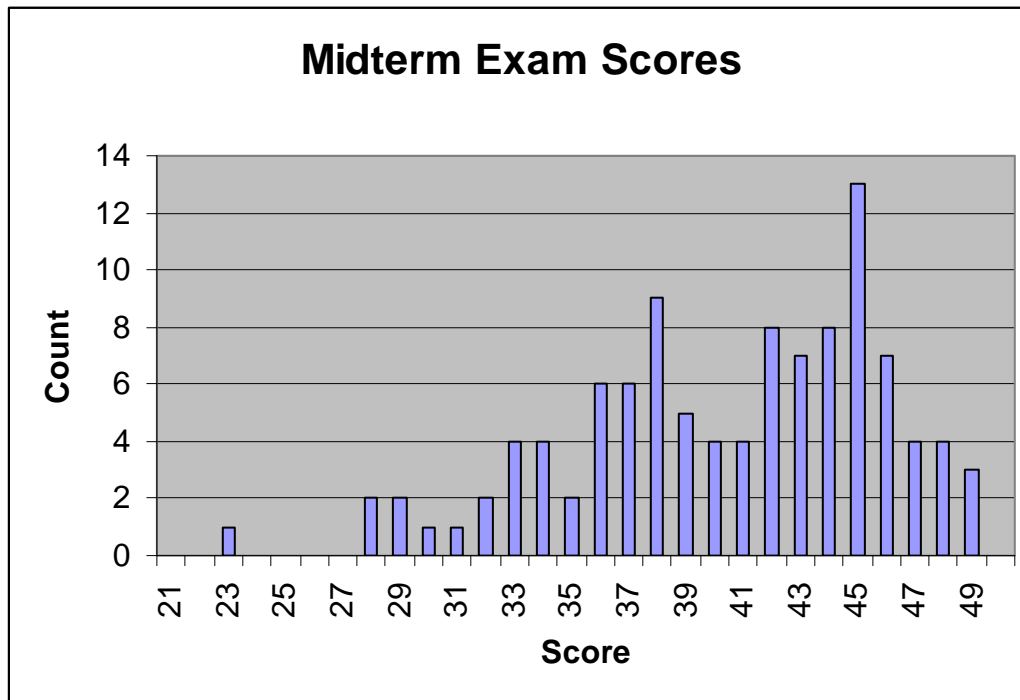


CS110 - Principles of Computer Systems

Midterm Exam Solution Winter 2010 (Average 40.4, StdDev 5.5)

(Total time = 50 minutes, Total Points = 50)



This examination is close book and close notes. You may not collaborate in any manner on this exam. You have 50 minutes to complete the exam. Please write your answers on the exam. Note there is one problem per page so the amount of space provided does not necessarily provide an indication of the expected length of the answer. In other words, do not feel compelled to fill every nanoacre of the exam with writing. Before starting, please check to make sure that you have all 7 pages.

Question	Points	Score
1	6	
2	7	
3	8	
4	7	
5	7	
6	9	
7	6	
Total	50	

1. (6 points) The course reader describes four generic categories of problems faced by systems. The categories were: *Emergent properties*, *Propagation of effects*, *Incommensurate scaling*, and *Trade-offs*. Which of these categories is helped by using a client and service architecture for a system? Justify your answer.
-

Graded by Sommer.

An answer receiving full credit had to state that the category primarily impacted was PROPAGATION OF EFFECTS. The justification had to in some way describe the enforced modularity brought about by the client service architecture (i.e. - by saying "enforced modularity" explicitly, describing message passing, etc.). Solutions that described other categories received partial credit IF the assertion was well justified.

2. (7 points) While running the Windows operating system on a simulator it was noticed that there were many procedure calls to routines that did nothing else but call another routine. The source code for these “null” procedures must of looked something like:

```
returnType
RoutineA(arglist)
{
    return RoutineB(arglist);
}
```

where *arglist* is the list of arguments to the function and *returnType* is the type of the return value.

The chain of “null” procedures sometimes reached up to dozens of calls before reaching a routine that actually did some real work. This seemed strange since it would be equivalent functionality to have the original routine directly call the routine that performed the work. It was further noted that the routines themselves were widely spread in the kernel like they were each coming from a different module of the kernel. This structure is related to one of the following concepts from the course.

Identify the concept and explain why this is the case.

- a) Naming conflict
- b) Layer bypass
- c) Enforced modularity
- d) Deadlock avoidance

Be sure to justify your answer.

Graded by Mendel

The structure was caused by Window’s deeply layered module structure trying to emulate a layer bypass. Rather than directly calling the desired layer, each of the layers being bypassed simply passed the call down. The likely reason they didn’t directly call the bottom most layer is they wanted to preserve the layering in case changes if the future required the layer to do something. They didn’t want to do a layer bypass.

Hence the structure is most related to **layer bypass**.

Some partial credit was given for other answers. Many students listed Enforced Modularity. Although this is all about modularity, there is no enforced modularity here.

3. (8 points) For each of the following elements of a computer system, state if the object would be best classified as a *memory*, *interpreter*, or a *communication link*. Briefly (a single sentence) justify your answer.
- Hash table
 - Java Virtual Machine
 - File
 - x86 processor
 - Firefox browser
 - gcc compiler
 - Microsoft Word editor
 - The Internet
-

Graded by Ryan.

Hash table

-Memory: A hash table is a data structure that allows for associative memory. It is NOT an interpreter because it doesn't have execution state--it just IS. English words are pretty ambiguous, but just because something has a mapping from one set of values to another does not automatically classify it as an interpreter for this definition.

Java Virtual Machine

-Interpreter: Interprets or rapidly compiles Java bytecode into machine-dependent instructions.

File

-Memory: An abstraction built on top of the disk to simulate a bucket o' data. This is NOT an interpreter for the same reasons as the hash table wasn't. It is not a communications link either, as last time I checked my jpgs weren't sending me emails just to say "hi".

x86 processor

-Interpreter: Kind of the most visible example of a von Neumann-style interpreter on the planet. Has register state, eats instructions, executes operations on data.

Firefox browser

-Interpreter: The primary task of the browser (thus the "best" answer for this question) is to interpret HTML/CSS/js into web pages and to interpret the user's actions on the page as commands. Given this perspective it's easy to see interactive software applications as interpreters for complex user command sequences (like Facebook pokes). It's not really justifiable to call the browser a communications link, as you're still connected to the network when you close Firefox. Firefox's purpose as an application is facilitating interaction between web pages and users.

gcc compiler

-Interpreter: Eats C/C++/Obj-C code and processes it into machine-dependent assembly. You best believe it takes a lot of state to change language syntax while maintaining the meaning of the original source.

Microsoft Word editor

-Interpreter: For the same reasons as Firefox above, only this time the application is executing user commands to make documents.

The Internet

-Communications link: The internet is a man-made communications backbone linking groups of computers the world over. Internet Protocol is a datagram service, meaning it takes data and discretizes it into individually addressed packets to be routed through connected machines. Classic communications link--put message into pipe, pipe holds message for a while, message emerges from pipe at destination.

4. (7 points) My cell phone behaves differently depending on my location. When I'm at Stanford I can call my office by dialing 7230474 but when travelling on the USA east coast I have to dial 16507230474 and while in France I had to dial +16507230474. Which of the following terms used in our discussion of naming best describe what is happening here? Be sure to explain your answer.
- Name space
 - Name mapping algorithm
 - Universe of values
 - Naming context
-

Graded by Sommer.

Full credit was given for stating D. naming context was the best description of the scenario. Additionally, the answer had to include a description of how context is location based in this case, with +1650 being the local/default context in the global namespace of phone numbers. Info had to be added to specify a context outside of the local/default context. Deductions are results of either selecting the wrong choice or an unclear explanation of the scenario.

5. (7 points) What are the tradeoffs between using a RPC versus a procedure call for accessing a module? Your answer should include at least one advantage and one disadvantage for each of the approaches.
-

Graded by Mike

One advantage of RPC is providing enforced modularity and reducing error propagation. Another advantage is that a slower machine can offload computationally heavy work to a more powerful machine. Disadvantages of RPC are:

- RPC introduces new failure modes. Messages can get lost.
- Global variables cannot be used with RPC.
- Arguments and result need to be passed by value, resulting in a lot of information copying.
- RPC can take more time because of reasons above as well as network delays.

6. (9 points) For each of the C language statements below, state if the statement has a race condition if multiple threads execute it simultaneously. You can assume that X is an C integer data type (e.g. char, short, int, long, or long long). Function getint() and printf() are assume to be thread-safe (e.g. multiple threads can call into it without race conditions).
- a. `X = X + 1;`
 - b. `X = getint();`
 - c. `{ int tmp; tmp = X; printf("%d\n",tmp); }`
-

Graded by Haider

- a) [Yes] Thread T1 and T2 both read the value of X, T1 adds 1 to the value of X it read and writes the result into X. T2 also adds 1 to the value of X it read and writes its result into X. The result is that the value of X is incremented by 1 and not by 2.
- b) [Yes] This is the multiple writers conflict. Threads T1 and T2 both call getint() and get different values. Assuming X is a long long and requires multiple writes to be fully written into memory then T1 could have written half the value into X when its gets switched to T2 which goes ahead and writes its value into X. When T1 wakes up again it writes the other half into X. The resulting value in X is neither equal to thread T1's value nor thread T2's.
- c) [No] The variable tmp is a local variable and is allocated on the stack. Each thread has its own stack so tmp is private to each thread. There is thus no conflict in writing the value to tmp.

7. (6 points) A famous database researcher criticized the Unix file system for what he called the “double fault” problem. His database system kept its data in a very large file in a Unix file system. The researcher noticed that accesses to the database file seemed to generate twice the number of disk read requests as compared to the number of file read system calls going to the database file. This problem did not occur if the database did not go through the file system and instead stored data directly on the disk. Using your knowledge of the Unix file system implementation explain how this could happen.
-

Graded by Haider

In Unix version6 the file system was optimized for accessing small files. For small files the inode contained pointers directly pointing to the data blocks. Large files were handled using indirect and double indirect blocks which meant going through extra layers of indirection to get to the data blocks for large files.

The database when stored as a single large file suffered from the overhead of these extra indirections. It generated twice the number of disk reads as on the average it had to read in one indirect block to get to the required data block.