University of California, Berkeley
College of Engineering
Computer Science Division – EECS

Spring 2013 Anthony D. Joseph

Final Exam *Solutions*

May 17, 2013
CS162 Operating Systems

Your Name:	
SID AND 162 Login:	
TA Name:	
Discussion Section Time:	

This is a **closed book and TWO 2-sided handwritten notes** examination. You have 170 minutes to answer as many questions as possible. The number in parentheses at the beginning of each question indicates the number of points for that question. You should read **all** of the questions before starting the exam, as some of the questions are substantially more time consuming.

Write all of your answers directly on this paper. ***Make your answers as concise as possible.*** If there is something in a question that you believe is open to interpretation, then please ask us about it!

Good Luck!!

QUESTION	POINTS ASSIGNED	POINTS OBTAINED
1	19	
2	9	
3	12	
4	13	
5	12	
6	23	
7	12	
TOTAL	100	

1. (19 points total) Short answer questions.
   a. (12 points) True/False and Why? **CIRCLE YOUR ANSWER.**
      i) The timers associated with the Two-Phase Commit, enable a guarantee that a commit or abort occurs within a fixed amount of time.

      TRUE FALSE
      Why?
      FALSE. *The correct answer was worth 1 point. Good reasoning was worth 2 points. If the coordinator crashes and all slaves have sent a READY message, then a decision about committing or aborting must wait until the coordinator restarts.*

      ii) Writes to an SSD are significantly slower than Reads from an SSD.

      TRUE FALSE
      Why?
      TRUE. *Writing an SSD is slower because of the high voltages and currents required to store data. In addition, when an erase is required, the time is much greater. The correct answer was worth 1 points and the justification was worth an additional 2 points.*

      iii) With demand paging, adding memory to a system always improves the cache hit rate.

      TRUE FALSE
      Why?
      FALSE. *Adding a memory to a demand paging system using a FIFO replacement policy can decrease the hit rate – this is known as Belady's Anomaly.*

iv) It is not possible to implement user-kernel separation without hardware support for dual mode operation.

TRUE FALSE

Why?

FALSE. *Software-based techniques can be used in place of hardware support.*

b. (4 points) What is authentication? Describe two fundamentally different schemes that can be used for authentication.

(1) Something in the user's possession (key, magnetic card, smartcard)
(2) Something the user knows (password)
(3) Something the user is – a biometric attribute (fingerprint, retinal scan, signature, handprint)

c. (3 points) Historically, what has been the difference between a computer virus and a computer worm?

A computer virus requires human intervention to spread (e.g., visiting a web site, running a Trojan program, connecting a USB key, etc.), while a worm spreads without any human intervention

2. (9 points total) Filesystems.

Consider a client using the 4.2 BSD FFS to store files. For this problem, assume that the file's inode has already been read into memory from the disk, there is an **(initially empty) filebuffer cache**, and the following constants:

T_y – CPU cycle time in nanoseconds
F – File size in bytes
T_L – Average time to read a block from the local disk, including filebuffer cache lookup and access time
B – Filesystem block size in bytes
S – Disk sector size in bytes
T_C – Average time to lookup and read a block from the filebuffer cache

The filebuffer is capable of storing the various types of blocks associated with files.

a. (3 points) On average, how long will it take to read a 5 block file?

*Since the inode is in memory and contains direct pointers to the first 10 blocks, it will only take 5 * T_L time to read the file.*

b. (3 points) On average, how long will it take to read a 49 block file?

*Since the inode is in memory and contains direct pointers to the first 10 blocks, it will take 10 * T_L time to read the first 10 blocks of the file. The remaining 39 block pointers will require a read of the indirect block (T_L) and the 39 blocks themselves (39 * T_L). So the total time will be 50 * T_L.*

c. (3 points) Now consider the performance when the cache is warm. Assuming that repeated reads of files yield an average cache hit rate of 80% and no compulsory misses, how long will it now take to read a 49 block file?

*We can reuse our calculation from part (b). 80% of the block references will hit in the cache, yielding and access time of 0.8 * 50 * T_C + 0.2 * 50 * T_L = 40 * T_C + 10 * T_L.*

3. (12 points) Address Translation.

Consider a multi-level memory management using the following virtual addresses:

Virtual seg #	Virtual Page #	Offset

Each virtual address has 2 bits of virtual segment #, 8 bits of virtual page #, and 12 bits of offset. Page table entries are 8 bits. *All values are in hexadecimal*.

Translate the following virtual addresses into physical addresses, or write *INVALID* if the virtual address is invalid:

Virtual Address	Physical Address
0x304345	***0x012345***
0x011BCD	***0x17BCD***

Virtual Address	Physical Address
0x109876	***INVALID!***
0x20A123	***0x4C123***

Each correct translation is worth 3 pts. For answers that listed a translation when it is invalid, we deducted 1 pt.

Segment Table

Segment #	Start	Size	Flags
0	0x2004	0x30	Valid, read only
1	0x0000	0x10	Invalid
2	0x2040	0x30	Valid, read/write
3	0x1010	0x10	Valid, read/write

Physical Memory

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x0000	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
0x0010	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D
….																
0x1010	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
….																
0x2000	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11
0x2010	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21
0x2020	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31
0x2030	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41
0x2040	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51
0x2050	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61
0x2060	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71
0x2070	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81

4. (13 points total) Concurrency.

a. (7 points) List the four properties of transactions. For each property, give a one-sentence explanation of the property.

Atomicity, Consistency, Isolation, and Durability (ACID).

b. Two-Phase Commit: In this problem, we will explore Project Four and the Two-Phase Commit (2PC) protocol you implemented for the project.

i. (3 points) Explain what the 2PC protocol is used for and why it is necessary.

2PC is used to guarantee consistency between the coordinator and the KVServers (SlaveServers).

ii. (3 points) In your implementation of 2PC, we did not require that you support coordinator failure. Explain how not supporting this type of failure simplified your implementation?

No logging at coordinator.

5. (12 points total) Demand Paging
For each of the following page replacement policies, list the total number of page faults and fill in the contents of the page frames of memory after each memory reference.

a. (4 points) FIFO page replacement policy:

Reference	**A**	**B**	**C**	**B**	**A**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**A**	**C**	**B**	**D**
Page #1	***A***	*A*	*A*	*A*	*A*	***D***	*D*	*D*	***C***	*C*	*C*	***B***	*B*	*B*	*B*	*B*
Page #2	-	***B***	*B*	*B*	*B*	*B*	***A***	*A*	*A*	***D***	*D*	*D*	*D*	***C***	*C*	*C*
Page #3	-	-	***C***	*C*	*C*	*C*	*C*	***B***	*B*	*B*	***A***	*A*	*A*	*A*	*A*	***D***
Mark X for a fault	***X***	***X***	***X***			***X***	***X***	***X***	***X***	***X***	***X***	***X***		***X***		***X***

Number of FIFO page faults? ______________**12**

b. (4 points) LRU page replacement policy:

Reference	**A**	**B**	**C**	**B**	**A**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**A**	**C**	**B**	**D**
Page #1	***A***	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	***D***	*D*	*D*	*D*	***C***	*B*	***D***
Page #2	-	***B***	*B*	*B*	*B*	*B*	*B*	*B*	*B*	*B*	***A***	*A*	*A*	*A*	*A*	*A*
Page #3	-	-	***C***	*C*	*C*	***D***	*D*	*D*	***C***	*C*	*C*	***B***	*B*	*B*	*C*	*C*
Mark X for a fault	***X***	***X***	***X***			***X***			***X***	***X***	***X***	***X***		***X***		***X***

Number of LRU page faults? ______________**10**

c. (4 points) MIN page replacement policy:

Reference	**A**	**B**	**C**	**B**	**A**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**A**	**C**	**B**	**D**
Page #1	***A***	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	***D***
Page #2	-	***B***	*B*	*B*	*B*	*B*	*B*	*B*	***C***	*C*	*C*	*C*	*C*	*C*	*C*	*C*
Page #3	-	-	***C***	*C*	*C*	***D***	*D*	*D*	*D*	*D*	*D*	***B***	*B*	*B*	*B*	*B*
Mark X for a fault	***X***	***X***	***X***			***X***			***X***			***X***				***X***

Number of page faults? ______________**7** – *Last column may place D anywhere since there are no additional references.*

6. (23 points total) Networking.

a. (8 points) Describe the execution steps for Remote Procedure Call (RPC) between a client and a server. Briefly explain why argument/return value marshaling is necessary.

i) *Client invokes a library routine called a client stub with parameter arguments.*
ii) *Client stub generates a message to be sent by marshaling the arguments.*
iii) *Client stub does OS syscall to send a message to the server.*
iv) *OS kernel on server receives the message and calls the server stub.*
v) *Server stub unpacks parameters and invokes server routine.*
vi) *Server routine runs and returns return values to the server stub.*
vii) *Server stub generates a reply message by marshaling the return values.*
viii) *Server stub does OS syscall to send a message to the client.*
ix) *OS kernel on the client receives the message and returns to the client stub.*
x) *Client stub unpacks return values and returns to the client.*

Argument/return value marshaling is necessary because memory data structures must be converted into byte strings, pointers must be dereferenced, byte ordering differences between the client and server must be reconciled, etc.

b. (6 points) List THREE advantages and/or disadvantages of using layering for networking protocols.

Pro: encapsulation, reuse/modularity
Cons: inefficient implementations

c. (4 points) Consider the end-to-end argument and the careful file transfer example in the Saltzer, Reed, and Clark paper. In **three sentences or less**, explain whether using TCP as the networking protocol simplifies or does not simplify an implementation of careful file transfer, and justify your answer.

*Using TCP will **not** simplify the implementation, since while TCP guarantees correct delivery of bytes from host A to B, the careful file transfer will still need to perform an end-to-end check that the bytes on the disk at B are the same as the bytes on the disk at B. The only benefit from using TCP is that retransmissions can occur on a smaller scale.*

d. (5 points) Consider a sliding window protocol and the following constants:

T_{RTT} – Roundtrip latency between end hosts = 80 milliseconds
P – Packet size = 1,000 bytes
C – Link capacity = 10^9 bits/second (approximately 1 Gbit/sec)
B – Disk block size = 500 bytes
T_{cycle} – Processor cycle time = 10 nanoseconds

What is the window size W to match link's capacity, C?

$Throughput = W * P / T_{RTT}$
$C = W * P / T_{RTT}$
$W = C * T_{RTT}/P$
$= 10^9 \text{ bits/sec} * 80 * 10^{-3} \text{sec} / 8{,}000 \text{ bits} = 10^4 \text{ packets}$

7. (12 points total) E-mail Security.
One of your friends proposes a new way to send authenticated messages: a sender "signs" their email by adding at the bottom of an otherwise normal email message, the sender's name and the date encrypted using the sender's private key. The message itself is unencrypted, and includes the sender's public key so that the signature can be validated.

Explain why this gives a completely false sense of security, by outlining three different ways that you could make it appear that the sender signed mail saying "Berkeley CS is #20." The definition of "different" is that each attack has a unique fix. For each of the three attacks you list, give a countermeasure that the sender/receiver could take to protect themselves against just that one attack, where the countermeasure would not help against any of the other attacks you list.

You may assume that the sender and receiver are on different machines, that both are running on "diskless" workstations whose OS and application files are provided by an NFS server, and that you have the ability to spy on and/or alter packets on any network at the sender's or receiver's sites.

Attack/Countermeasure #1:

Missing HMAC on messages, public key CA trust, files unencrypted over NFS.

Attack/Countermeasure #2:

See above.

Attack/Countermeasure #3:

See above.