Assignment 2 - Solutions

1. For a uniprocessor server **(5 points)** -

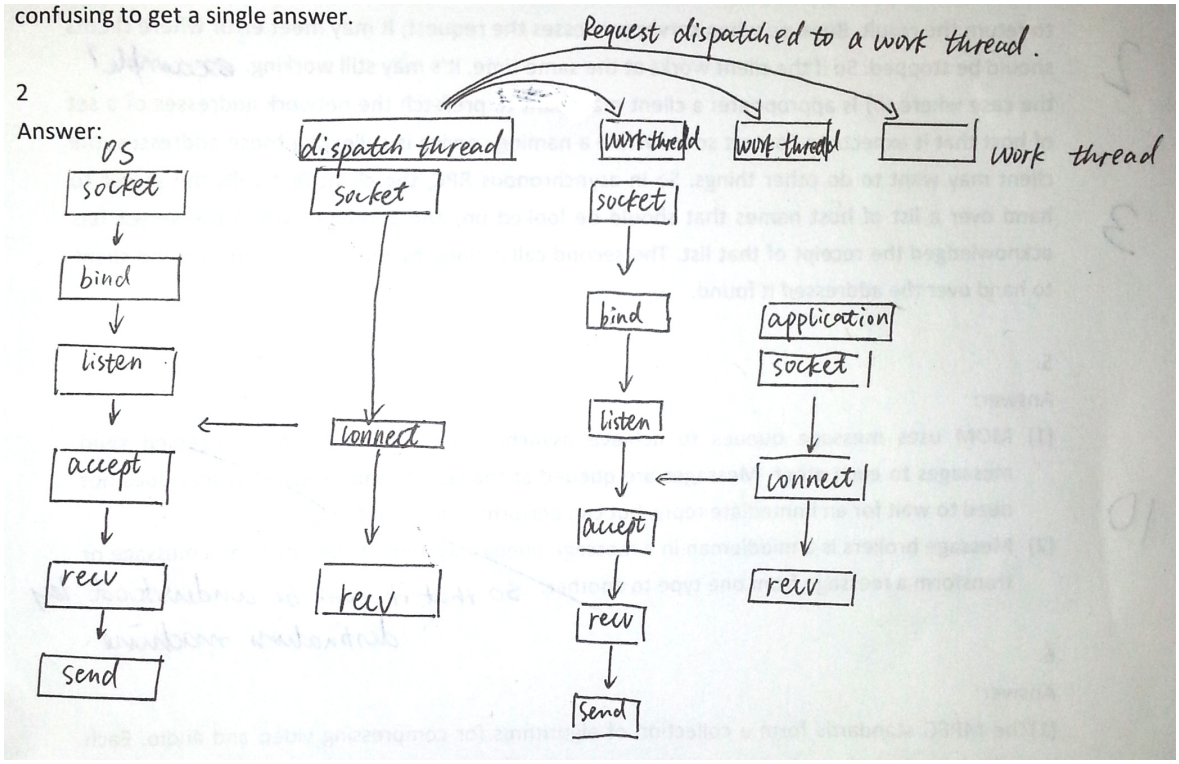
Access Time = (⅓)*30 + (⅔)*(30+90) = 90ms
Req/s = 11.11

For multithreaded server **(5 points)** -

The access time would approximately be 30ms for a large number of processes because if a thread doesn't find the data in the cache, a new thread can be created to continue with execution while the thread that couldn't find data goes for a disk access. It involves quite a few assumptions and that's why I have given full credit for any answer.

Access Time = 30ms
Req/s = 33.33

2. Any high-level, block-level design was given full credit. **(10 points)**
I have posted an answer from one of the students. Most of the answers were similar to this.



3. The expected answer was that a process can be created at the machine which contains F which can handle requests for F. Note that the process at this machine need not be created using fork(). That would be a special case where the process on this machine would be identical to the process P and would require access to the file F, which is not necessary.
I have given full credit for answers which talk about FTP, URL, RPCs if you have given proper explanation. Note that even these would require a process at the local machine to handle requests for the file.
I have taken off points for answers which talk just about references (without a stub at the local machine) or global caches because the whole point of the question was that you shouldn't move the resource F.
(10 points)

4. Synchronous RPC - Client has to wait until it receives reply from server **(2 points)**
Asynchronous RPC - Client only waits till it gets an ACK from server **(2 points)**

Examples :
Synchronous RPCs - File Transfer (or any secure communication), Berkeley Sockets, MPI or any other appropriate example **(3 points)**
Asynchronous RPCs - Message Oriented Middleware, e-mail, News or any other appropriate example **(3 points)**

5. Asynchronous communication is implemented using message queues. Messages sent are stored in queues and the sender doesn't have to wait for an immediate reply and can continue with its execution. **(5 points)**

Message broker converts incoming messages so that they can be understood by the destination application. **(5 points)**

6. Middleware offers a collection of interfaces for controlling audio and video streams including interfaces for controlling devices such as monitors, microphones etc. Each device and stream notify an application when an event has occurred and the application in turn tells the middleware what to do with the incoming streams. It uses a locally available synchronization specification to synchronize the audio and video streams. **(7 points)**
I have given full credit for any acceptable answer.

Reducing jitter - Buffering/Interleaved Transmission **(3 points)**