Take home exam clarifications

Question 1

1) M_n(p,k) is defined to be the set of pages in memory after the k-th page request has been handled, and n is the number of pages in memory. What is "the number of pages in memory"? Is it the total capacity of the memory, or is it the number of pages from set p in memory?
   Clarification: "n" is the capacity of memory, i.e. the number of pages that fit in it.

2) It asks to "show" can we just explain in words instead? Reason, in part c we need to show for any algorithm, this is quite impossible unless we are only to use the algorithms we have discussed in class. Also for d I believe that LRU does satisfy the containment property always and I am not sure how to show that unless I break out my discrete math book, yuck!!!
   Clarification: No it is not acceptable to explain in words. What we need is a proof, and detailed discrete math concepts are not required for it. What we require is a sequence of reasoning with some math that shows that this property has nice consequences on algorithm design. Further, just showing that the property in (c) holds for algorithms discussed in class is not a proof!

Question 2

1) Question 2: In the set-up of the problem, you state that there is a transfer rate of t bytes/sec. This is the DMA transfer rate to memory and not the actual rate of transfer from disk to some sort of internal buffer in the disk, correct?
   Clarification: This is the DMA transfer rate.

2) Question 2: We are to assume that "the transition time for the disk after reading a sector and before it can read the next sector is less than the time taken to rotate through one sector". Does this mean that the disk can or can not read an entire track at its rotation rate of r revolutions per second? Does it read a sector, stop rotating, then start rotating and read the next sector?
   a. We are to assume that the OS fetches one block at a time. Does this mean that for each of the N blocks in the file, there is a set-up time of d seconds associated with the DMA device?
   b. I don't remember "interleaved sector allocation" ever being defined. I can't find it in the lecture notes or in the textbook.
   c. Do we know anything about the per-track cache? We don't need to read any sectors more than once, so how does this help unless the cache already contains some of the sectors we need?

   Clarification: The disk cannot read the entire track in a rotation. For (a), there is a DMA set up time for every block. (b) see slide 4 of lecture 27. Contact me if you have further questions. (c) think of how in one rotation, DMA can transfer only some of the blocks!

3) (b) In part b you say that the disk controller implements an interleaved sector allocation. What is the interleave ratio? Is it just single (2:1) interleave? (c) you say that the disk controller also adds a "per-track cache in the disk". I am confused by this statement. Could you please elaborate? I see many different ways to interpret this statement -- does this mean that, when asked for a single sector from a certain track on the disk, the controller will read the surrounding sectors on that track and cache up to a whole single track in its internal buffer? Or does it mean something else entirely?
   Clarification: (b) It is a single interleave. Lecture 27, slide 4, figure b.

4) Question 2: Where do we assume that the head will be at before going to track 0 and the sector that the file starts at? Further, it seems contradictory in one of the emails when you say that if reading a sector and before it can read a next sector is less than the time taken to rotate thru one sector, makes me believe that you can read one sector and immediately read the next. However, your answer to that email stated that the disk cannot read the entire track in one rotation.
Clarification: The head could be anywhere, and we add the average rotational delay to your expression. To clarify the second part, a disk will read a sector, and while the sector is being transferred using DMA, the disk head will have gone into the second sector. By the time the DMA engine is ready to transfer the next sector, the disk head is well beyond the beginning of the sector. Think of what we talked about when we discussed interleaving! The statement "reading a sector and before it can read a next sector is less than the time taken to rotate thru one sector" implies that single interleaving is sufficient, and we do not need double interleaving for this particular disk.

5) Question 2: Are DMA transfers done in units of blocks or units of sectors? It is stated that "the OS fetches one block at a time" and "there is a DMA set up time for every block", but later on, it is stated that the single interleave will help avoid having to make another rotation when DMA is being done on the first sector. It seems natural to assume that the OS asks for data from the disk in units of sectors, but the two earlier statements seem to contradict this. Which is it?
Clarification: DMA transfers are done in units of sectors. OS fetches data in blocks and DMA satisfies it by transferring data in sectors. BIG CLARIFICATION: Assume the per-track cache is of size 1 track, and not 1 sector. The algorithm proceeds as follows: The disk controller can keep an entire track on the cache. Note that here one can avoid the initial rotational delay to get to a particular sector since you can start on any sector and read the entire track in the cache in one rotation.

6) Question 2: How big is the disk buffer? That is, can the disk buffer hold 2 sector's worth of data, so that if after reading the first sector, and the buffer holds the first sector's data, there is no delay when the disk moves to the second sector(which would be interleaved) to read the second sector? The logical progression of events would be: disk reads first sector, disk puts first sector in buffer, disk must now use transfer rate of t bytes/sec to get data to memory AND disk rotates to next sector to read second sector in block disk reads second sector OR disk missed the second sector and must rotate back etc. The question is: does the disk have to complete transferring the data out of the buffer before rotating to the next sector? If it does, then, does that mean that the disk head flew by the next sector by the time it finished transferring the data out to memory, thus incurring an additional rotation time? I ask this because in part c, the cache would alleviate this flaw, but then interleaving may not have an effect in part b. Furthermore, the "transition time" between reading interleaved sectors: is this referring to the transfer of the data from the buffer to the memory?
Clarification: The disk keeps rotating when there are more requests. It will not stop for the data transfer to end. Note that the cache can store one track. This should indicate the difference between interleaving and caching.

7) Question 2c: If interleaving is implemented and we are to assume that the track cache is of size one sector, then the cache really serves no useful purpose since interleaving makes the data in the track cache irrelevant (i.e. want sectors 0,1, finished reading sector 0, doing DMA buffer transfer, but track cache has sector 4:

using lec27-pg4-figb). Is the one sector size cache the correct assumption we should be making?
Clarification: I am sorry for the confusion. Assume the per-track cache is of size 1 track.

8) Does simultaneously asking DMA to fetch c blocks imply that the setup time is now per c blocks.
Clarification: Yes.

9) Question 2: Clarification on per-track caches (Note that if you have solved this question under a different set of assumptions then state it in your answer. For those who want to know more, I am sending in more details) For some disk drives, the time required to seek to a new cylinder is still more than the rotation or transfer time. So, once the driver has gone to the trouble of moving the arm somewhere, it does not matter a whole lot if the driver reads one sector or the whole track. Some disk drivers use this property to maintain a secret per-track cache. So, the entire track is read in one rotation and stored in the cache, as opposed to using up one half rotational delay + one sector time to read a sector. If a sector is in the cache, then no head movement is required. Sorry about the earlier mail when I said that the per-track cache is one sector. If you have solved this problem under that assumption, it is still alright. Remember to state the algorithm you are using.

10) Further clarifications on Question 2: The cylinder does not have any skew. Also, assume that the disk has one platter and data on one side. Further, the time you have to calculate is the time to read the entire file from the beginning to the end. So, the initial head movement to get to the first block of the file should NOT be there in the expression.

11) Question 2: For b, is the interleave spacing great enough to assume that the next sequential sector can be read immediately without having the disk to revolve on average half way around. Further, is the spacing so great, that there is an extra time factor we need to revolve to it? For c, is the transfer time now considered to be the time into the cache, and such is it the same transfer rate of t bytes per second (which is an assumption at the beginning of the question)? And, is the total time we are asked to compute the time to put it in the cache, or is there an extra factor we need to include to get it out of the cache? Also, what exactly is the role of the second cache in the OS?
Clarification: Yes. The interleave spacing is just enough to ensure that the head does not have to go all the way around. For (c) there is no extra overhead of caching you need to take into account. For (d) the cache reduces the DMA overhead.

Question 3

1) Are you asking what is the size of the page table for a single process? Does the page table have entries for every possible virtual address, or just the ones that have been allocated by the program so far?
Clarification: (a) I am asking the maximum size.

2) (a)For question 3, part A, The per-process size of the page table depends on the number of pages for the particular process, can we simply state that is is 4 * the number of pages being references, or do you want it to be the maximum page size? (b) Doesn't the size of the virtual memory space depend on the number of processes, since each one can address its own 2^32 bytes of memory? I was wondering because

the number of levels required is directly related to the size of the entire virtual memory space, which, I believe, depends on the number of running processes. Since we aren't told how many processes are running, then we can't decide how many levels are required, right?
Clarification: (a) I want the maximum page size (b) In this question you are required to optimize the page table for each process, assuming the max page table size. So, the page table is organized in a hierarchical way for each process.

3) Question 3: c) What exactly do you mean by "reaching" the desired memory position? For instance, with two levels, we would require one read for level one, one read for level 2, and then would you include the third read to fetch the memory at the desired location, or do you consider discovering the location "reaching" it?
Clarification: In the scenario you describe here, we require 3 memory accesses to reach the memory location.

4) Question 3d: what is the point of having an entry to tell which virtual address the particular physical address corresponds to? This would be useless, because then, each memory look up would have to linearly search through all 32,768 page table (one for each page in main memory) entries to find the correct virtual address. Do we need another table for which entry in the table corresponds to which virtual address?
Clarification: You do not another table. Think of adding more fields in the table.

5) Question 3c: Is there a trick to this, it seems too simple or I've misinterpreted it? Shouldn't it just be the number of levels times t.
Clarification: You are on the right track.

6) Question 3: How does reorganizing the page table reduce the memory requirement of the process? Does the page table's total physical size actually reduce, or does this allow us to just keep certain levels in memory and others on disk? Also, how are the pointers organized? Does the first level contain all page table data, except for one word, which contains a pointer to the next page, like a linked list? Or is it more like a tree, where each level contains multiple pointers? Does the first level contain only pointers, like the Unix filesystem's indirection scheme for i-nodes?
Clarification: Reorganizing the page table removes the contiguous memory restriction of the page table. Further, it allows us to keep part of the page table in disk. As for pointers for the hierarchical paging scheme, each pointer except the last one stores pointers. Think of it as a tree, where the leaves are page tables that point to the real memory locations, the second level pages the page table, the level above it pages the page table of the page table, and so on.

7) Question 4: Think of the MAC layer as the layer below the network layer, and above the physical layer. This is the link layer, and it is here that CSMA/CD is implemented and I want you to implement encryption at this level. The reason we want link level encryption is that we only want data confidentiality over the wireless link, and not end-to-end.

8) Question 3(d), what do you consider "sufficient"? Obviously, with just the two fields, we can perform a linear search to find the desired virtual memory address. So it's really slow, but it works. Are you looking for us to add other fields to make this scheme perform with constant-time lookup? Is the table ordered by the physical address field? Is there more than one correct answer?

Clarification: Can you really get your scheme to work with just two fields, i.e. the virtual and physical address? Think about it a bit more. Further, for part (f) I expect an answer that can be implemented practically! So, while ordering by physical addresses might be feasible, ordering by virtual addresses will not be a good answer. But is there any benefit of sorting by physical addresses? Think of alternative strategies. Your modification should give an answer better than linear, and you are allowed to use another table.

9) Are you saying that it would be okay to give a scheme in part (d) that has linear-time lookup? By physical page, you mean RAM, correct? So in our virtual memory table/physical page table, each physical page is guaranteed to have a mapping to a virtual memory location, but the converse is not true, right?
Clarification: Yes, for part (d) a linear time lookup is acceptable. By physical page I mean the RAM, and so each physical page mostly should have a mapping, but the reverse is not true. Note that the hints were for part (f).

Question 4

1) Question 4: What is a bootstrap procedure?
Clarification: Bootstrap procedure is the initialization routine. In this question, it is the initialization required for your algorithm.

2) Question 4: How in detail do we have to go about our cryptography algorithms? We were only given a brief overview of some symmetric cryptography and public/private key pairs. Algorithms like AES and DES were mentioned but never covered in detail. Can we just say that we could use these algorithms to encrypt our data, or do we need more details, like take our plaintext, how to convert it into cipher text, how to decrypt it back, etc?
Clarification: In this question, I DO NOT expect you to get into the details of the encryption algorithm. Give a high level overview of your solution given the kind of description we covered in class.

3) Question 4: Should we address the key transfer problem or can we just assume that the user and access point have the correct keys.
Clarification: You need to state this assumption in the bootstrapping part of your answer. I do not expect you to give details of the key exchange protocol since it was not covered in class.

4) Question 4: Are we designing an encryption scheme for all the wireless hosts to share (protecting against outside-the-network attackers), a scheme where nobody in the wireless network trusts anyone and they all encrypt their data differently, or a scheme where only one host is encrypting and everyone else is sending in the clear?
Clarification: Think of it as a real wireless network. For example, you are sitting in CTB with your laptop. You are opening a telnet connection on a machine in the department. Over the wireless network all your packets will go open in the air, and anyone sitting, say in Ruloffs, with a sniffer on his laptop can see all your packets. And in the case of telnet, he can see your password too! This question asks you to implement a link layer encryption mechanism, such that packets over the air cannot be seen by a person sitting with a sniffer around you. So, in the case of telnet the packet is sent encrypted from your machine to the access point.

5) I have a question concerning problem 4 on the Take Home Final. You ask us to "design two schemes" for encrypting over wireless. Are we allowed to use the encryption techniques discussed in class? Specifically, it seems like RSA and AES

can be used here, and they satisfy your requirement of one public key system and one symmetric system. Am I allowed to explain how they can be used here, or do I need to develop a cryptosystem other than what we talked about?
Clarification: You do not have to develop a new cryptosystem. You solution should use algorithms discussed in class, and only describe how to use them for this particular problem.

Question 5

1) Question 5: Can we assume the existence of a semaphore broadcast operation that atomically wakes up all threads waiting on a semaphore and sets the internal counter to zero?
Clarification: No. You can only assume the standard P() and V() operations on the semaphores.

2) Question 5: Can a costume designer be doing any work/having any problems if an alien is not there? IE: Can 3 costume designers be having troubles if none of the aliens show up? Does Garth go back to sleep when less than 3 costume designers are having trouble, or just when no costume designers are having trouble?
Clarification: Yes, costume designers could have problems even if the aliens are not there. Further, Garth does not wake up until 3 costume designers are having trouble.

3) Question 5: What does Garth do if he is awoken and six alien characters are present, but there are not three costume designers present?
Clarification: In this case Garth finishes the alien scene, and his process ends.

4) Question 5: Can we use monitors? It says to use semaphores, but you can implement monitors using semaphores and it seems like a hassle if you said no.
Clarification: No, you are not allowed to use monitors! The purpose of this question is to increase your expertise in implementing synchronization using semaphores.

5) After we have met the condition, either one, to wake up Garth do we really care about the threads for the aliens or the designers? For example if I simply have a global variable to keep track of the alien threads coming in and they increment that and then check if that is equal to 6 call Garth if not then it falls through. Is that incorrect? I was thinking that it would be ok b/c what would the threads for the aliens really do at this point and I would not want to send each of the six threads to Garth but simply create a single Garth thread once the condition is met.

```
void Aliens() {
        P(mutex_aliens);
        alien_count++;
        V(mutex_aliens);
        if(alien_count == 6) {
                Garth(true);
        }
}
```

Clarification: Yes we do! When does the alien in this code example shoot the movie? If the count is not 6 he just falls through, and so the processes are not correctly synchronized. So, an alien process should wait until 6 aliens have arrived. As for the costume designers, once their problems are solved, they should go back to doing their work.

6) What does “get done with movie” mean? If Garth wakes up and finds the aliens ready and the designers needing help he shoots the movie but does that mean he no longer cares about the designers or does he take care of them after he shoots?
Clarification: No, he does not take care of the designers. Once he is done shooting the movie, he ignores the rest of his staff and his process ends. He might as well go on a vacation.

7) Am I correct in assuming that Garth terminates when the scene is shot, the aliens can terminate when they have arrived, and the costume designers have to sit in a continuous loop?
Clarification: Yes, you are correct.

8) I've read the clarifications, and I'm still not sure what is supposed to happen in the following case: Garth wakes up and sees that there are not 6 aliens ready, but three costume designers are having trouble. So he helps one of the costume designers, and no new aliens come and no new costume designers have trouble. Now, does he go back to sleep, because there are only two costume designers who need help (not three), or does he help the other two costume designers, and only when he's finished helping all three costume designers does he re-evaluate whether or not he should go back to sleep?
Clarification: In this case, he helps all the 3 designers in an atomic operation. So, all the three costume designers are helped together. So, the situation you have described will not happen.

9) For question 5, will all costume designers have trouble, or will some exist (to satisfy the requirement of there being three of them) and others may randomly have problems? Do we have to code this random behavior of whether the designer has trouble? Also you stated in a clarification that if 6 aliens are present the movie is shot, but in the question it states that 6 aliens have to be present and 3 costume designers have to be present. Which is it?
Clarification: You do not have to code in the randomness. Further, it is an OR condition, that is Garth wakes up either if 6 aliens are present OR 3 costume designers are having problems. However, whenever 6 aliens are present, then even if 3 costume designers are having problems, Garth will shoot the alien scene and his process ends.
The costume_designer process looks as follows:

```
Do {
        // Designing costumes
        // am having problems
        … your code
        // problem solved
        … your code
} while true;
```

The alien process looks like:

```
        // was shooting in another movie
        // ready to go to Garth
        … your code
}
```

Question 6

1) Question 6: Does the restart the thread happen when there is a deadlock, therefore should the option read: detect deadlock and restart...? If not, then when would you restart the thread?
   Clarification: The thread is restarted if all its requested resources are not available and it needs to wait. We do not wait for deadlock to occur.

Question 7

1) Question 7: How do I know what reaction to packet loss is right?
   Clarification: Think in terms of router congestion. Packet loss is caused by congestion, and there is congestion in a network if there are too many packets. So, what is a good reaction to packet loss in this case? Similarly, why is there a packet loss in the case of wireless network? Is a reaction similar to network congestion a good (correct) reaction?

2) Question 4: I have thought up most of a solution to the problem posed in question 7, but I am worried about two things:
   a. Worst-case, the solution could place quite a computational load on the access point (in terms of memory especially).
   b. The solution violates the end-to-end principal.

   Should I worry too much about these things?
   Clarification: Theoretically you would. But in reality, and for purposes of this exam, I do not expect you to be too bothered by any of these issues.

3) Question 7: I am somewhat confused by the wording of part b. The way I read the question, we are asked to give techniques for reducing or solving the problems identified in part a. We may:
   a. sniff and parse all packets
   b. replay old packets
   c. construct and send new packets
   d. Am I correct in assuming that we may do all of these things? Are there other changes we may make that I did not list?

   Clarification: These are hints, and you may use any/all of them to construct the solution. You might not need anything else for a working solution. However, any other solution which reduces the problem will also get full points. Think about access points looking at transport layer packets.