Name	UBID	Seat	Side

Question:	1	2	3	4	5	6	7	8	9	10	Total
Points:	10	5	5	5	5	5	5	20	25	25	100
Score:											

CSE 421/521 Final Exam

11 May 2015

Please fill out your name and UB ID number above. Also write your UB ID number at the bottom of each page of the exam in case the pages become separated.

This final exam consists of four types of questions:

1. **Ten multiple choice** questions worth one point each. These are drawn directly from the second-half lecture slides and intended to be (very) easy.
2. **Six short answer** questions worth five points each. You can answer as many as you want, but we will give you credit for your best four answers for a total of up to 20 points. You should be able to answer the short answer questions in four or five sentences. These are mostly (but not entirely) drawn from second-half material.
3. **One medium answer** question worth 20 points drawn from second-half material. Your answer to the medium answer should span a page or two.
4. **Two long answer** questions worth 25 points each, integrating material from the entire semester. Your answer to the long answer question should span several pages.

Please answer each question as **clearly** and **succinctly** as possible—feel free to draw pictures or diagrams if they help. The point value assigned to each question is intended to suggest how to allocate your time. **No aids of any kind are permitted.**

There are 15 scratch pages at the end of the exam. When using them, please clearly indicate which question you are answering.

I have neither given nor received help on this exam.

Sign and Date: _____

Multiple Choice

1. (10 points) Answer all **ten** of the following questions. Each is worth **one** point.

   (a) Our unexpected April 1st visitor was
   ◯ handing out beads. ◯ wearing a blue body suit. ◯ not shy. ◯ Nick DiRienzo.

   (b) What was one approach that Wickizer et. al used to improve Linux scalability to many cores?
   ◯ Increasing the length of OS critical sections ◯ Sloppy counters ◯ Reducing application performance ◯ Asking Zihe

   (c) The RAID design principle can be best summarized as
   ◯ building expensive things from many inexpensive parts. ◯ building reliable things from many unreliable parts. ◯ five levels. ◯ striping.

   (d) Filesystems use _____ to map filename components to inode numbers.
   ◯ inodes ◯ directories ◯ superblocks ◯ Guru

   (e) After a crash, which of the following would indicate data loss for a journaling filesystem?
   ◯ multiple checkpoints ◯ journal entries after the last checkpoint ◯ incomplete journal entries

   (f) Which of the following makes it *easier* to virtualize the x86 architecture?
   ◯ hardware page tables ◯ instructions that are not classically virtualizable
   ◯ multiple privilege levels ◯ Jinghao's blog

   (g) Which of the following is *not* a good hint that a page might be a good page to swap out?
   ◯ It hasn't been used for a while ◯ It's clean ◯ It's currently loaded into a core's TLB

   (h) Applications will run exactly the same in a virtual machine as they would on real hardware.
   ◯ True ◯ False

   (i) Which of the following applications has become very similar to the traditional operating system?
   ◯ The web browser ◯ iTunes ◯ Microsoft Word ◯ `sys161`

   (j) We discussed all of the following kernel designs *except*
   ◯ microkernels. ◯ monolithic kernels. ◯ exokernels. ◯ xenokernels.

 UB ID: ☐☐☐☐☐☐☐☐

Short Answer

Choose **4 of the following 6** questions to answer. You may choose to answer additional questions, in which case you will receive credit for your best four answers.

2. (5 points) We have described (at least) two places where the data structure used by the OS presents a tradeoff between desirable properties. Identify one (1 point), describe what is is for (2 points) and describe the tradeoff (2 points).

UB ID: | | | | | | | | |

3. (5 points) Draw a simple diagram of a spinning disk. Label three parts (1 point each) and point out the major source of latency (2 points).

UB ID: | | | | | | | | |

4. (5 points) Describe the difference between full virtualization and paravirtualization (2 points), and how one way that the guest OS manipulates the virtual machine is handled differently by the two approaches (3 points).

UB ID: | | | | | | | | |

5. (5 points) State Amdahl's Law and describe how it guides the process of performance improvement.

UB ID: | | | | | | | | |

6. (5 points) Recall that in RAID level 1 (RAID 1) array, both drives store identical contents. (Assume the drives are spinning disks.) First, explain why you would expect to see a significant performance difference between reads and writes to and from a RAID 1 array (3 points). Second, describe how to coordinate RAID 1 reads to further improve performance (2 points).

UB ID: | | | | | | | | |

VPN	SLT
0	
1	450
2	
3	
4	
5	120
6	
7	
8	
9	320

(a) **TLT**

VPN	PPN
0	31
1	
2	
3	72
4	62
5	
6	40
7	43
8	
9	

(b) **SLT at 320**

VPN	PPN
0	94
1	
2	
3	10
4	13
5	
6	8
7	
8	
9	

(c) **SLT at 120**

VPN	PPN
0	9
1	84
2	
3	39
4	
5	
6	14
7	10
8	
9	

(d) **SLT at 450**

Table 1: **Process Page Tables**

7. (5 points) The `HappierStudent` architecture has byte-addressable memory with 10-byte virtual pages. The `HappierOS` operating system uses two-level pages tables to support 1000-byte virtual address spaces by dividing the virtual page number into a 10-byte second-level index and using the rest as the top-level index.

   First, given the virtual address `465`, identify the virtual page number, the offset, and the first and second level page table indices (1 point).

   Second, given the top-level page table (TLT) and second-level page tables (SLT) above for the currently-running process provided above, indicate the result of the following four virtual to physical page translations (1 point each).

   1. `182`
   2. `543`
   3. `901`
   4. `802`

UB ID: | | | | | | | | |

Medium Answer

8. (20 points) The Exobrowser v. The Exokernel

James Mickens has described his Atlantis browser as an "exokernel browser"—which we will (without permission) shorten to the exobrowser. This terminology reflects that Atlantis shares design principles from the exokernel OS design proposed and implemented by the Dawson Engler, Frans Kaashoek, and other members of the MIT Parallel and Distributed Operating Systems (PDOS) group, in another research paper we read this semester.

First, describe how the exokernel interface differs from the traditional monolithic OS interface (5 points). Second, describe how the exobrowser interface differs from the traditional monolithic browser interface (5 points). Finally, pick one of the design goals that inspired the original exokernel OS design. Describe it (5 points), and discuss how it translates to the exobrowser (5 points).

 UB ID: | | | | | | | | |

Why?	*Functionality* Does it work?	*Speed* Is it fast enough?	*Fault-tolerance* Does it keep working?
Where?			
Completeness	Separate normal and worst case	Shed load End-to-end Safety first	End-to-end
Interface	Do one thing well: Don't generalize Get it right Don't hide power Use procedure arguments Leave it to the client Keep basic interfaces stable Keep a place to stand	Make it fast Split resources Static analysis Dynamic translation	End-to-end Log updates Make actions atomic
Implementation	Plan to throw one away Keep secrets Use a good idea again Divide and conquer	Cache answers Use hints Use brute force Compute in background Batch processing	Make actions atomic Use hints

Figure 1: **Summary of the Hints**

Long Answer

9. (25 points) Hints for Computer System Design

Choose five of Butler Lampson's hints for computer system design shown in Figure 1. For each, describe the hint (2 points) and give an example of how to use it—possibly drawn from OS/161 (3 points).

 UB ID: | | | | | | | | |

10. (25 points) Cross-Device I/O Sharing

Today, a growing number of users interact with multiple personal computing devices: smartphones, tablets, laptops, and desktops. In this multi-device world, the single-device silos that traditional operating system designs were intended to operate seem increasingly obsolete and can prevent seemingly natural cross-device interactions. To illustrate this, consider the following two examples:

- Alice has music playing on her smartphone. When she sits down at her laptop, she wants the music to seamlessly begin playing from her laptop speakers.
- Bob is playing a racing game on his tablet. But instead of controlling it by tilting the tablet, which disrupts his view of the display, he wants to control the tablet game by tilting his smartphone.

To support such scenarios, let's consider how the OS may be able to provide a common platform for cross-device I/O sharing.

To begin, observe that operating systems such as Linux typically reuse the file abstraction to expose many different types of I/O devices to processes. For example, to play music through the sound card the music player would write to the pseudo-file `/dev/audio` and to read gyroscope (tilt) readings the game would read from the pseudo-file `/dev/gyroscope`. Like other places where Linux reuses the file abstraction, there are no contents stored on disk for these pseudo-files. Instead, file operations are converted into the necessary communication with the appropriate device.

Given this information, first describe a design for an OS cross-device I/O sharing system that would allow applications to make use of I/O devices on other nearby devices (10 points). Second, explain why it might be preferable in this case to modify the OS, rather than implement per-application solutions (5 points).

Third, imagine that you were presenting your design to a skeptical audience[1]. Discuss two aspects of your approach that you would need to evaluate to demonstrate that this was a workable solution (5 point each).

[1] Let's say, a future boss. Or Linus Torvalds.

 UB ID: | | | | | | | | |