5c. (*6 points*) An upgrade to a supercomputer fixes that problem, but it's too late -- Louis is obsessed with performance. He switches from RPC to an asynchronous protocol, which groups several requests into a single packet if they are made within 2 milliseconds of one another. On his network, which has a very high latency, he notices that this speeds up some workloads far more than others. Describe a workload that is sped up and a workload that is not sped up. (An example of a possible workload would be one request every 10 milliseconds.)

5d. (*6 points*) As a design engineering consultant, you are called in to critique Louis's decision to move from RPC to asynchronous client/server. How do you feel about his decision? Remember that the Toastac software sometimes fails with a "Malfunction 54" instead of toasting properly.

Name: ______________________________



5. (*24 points*) Louis P. Hacker-Reasoner (no relation) bought a used Therac-25 for $14.99 at a yard sale. After some slight modifications, he's hooked it up to the Ethernet as a computer-controllable turbo-toaster, which can toast one slice in under 2 milliseconds. (Louis likes his toast black.) Louis decides to use RPC to control the Toastac-25. Each toasting request starts a new thread on the server, which cooks the toast and returns an acknowledgement (or perhaps a helpful error code, such as "Malfunction 54"). Here's what the server thread looks like:

```
server thread:
    lock(message_buffer_mutex);
    decode message;
    lock(accelerator_mutex);
    unlock(message_buffer_mutex);
    cook toast;
    lock(message_buffer_mutex);
    put acknowledgement in message buffer;
    send message;
    unlock(accelerator_mutex);
    unlock(message_buffer_mutex);
    exit
```

5a. (*6 points*) To his surprise, the toast server stops cooking toast the first time it is heavily used! What has gone wrong? Refer specifically to the code above.

5b. (*6 points*) Once Louis fixes the multithreaded server, the Toastac gets more use than ever. However, when the Toastac has many simultaneous requests (i.e., there are many threads), he notices system performance degrading badly -- much more than he expected. Performance analysis shows that competition for locks is not the problem. What could be going wrong? Explain briefly.

Name: ______________________________

4c. (*6 points*) Alyssa decides that grouping words still isn't fast enough, so she wants to know how long it would take if she used asynchronous messaging (with grouping words) instead of RPC. How long will it take to spell check the same file? (She is careful to resend packets that are not acknowledged but, for your calculations, you may assume that packets are not lost or reordered.)

4d. (*6 points*) Alyssa is so pleased with the performance of this last design that she decides to use it (without grouping) for a banking system. The server maintains a set of accounts and processes requests to debit and credit accounts (i.e., modify account balances). One day Alyssa deposits $10,000 and transfers it to Ben's account immediately afterwards. The transfer fails with a reply saying she is overdrawn. But when she checks her balance afterwards, the $10,000 is there! Draw a time diagram explaining these events.

Name: ______________________________



4. (*22 points*) Alyssa P. Hacker is implementing a client/server spell-checker. The client scans an ASCII file, sending each word to the server in a separate message. The server checks each word against its database of correctly spelled words and returns a one-bit answer. The client displays the list of incorrectly spelled words.

4a. (*5 points*) The client's cost for preparing a message to be sent is 1 millisecond, regardless of length. The network latency is 10 milliseconds, and network bandwidth is infinite. The server can look up a word and determine whether or not it is mispelled in 100 microseconds. Since the server runs on a supercomputer, its cost for preparing a message to be sent is zero milliseconds. Both the client and server can receive packets with no overhead. How long will Alyssa's design take to spell check a 1,000 word file if she uses RPC for communication (ignore acknowldgements to requests and replies, and assume that packets are not lost or reordered)?

4b. (*5 points*) Alyssa does the same computations that you did and decides that the design is too slow. She decides to group several words into each request. If she packs 10 words in each request, how long will it take to spell check the same file?

Name: ______________________________

2. (*18 points*) Ben Bitdiddle has written a program with 16 major modules of code. (Each module contains several procedures.) In the first implementation of his program, he finds that each module contains at least one call to every other module. Each module contains 100 lines of code.

2a. (*1 point*) How long is Ben's program in lines of code?

2b. (*4 points*) How many module interconnections are there in his implementation? (Each call from one module to another is an interconnection.)

Ben decides to change the implementation. Now there are 4 main modules, each containing 4 submodules in a one-level hierarchy. The 4 main modules each have calls to all the other main modules, and within each main module, the 4 submodules each have calls to one another. There are still 100 lines of code per submodule, but each main module needs 100 lines of management code.

2c. (*2 points*) How long is Ben's program now?

2d. (*5 points*) How many interconnections are there now? Include module-to-module and submodule-to-submodule interconnections.

2e. (*6 points*) Was using hierarchy a good decision? Why or why not?

Name: ______________________________



3. (*21 points*) Mike R. Kernel is designing the Intel P97 computer system, which currently has one page table in hardware. The first tests with this design show excellent performance with one application, but with multiple applications, performance is horrible. Suggest 3 design changes to Mike's system that would improve performance with multiple applications, and explain your choices briefly. You cannot change processor speed, but any other aspect of the system, from disk to MMU design, is fair game.

Name: ______________________________

1. (*15 points*) We've discussed 4 principles for dealing with complexity: modularity, abstraction, hierarchy, and level definition. For each system organization below, choose *the most important* principle it uses or implements that makes it effective, and explain *briefly* why the system organization is an example of that principle. *An answer without explanation will receive no credit.* If you think two or more principles are equally important, choose one, and explain it well.

1a. *(3 points)* Protocol stack

1b. *(3 points)* Client/server

1c. *(3 points)* Virtual memory

1d. *(3 points)* RPC

1e. *(3 points)* Microkernel

Name: ____________________________

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY



# Quiz 1

This quiz contains five problems whose relative weights are given below. Please write your answers in the space provided, and hand in this booklet at the end of the examination.

Put your name on this cover sheet AND at the bottom of each page of this booklet. Be sure that you don't put part of the answer to a problem on the back of a sheet for another problem.

Some parts of some problems may be much harder than others. Read them all through first and attack them in the order that allows you to make the most progress. Some problems are intentionally ambiguous; be sure to write down any assumptions you make. Show your work, or you risk losing partial credit. Be neat. If we can't figure out your answer we can't give you credit. On answers that involve numbers, be sure to clearly specify the units of your answer.

Circle your recitation:

| | | | | |
|---|---|---|---|---|
| **10:00** | 1. Shrira/Litt | | | |
| **11:00** | 7. Corbató/Fu | 5. Ganger/Yerushalmi | 12. Shrira/Kohler | 2. Wetherall/Legedza |
| **12:00** | 6. Wetherall/Yerushalmi | | | |
| **1:00** | 3. Agarwal/Legedza | 11. Corbató/Litt | 9. Ganger/Fu | 8. Saltzer/Engler |
| **2:00** | 4. Agarwal/Kohler | 10. Saltzer/Yerushalmi | | |

*For Official Use Only*

| Problem | Your Score | Maximum Score |
|---|---|---|
| 1 | | 15 |
| 2 | | 18 |
| 3 | | 21 |
| 4 | | 22 |
| 5 | | 24 |
| Total | | 100 |

Name: ____________________________