

## **CPSC 261**

### **Sample Final Exam Questions**

**Note:** the questions in this document do not constitute an actual final exam. However, the questions are representative of the kinds of questions that could be asked. Please keep in mind that this set of questions does not exhaust all the possibilities and therefore should not be used as your primary source of study material. Also note that this set of questions covers material that was not covered on the midterm. You should look at the sample midterm questions as well.

**Question 1. Virtual memory terminology**

Define each of the following terms:

1. Physical address
2. Virtual address
3. Address translation
4. Page tables
5. TLB
6. Memory-mapped file
7. Copy-on-write
8. Page fault
9. On demand (as applied to the “bring in” policy)
10. Pre-fetching

**Question 2. Modularity terminology**

Define each of the following terms:

1. Modularity
2. Abstraction
3. Soft modularity
4. Hard or enforced modularity
5. Client
6. Server
7. Virtualization
8. User-mode
9. Kernel-mode
10. Trap

**Question 3. Virtualization terminology**

Define each of the following terms:

1. Virtual processor
2. Virtual address space
3. Virtual disk
4. Virtual machine
5. Aggregation
6. Multiplexing
7. Emulation
8. Virtual communication channel
9. RAID
10. Thread

**Question 4. Short answer**

1. How are the physical resources in a computer managed? How are the virtual resources in a computer managed? Why are these two mechanisms different?

2. Which is more important for an operating system: correctness or performance? Explain your answer carefully.
3. Describe one benefit for keeping the interface to a virtual resource exactly the same as the interface to the corresponding physical resource.
4. Describe one benefit for modifying the interface to a virtual resource so that it is not exactly the same as the interface to the corresponding physical resource.
5. Describe the circumstances under which using the read system call will be faster than using the fread library call. Explain why it is faster.
6. Describe the circumstances under which using the read system call will be slower than using the fread library call. Explain why it is slower.
7. What is the difference in performance between a memory cache and main memory? That is, how many times slower (approximately) is main memory than the cache?
8. What is the difference in performance between main memory and disk?
9. What do the previous two questions tell you about the desirable hit rate for a virtual memory compared to a cache memory?
10. What is the “single writer principle”?
11. How does the single writer principle reduce the complexity of concurrent programs?
12. What is the difference between a mutex and a condition variable?
13. What is a deadlock?
14. Describe a simple technique for avoiding deadlock in threaded programs with multiple mutex’s. Explain how it works.
15. What is the granularity of isolation for:
  - a. Virtual processors (threads)
  - b. Virtual address spaces (processes)
  - c. Virtual disks (files in a file system)
  - d. Virtual communication channel (sockets)
  - e. Virtual machines
16. What does the notion of fairness in an operating system refer to?
17. What is a mechanism?
18. What is a policy?
19. What does it mean to separate mechanism from policy?
20. How does an operating system separate mechanism from policy?
21. Why is it a good idea to separate mechanism from policy?

### **Question 5. POSIX Threads**

If I have:

```
long x = 0;
#define N 1000000

void count() {
    int i;
    for (i = 0; i < N; ++i)
        x++;
}
```

What is the expected value of x if:

1. One thread executes the function `count()`?
2. Two threads execute the function `count()` simultaneously on 2 different processor cores?
3. The declaration of x is changed to `volatile long x = 0` and one thread executes `count()`?
4. The declaration of x is changed to `volatile long x = 0` and two threads execute `count()` simultaneously on 2 different processor cores?
5. How can the program be modified to get the “right” semantics when multiple threads execute the function `count()` simultaneously on multiple processor cores?

### **Question 6. Virtual Memory Organization and Address Translation**

1. Describe how a virtual memory system determines whether a particular page is in physical memory.
2. Explain why virtual memory is harder to implement than a memory cache.
3. In a typical computer these days, how many physical memory pages are there? This question requires that you know how big a page is and how much memory is in a typical computer.
4. Explain the process of address translation as implemented on the X86\_64 in 64-bit mode.
5. Imagine a virtual memory system with 16 physical pages of memory and 13 virtual pages. Each page is 16 bytes big. The location of pages is given by the following table:

Virtual Page Number	Physical Page Number
0	6
1	on disk at address X1
2	2
3	4
4	on disk at address X4
5	on disk at address X5
6	on disk at address X6
7	on disk at address X7
8	5
9	7
10	on disk at address X10
11	on disk at address X11
12	1

- a. How many physical pages are currently free?
  - b. What is the physical address corresponding to the virtual address 0x98?
  - c. What is the virtual address corresponding to the physical address 0x12?
  - d. What is a virtual address that will cause a page fault if accessed?
6. Real hardware doesn't implement virtual to physical address mappings with a single table as shown above. Why not?
  7. Suppose that we want to implement the virtual to physical mappings shown above with a two-level page table. That means that there is a single page directory (top level page table) that points to page tables (bottom level page tables) that point to physical pages. Each level of the page table has 4 entries.

- a. How many virtual pages can be mapped to physical pages using this table?
  - b. What is the maximum size of a virtual address? Or alternatively, how many of the bits in a virtual address are considered when performing address translation?
8. All page table entries consist of a “present” bit indicating that the entry is valid, and the (physical) address of either the next level of the page table (for the top level) or the address of the physical page (for the bottom level). Each of these entries is 32 bits long. This is bigger than required, but we are anticipating eventually having more than the 256 bytes of memory currently installed! The present bit is encoded in the least significant position of the entry. You may use any otherwise unused physical page to hold the page tables.
- a. Draw the page tables. Each entry in the page table should be a hexadecimal number. Only the least significant 8 bits of the entries will be interesting. What is the physical address of the top-level entry of the page table?
  - b. How many physical pages of memory are used to hold the mapping information?

### **Question 7. Mutex's and Condition Variables**

What is wrong with the following programs? Assume that all undeclared variables have been appropriately declared and initialized correctly depending on their types. An independent thread executes each of the functions named `threadN`. Explain each of your answers carefully and thoroughly.

1.

```
void thread1() {
    pthread_mutex_lock(&A);
    pthread_mutex_lock(&B);
    pthread_mutex_unlock(&B);
    pthread_mutex_unlock(&A);
}
void thread2() {
    pthread_mutex_lock(&B);
    pthread_mutex_lock(&A);
    pthread_mutex_unlock(&A);
    pthread_mutex_unlock(&B);
}
```
2.

```
void thread1() {
    pthread_mutex_lock(&L);
    if (!condition)
        pthread_cond_wait(&C, &L);
    // Do something that requires condition being true
    pthread_mutex_unlock(&L);
}
void thread2() {
    pthread_mutex_lock(&L);
    condition = 1;
    pthread_cond_signal(&C);
    pthread_mutex_unlock(&L);
}
```

3.

```
void thread1() {  
    pthread_mutex_lock(&L);  
    x++;  
    pthread_mutex_unlock(&L);  
}  
void thread2() {  
    if (x == 3)  
        // do something that requires x to be 3  
}
```

**Question 8. Disks and File Systems**

1. What is the purpose of the superblock of a file system?
2. What is an i-node?
3. Is the name of a file in the Unix file system stored in the i-node? Explain.
4. Is the access control information of a file in the Unix file system stored in the i-node? Explain.
5. What is special about the file: /dev/null?
6. What is special about the file: /dev/zero?
7. What is a seek (in the context of a disk)?
8. What does the term “rotational delay” refer to (in the context of a disk)?
9. How much data can be retrieved from a disk device in 1 second if:
  - a. The data is located sequentially on the disk?
  - b. The data is located randomly on the disk?

Carefully list and explain any assumptions that you must make in order to answer these questions.

**Question 9. Enforced Modularity**

A key component of the mechanism that provides hard modularity in operating system supported abstractions (or virtual resources) is the separation between user-mode and kernel-mode execution.

1. Explain what these two terms refer to: user-mode and kernel-mode.
2. What mechanism is used to transfer control between user-mode and kernel-mode? How does this mechanism work? What restrictions are placed on this mechanism?
3. What mechanism is used to transfer control between kernel-mode and user-mode? How does this mechanism work? What restrictions are placed on this mechanism?
4. What is the difference between a type 1 and a type 2 virtual machine?
5. Does one of the two types of virtual machine have an advantage when it comes to providing enforced modularity between virtual machine instances? Explain your answer very carefully! In particular, think about what the security people call the “attack surface”.

**Question 10. Networking**

1. What is a socket?
2. What is TCP?
3. What is UDP?
4. What is IP?
5. What is IPv6?

6. What is a router?
7. What is a switch?
8. What is the difference between a router and a switch?
9. Name and describe the 7 layers of the networking stack. For the purposes of this question, you can use either the names we used in class or the ISO standard names, but your descriptions must clearly delineate the differences between the layers.
10. Name and describe the purposes of the 4 system calls necessary for a server to be ready to receive network connections from clients.
11. Name and describe the purposes of the 3 system calls necessary for a client to initiate a connection to a server machine.
12. In the context of networking, what is a header?
13. What is a port?
14. Why are there addresses at (almost) every layer of the protocol stack? How are addresses at one layer related to the addresses at the neighbouring layers (above and below)?

### **Question 11. Scheduling**

1. Describe each of the following scheduling policies (in the context of threads) and identify the most significant advantage of each policy relative to all of the other policies.
  - a. First-come-first-served
  - b. Shortest-time first
  - c. Static priority
  - d. Dynamic priority
  - e. Round robin