs to be able to run on contiguous data, so these pages are gathered.

The other mechanism is virtual memory. This leads to the need for scatter/gather because it also contributes to the reason why pages are distributed across physical memory. If programs used only segmentation and physical addresses, we wouldn't need scatter/gather.

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?

Firstly, the ordering of data on disk before meta data in journal is important because this eliminates the risk of metadata that points to random garbage data on disk. Secondly, it is important that a success is sent after the journal has been written because it ensures that crash recovery will be possible with this file. If the process was notified of success before the journal was written, and the system crashes before the journal was written, the process will think that the file has been persisted on disk even though it hasn't, which will cause an error when the process tries to access that file.

paging / virtual memory
✓ (S)

all the data is put into a single I/O buffer to be processed.

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

A URL more closely resembles soft links

because URL's contain pathnames (as do soft links).

For example, if a file is moved in the file system to a different path, the symbolic link will no longer work because the file no longer exists in the path contained by the symbolic link. Similarly with URLs, if google.com/images was changed to google.com/media , then typing in the first link into a browser would not work.

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

The benefit of password salting is that it protects against dictionary attacks. In most cases, passwords are not completely random (i.e. "abc" "123") so attackers will use the more common patterns to try to find a persons password first. Using salt numbers is useful because the attacker must ~~not~~ try every 32-bit or 64-bit combination of each password in a dictionary attack. Also it helps more generally for passwords that are short in length for the same reason; there are millions of versions of each one.

changes to hardware

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

A factor is something that can be changed in an experiment so that useful results can be extracted from the data. An example of a factor is the amount of RAM that a system has. The choice of factors is important in evaluations of performance is important because correctly chosen factors should provide useful information about a system. If arbitrary factors are chosen, then the results of the experiment won't change significantly and no information will be acquired. An example of a good choice of factors is the number of processors used with an embarrassingly parallel program.

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

If a process has a file descriptor, this means that it was granted access to this file before. This means that the same process should be able to use the file for as long as the operating system lets processes keep open file descriptors. It essentially acts as a cookie. This is useful because otherwise processes would continually have to ask the OS to reopen the file which would cause serious overhead. In this way, the OS trades a bit of security for performance and convenience.

(a capability is a type of ticket in the form of bits that can be presented for access to a resource.)

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:

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

With all semaphore values at -1 after they try to get the right fork, there is deadlock

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

This solution is incorrect because it will cause deadlock in the case where every philosopher is able to pick up the left fork before any philosopher tries to pick up the right one.

(In other cases it works fine). This error is nondeterministic and depends on the order of execution of each Philosopher's actions. The reason this could happen is that the thread running one philosopher's actions could be scheduled to yield in the middle of its get_forks() function.

A correct solution would be: (pseudo code)

```
void getforks(int philos_num) {  
    if philosnum == 4  
        get right  
        get left  
    else  
        get left  
        get right  
}
```

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

Object oriented monitors designate an object as having to be locked whenever any interaction with it happens. This means that regardless of which member function is called, the locks for the object will be obtained (even if you don't need a particular function to be locked). Java synchronized methods allow the programmer to specify which functions should be serialized which allows for more flexibility but means that the programmer needs to correctly identify which methods contain critical sections. The tradeoff is that O.O.-monitors are safer but can cause unnecessary serialization whereas java sync methods allow for more flexibility but rely on the programmer for safety.

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

The purpose of a call back is to let the clients know that the file in their cache has been modified. This allows the clients to avoid having to repeatedly check with the server if the file in their cache has been modified (as with the Sun NFS).

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?

Certificates allow us to get a public key for an entity because the certificate itself contains the public key. We need to have the signature of the ^{trusted} distributor of the public key (e.g. Symantec) and the public key of that institution to verify the signature. If we did not have the public key for the trusted distributor, we cannot verify their signature because their signature was made with their own private key.

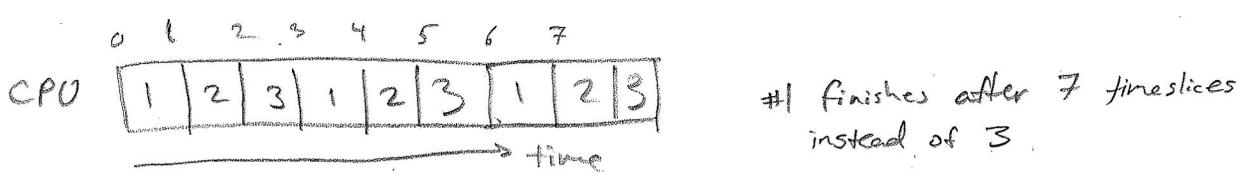
On the other hand, if we do know the distributor's public key, we can check the certificate's signature to make sure the certificate is not being given

out by a bad guy. This allows us to confirm that we are getting a valid certificate and protects us against bad certificates.

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

The purpose of the zombie state for processes is to allow for other processes to learn information about the process. This information includes how the process ended and which exit code was used.

This is useful for parent processes when their child processes complete or otherwise exit. From the information a parent process gathers, it can choose to act based on the results.



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

Our mean time to completion is likely to be damaged because one process has to wait an extra $(n-1) \times (\text{amount of time in timeslice})$ to complete its work (As shown in diagram above).

Another property that could be damaged is throughput. With extra context switches to make sure every process is running, we waste time that could be spent making progress in the process.

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?

If we order the free elements by size, this allows for quicker searches for a size that works with the method of allocation (e.g. worst fit or best fit, etc.). This would help the best and worst fit algorithms because they have to search the whole list before assigning a block to a request. Ordering free blocks by address allows us to combine (not sure how to spell coalesce) adjacent free blocks if they are adjacent in physical memory. This would help to avoid external fragmentation because smaller free blocks could be combined into larger, usable free blocks.

set: a, b, c, d

$a \rightarrow b \rightarrow c$ $b \rightarrow d \rightarrow d$ $d \rightarrow b \rightarrow a$
frame 1 frame 2 frame 3

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?

If we had 3 page frames available in RAM, but the looping sequence goes through a set of 4 pages, then LRU would perform poorly. It would ~~not~~ work fine for the first three iterations, but then you are ^{always} removing the page that ~~is~~ you will use two iterations from the current one. For example, you have gone from page 1 to 2 to 3. Now you need page 4 so you replace page 1 with page 4. But now you need page 1, which you just removed. To better handle this, we can designate a few pages that we swap out to replace with pages on disk so that we can keep the majority of pages in RAM. This way we only swap pages when we need to and only swap out the designated pages.

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

Load testing tests a program's ability to handle reasonable, normal usage scenarios. With load testing, different levels of workload are used to test how it performs. Stress testing is different because it uses unrealistically heavy workloads to see when the system will fail and its behavior under harsh conditions that aren't normally seen by the system. Stress testing is used when the designers want to see the robustness of their system. Websites like amazon.com likely use this to make sure their website can handle heavy traffic on days like Cyber Monday where many people do online shopping. ~~Ultimate, the ultimate test~~
~~Ultimate, the ultimate test~~