# Chapter 5 – Operating systems

**1. Give an example of the use of physical separation for security in a computing environment.**

**2. Give an example of the use of temporal separation for security in a computing environment.**

**3. Give an example of an object whose sensitivity may change during execution.**

**4. Respond to the allegation “An operating system requires no protection for its executable code (in memory) because that code is a duplicate of code maintained on disk.”**

**5. Explain how a fence register is used for relocating a user’s program.**

**6. Can any number of concurrent processes be protected from one another by just one pair of base/bounds registers?**

**7. The discussion of base/bounds registers implies that program code is execute only and that data areas are read-write-only. Is this ever not the case? Explain your answer.**

**8. A design using tag bits presupposes that adjacent memory locations hold dissimilar things: a line of code, a piece of data, a line of code, two pieces of data, and so forth. Most programs do not look like that. How can tag bits be appropriate in a situation in which programs have the more conventional arrangement of code and data?**

**9. What are some other modes of access that users might want to apply to code or data, in addition to the common read, write, and execute permission?**

**10. If two users share access to a segment, they must do so by the same name. Must their protection rights to it be the same? Why or why not?**

**11. A problem with either segmented or paged address translation is timing. Suppose a user wants to read some data from an input device into memory. For efficiency during data transfer, often the actual memory address at which the data are to be placed is provided to an I/O device. The real address is passed so that time consuming**

**address translation does not have to be performed during a very fast data transfer. What security problems does this approach bring?**

**12. A directory is also an object to which access should be controlled. Why is it not appropriate to allow users to modify their own directories?**

**13. Why should the directory of one user not be generally accessible to other users (not even for read-only access)?**

**14. File access control relates largely to the secrecy dimension of security. What is the relationship between an access control matrix and the integrity of the objects to which access is being controlled?**

**15. One feature of a capability-based protection system is the ability of one process to transfer a copy of a capability to another process. Describe a situation in which one process should be able to transfer a capability to another.**

**16. Describe a mechanism by which an operating system can enforce limited transfer of capabilities. That is, process A might transfer a capability to process B, but A wants to prevent B from transferring the capability to any other processes. Your design should include a description of the activities to be performed by A and B, as well as the activities performed by and the information maintained by the operating system.**

**17. List two disadvantages of using physical separation in a computing system. List two disadvantages of using temporal separation in a computing system.**

**18. Explain why asynchronous I/O activity is a problem with many memory protection schemes, including base/bounds and paging. Suggest a solution to the problem.**

**19. Suggest an efficient scheme for maintaining a per-user protection scheme. That is, the system maintains one directory per user, and that directory lists all the objects to which the user is allowed access. Your design should address the needs of a system with 1000 users, of whom no more than 20 are active at any time. Each user has an average of 200 permitted objects; there are 50,000 total objects in the system.**

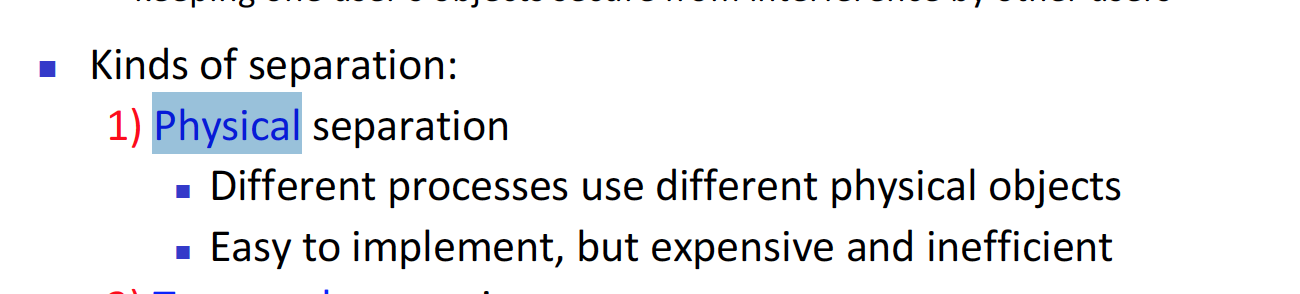
**20. A flaw in the protection system of many operating systems is argument passing. Often a common shared stack is used by all nested routines for arguments as well as for the remainder of the context of each calling process.**

**(a) Explain what vulnerabilities this flaw presents.**

**(b) Explain how the flaw can be controlled. The shared stack is still to be used for passing arguments and storing context.**

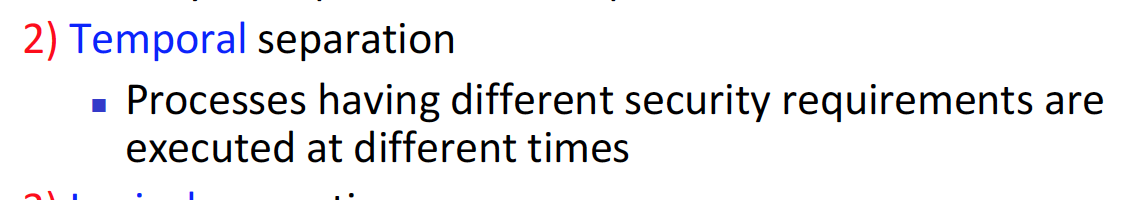
1. **Give an example of the use of physical separation for security in a computing environment.**

* One example could be literally using a separated hardware for multiple users, this way the computer components are separated. So there is no danger of sharing memory among multiple users.

**  
As seen above physical separation means using different process and physical objects it could mean using multiprocessors – since each process have their own process – and assuming they do not share memory with each other. Shareable I/O devices can be under this case.   
One example could be the printer – if only admins can use the printer then lower employee’s computer might not be even connected to the computer.**

1. **Give an example of the use of temporal separation for security in a computing environment.**

* Temporal separation is – LOL I forgot what temporal separation is. So, I will assume that is just memory separation. Then one user has each section of memory that is belonging to one user, and other users will have different section of the memory they can use.

**  
As seen above the temporal separation means that process is ran in different times. This could mean avoiding the race condition for processes. Or an example with this is when there are untrusted and trusted user in the network. The trusted user’s process will have more security req, and they will have priority over the untrusted user’s process.**

1. **Give an example of an object whose sensitivity may change during execution.**

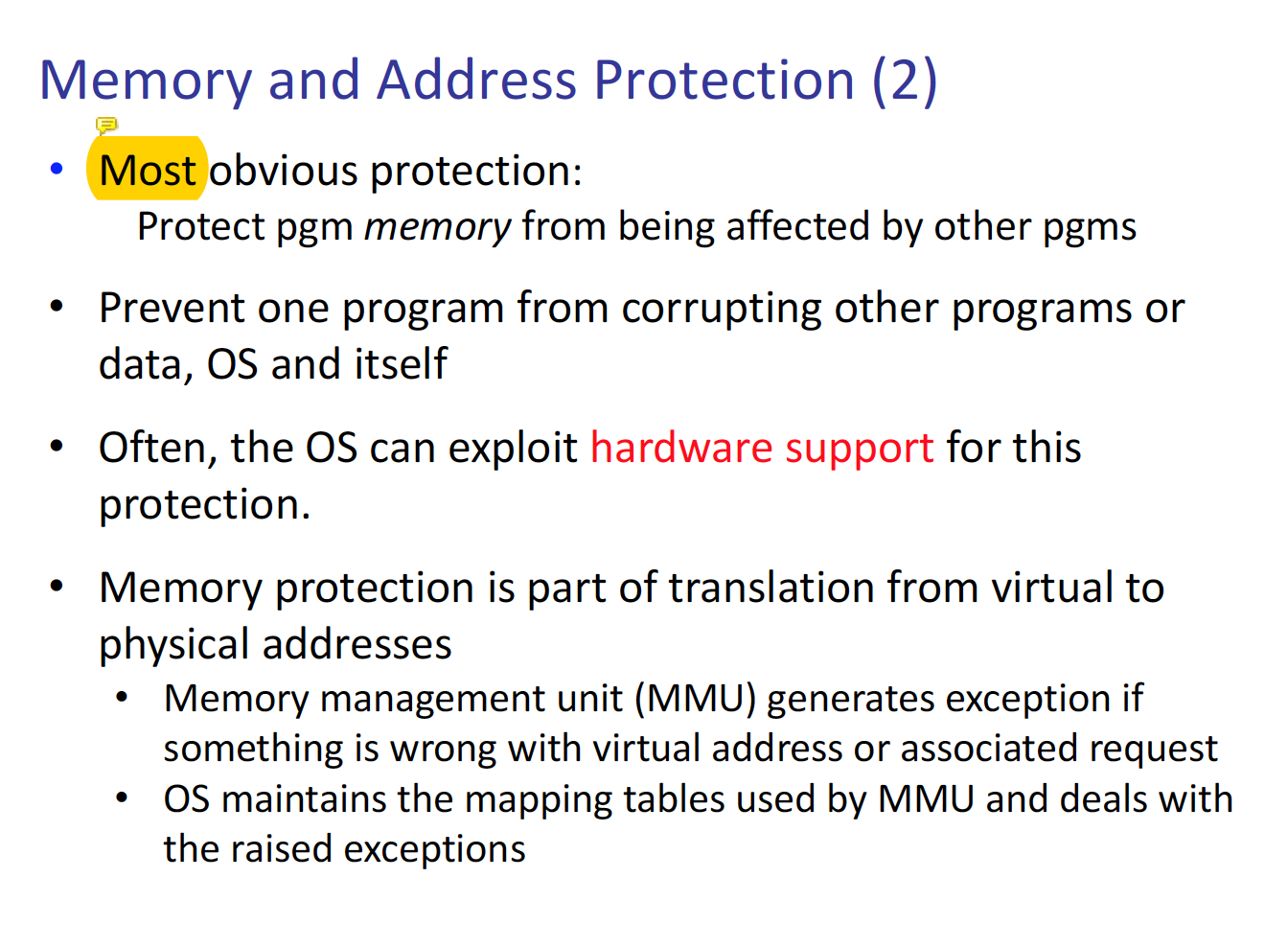
* Object whose sensitivity may change during execution – at the start of the execution, it is in user privilege, then once executed it moves to root privilege. As we saw in one of the labs. The ping command is perfect example, we can ping some IP address as a normal user, however the ping command needs to access the socket, there for need the root access to control the hardware.

**One of the possible answer of this question is encrypted data.**

**Let’s say that in a company, there exist a password for a vault. If this password exist in a plaintext format, then it is very sensitive since anyone can access the password. However, once the password in encrypted to cipher text it’s sensitivity goes down. Because, now only admins who have the key will be able to decrypt the cipher text.**

1. **Respond to the allegation “An operating system requires no protection for its executable code (in memory) because that code is a duplicate of code maintained on disk.”**

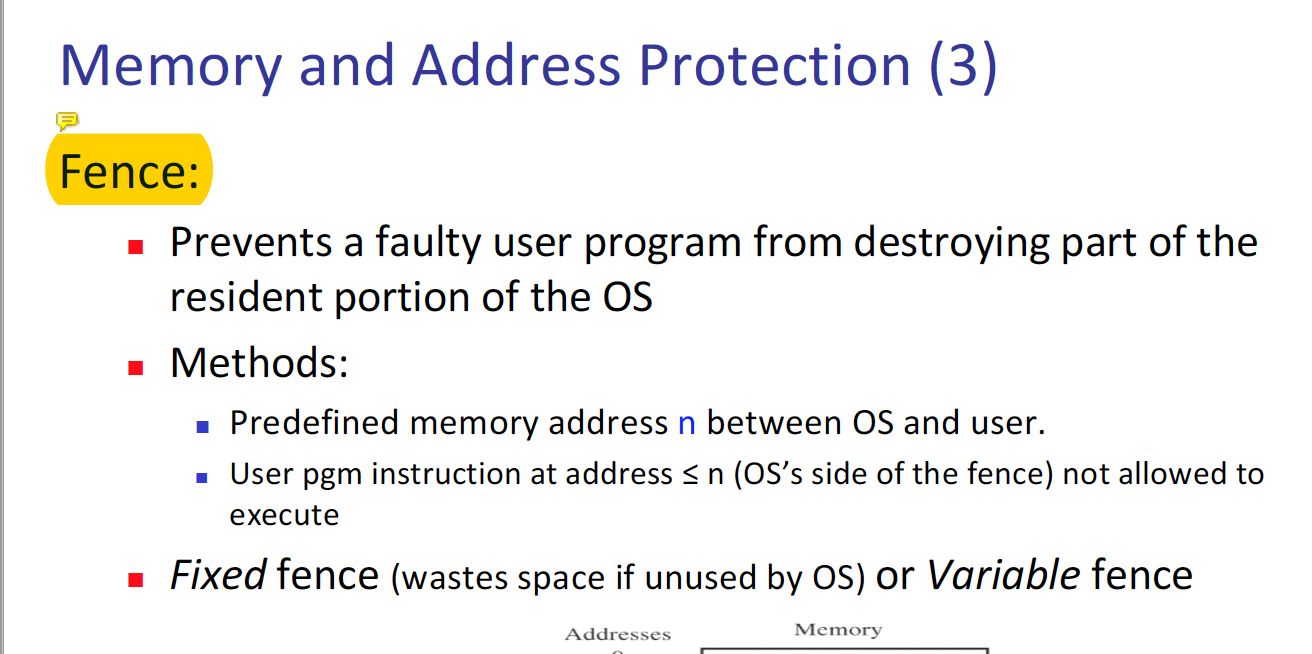
**- allegation : a claim or assertion that someone has done something illegal or wrong, typically one made without proof.**

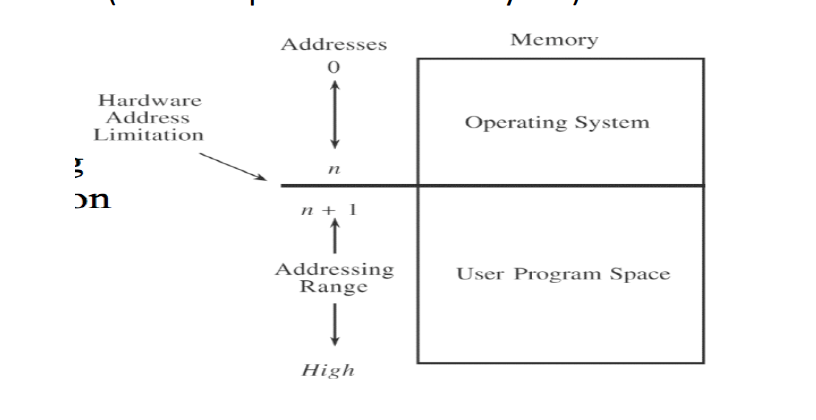
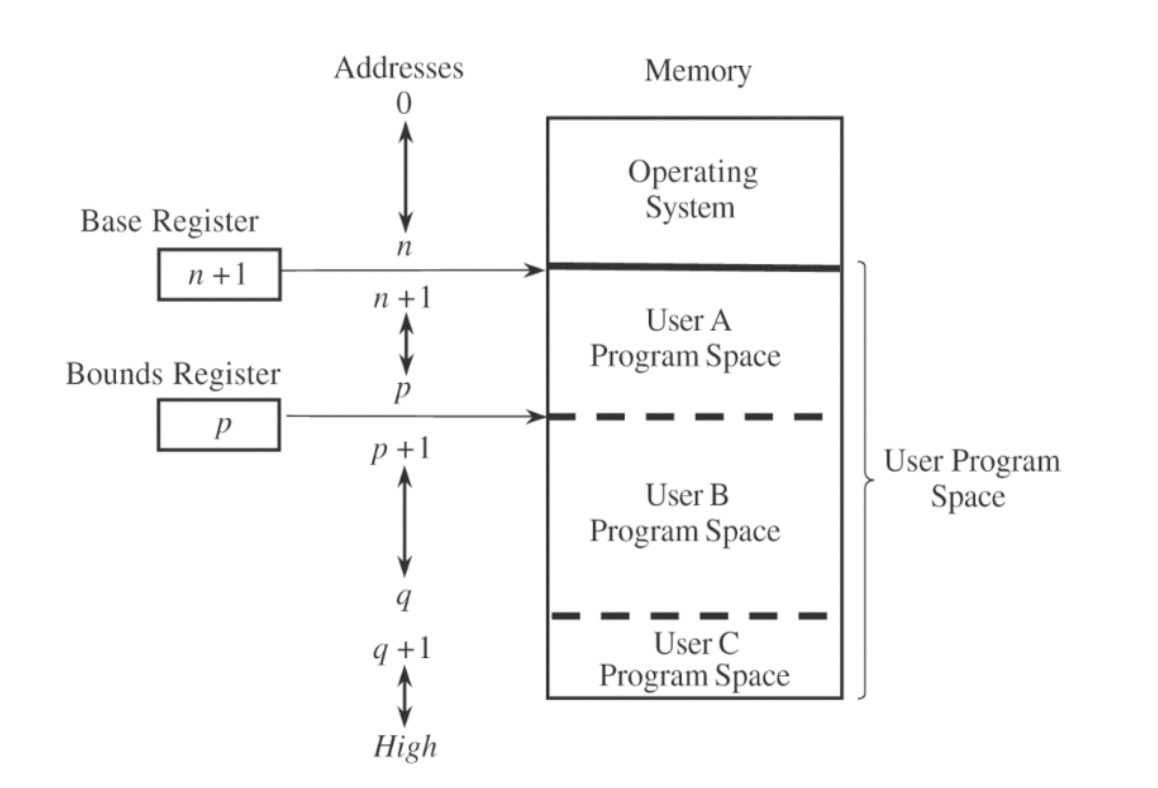
**-** So the claim is wrong, and we need to prove wrong, I guess that is what we needed supposed to do?   
Well operating system, controls the application program’s memory allocation, however I believe it needs to control itself too.  
As seen above in memory and address protection we can see, OS can corrupt the program. Therefore, it also needs to be controlled too.

**The correct answer for this question is that. Although the executable code is in a temporal memory, it can change the data in the disk memory. By causing buffer overflow or memory translation. So YES even the executable codes located in temporal memory needs to be controlled too.**

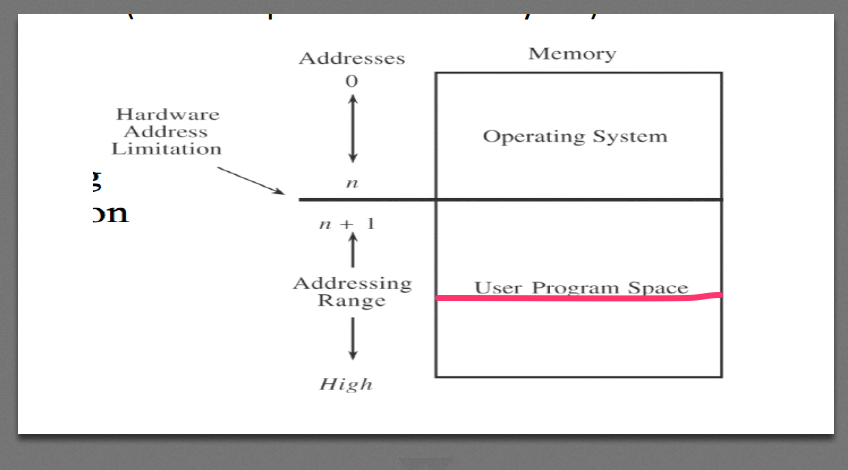
1. **Explain how a fence register is used for relocating a user’s program.**

* Fence is one of the memory and address protection method. In which there is a certain set of pre-allocated memory for the OS, n. And all the application programs cannot cross over that line

****So the user’s program will be relocated after the address n and have no choice but to be located after n, because of the fence.

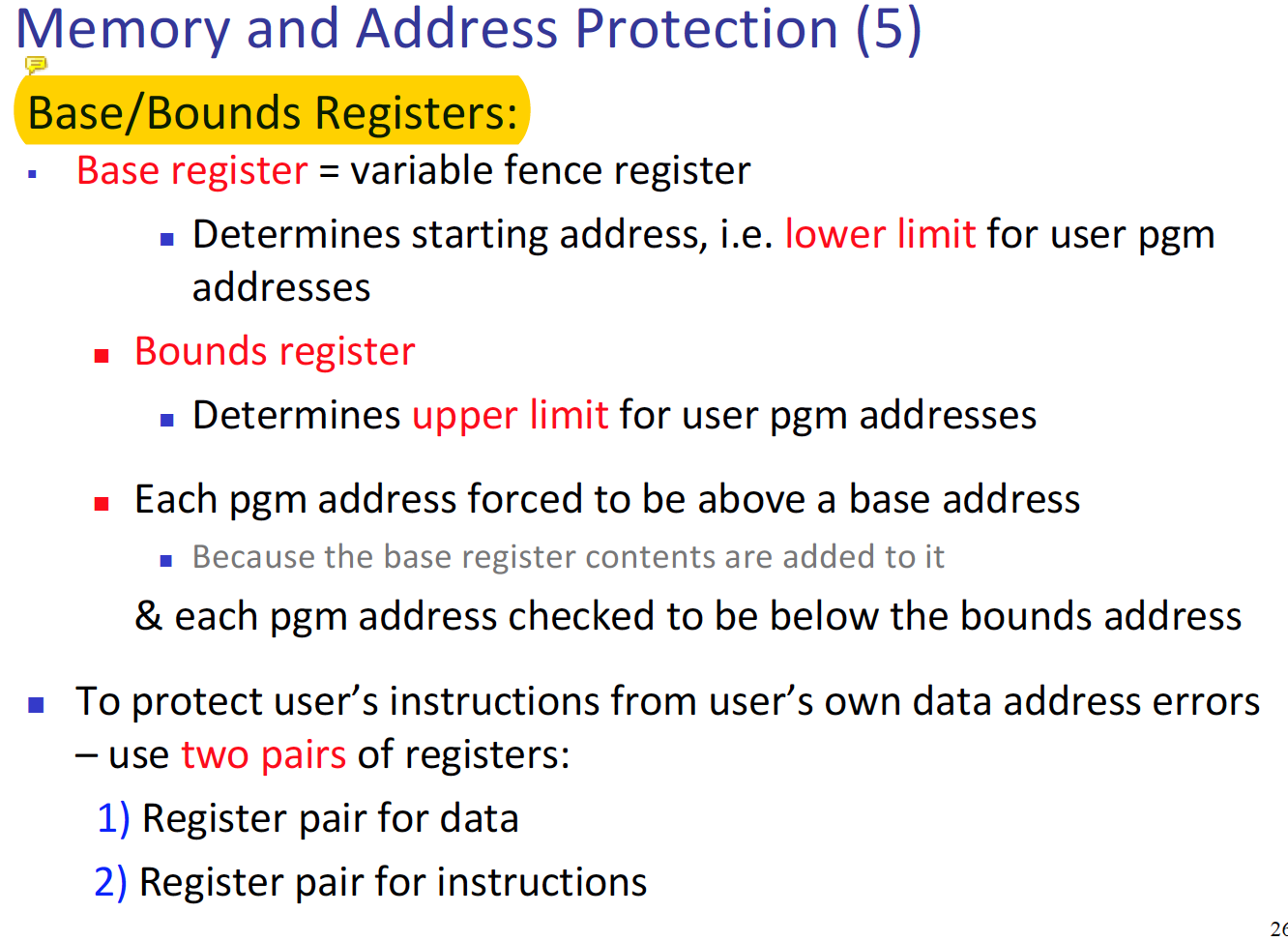
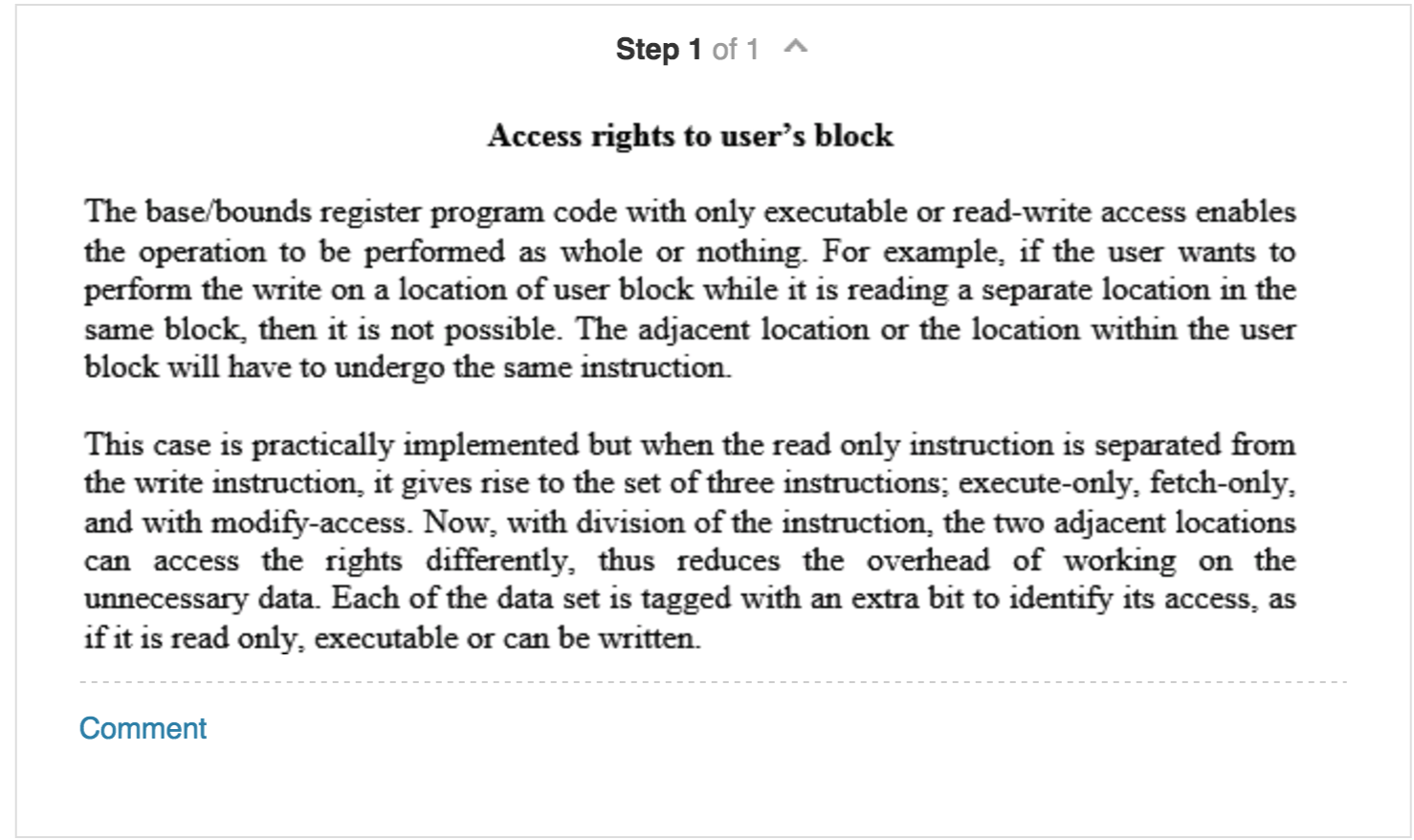
**The fence method does not relocate users program; it is relocation method that does that. We can assume two situation 1) OS is fixed 2)OS is dynamic   
In the first spot – it would be traditional fence method   
As seen above we already know that until n address it would be used by the OS, only after n we can perform relocation. Adding n to each users program.   
For dynamic allocation.  
There would exist an upper bound and lower bound for each user. And we can say that the addresses needed for OS is dynamic.**

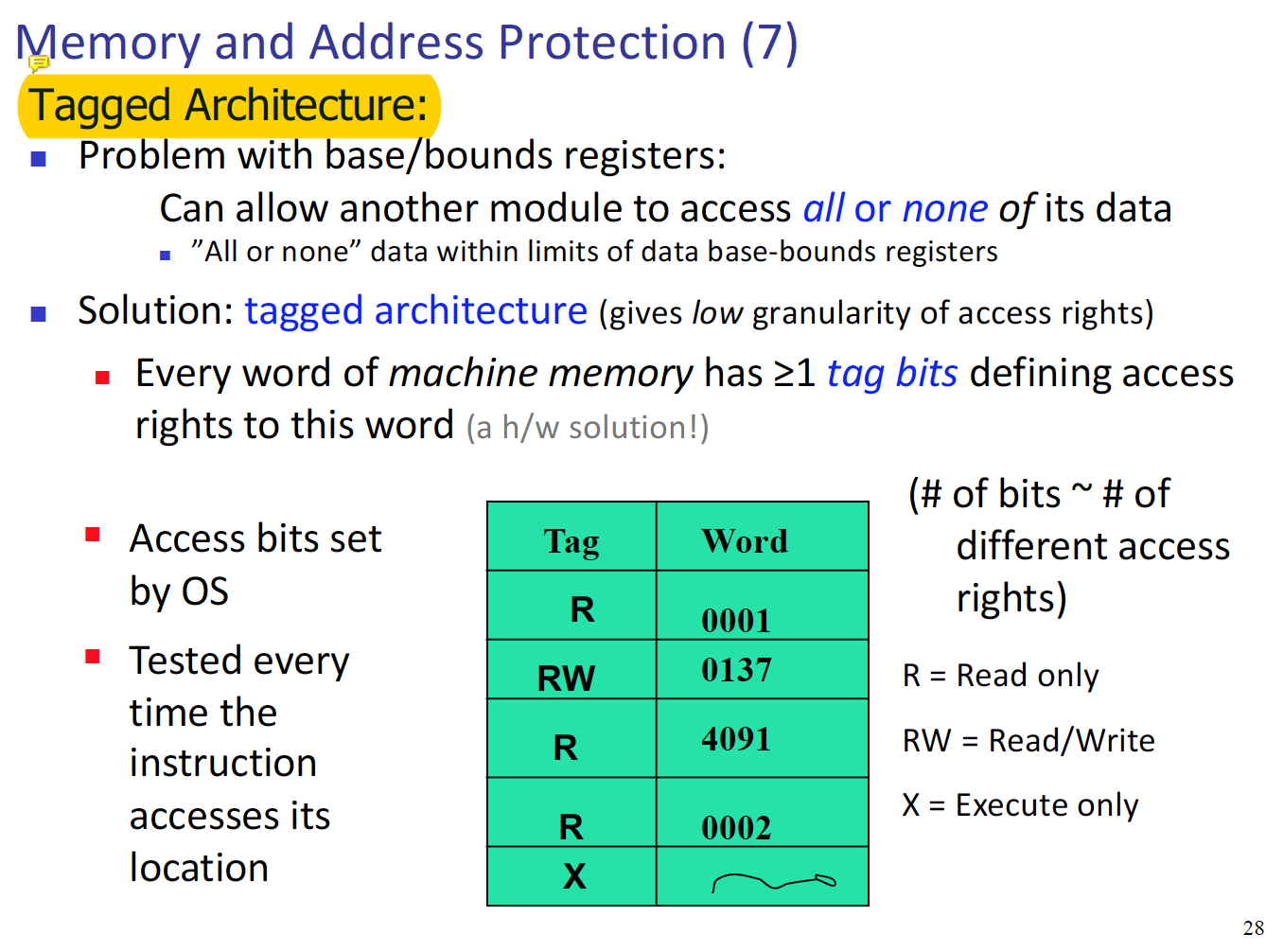
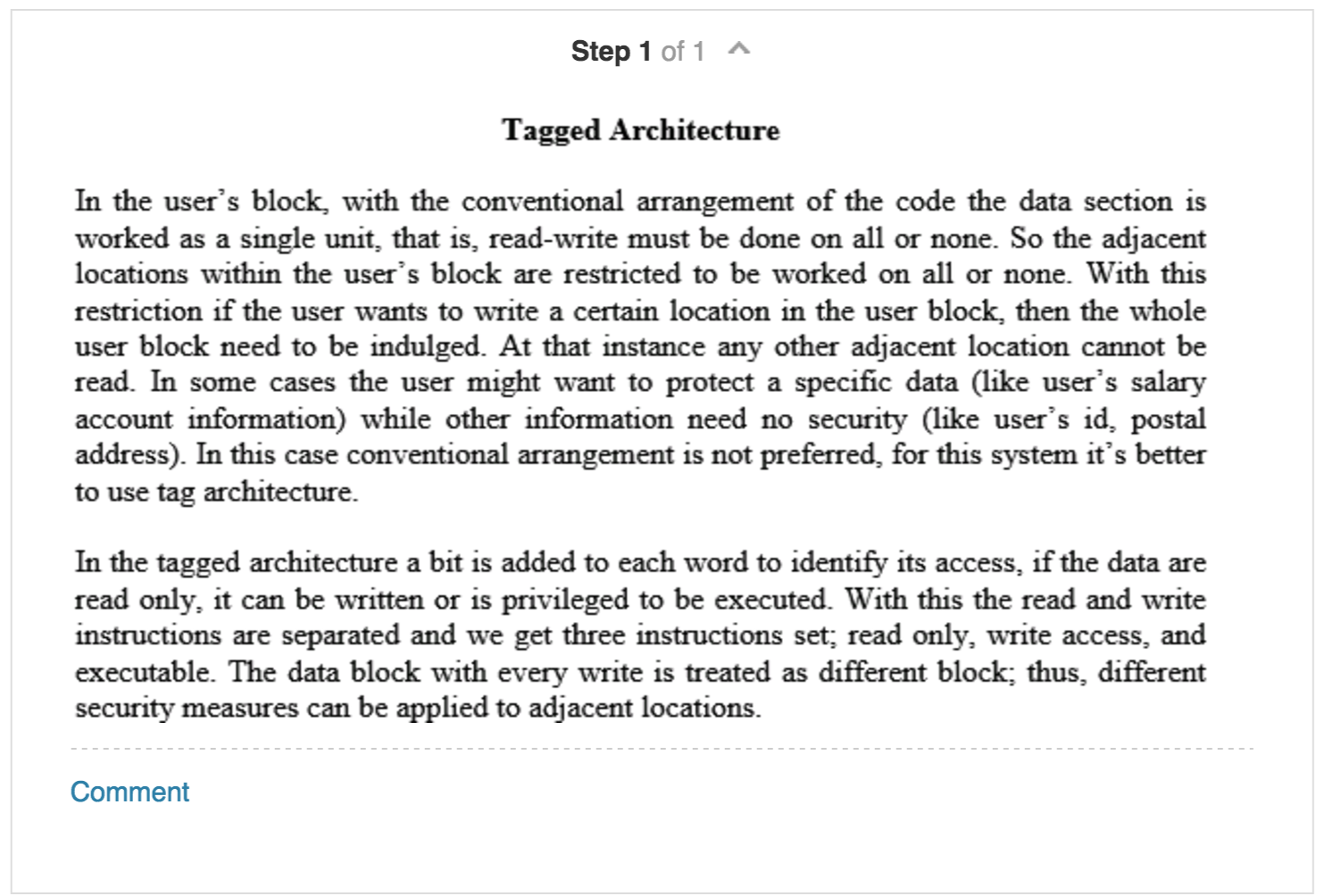
1. **Can any number of concurrent processes be protected from one another by just one pair of base/bounds registers?**

* I think this is a clear no, since one pair of base/bound register means there is one Fence (until n – which is for the OS) and middle part where one process can run and last part where the second process can run. So in total only 2 process can run with one pair of base/bound registers. (something like this)

**Since there is only one pair or base and bound register the maximum number of users we can separate is two. One can be located at the middle section and the other can be located at the bottom.**

1. **The discussion of base/bounds registers implies that program code is execute only and that data areas are read-write-only. Is this ever not the case? Explain your answer.**

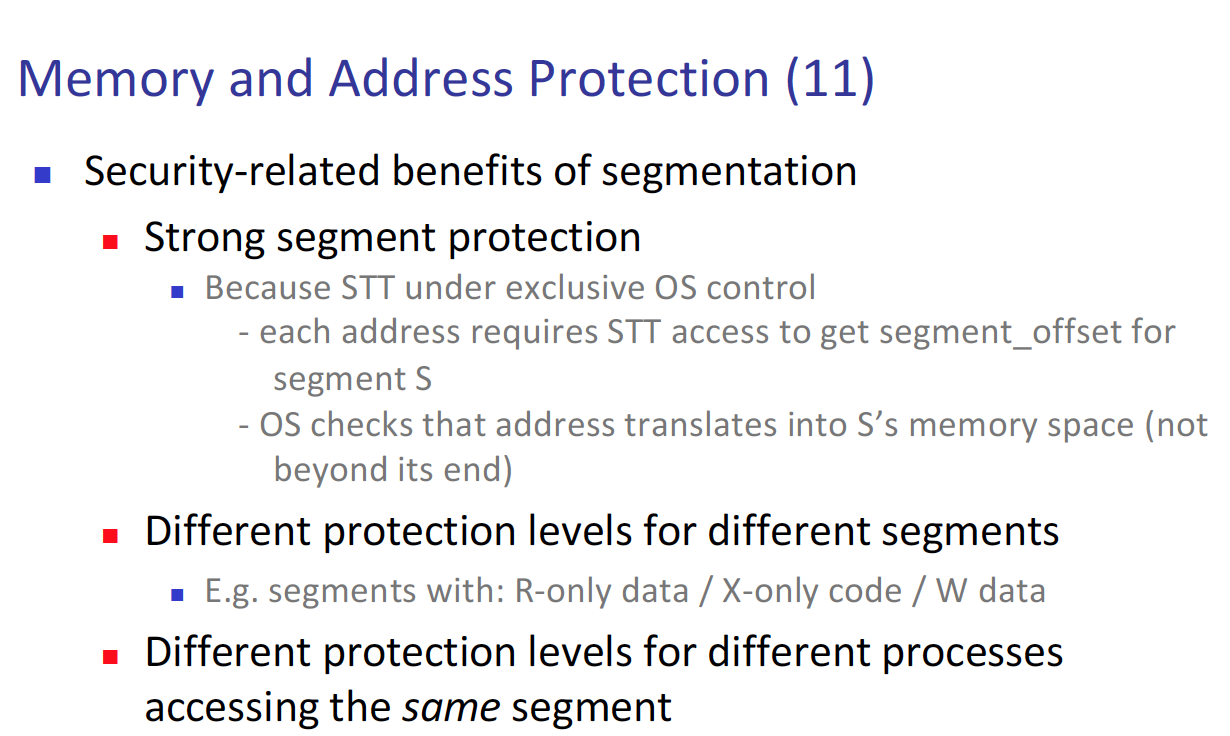
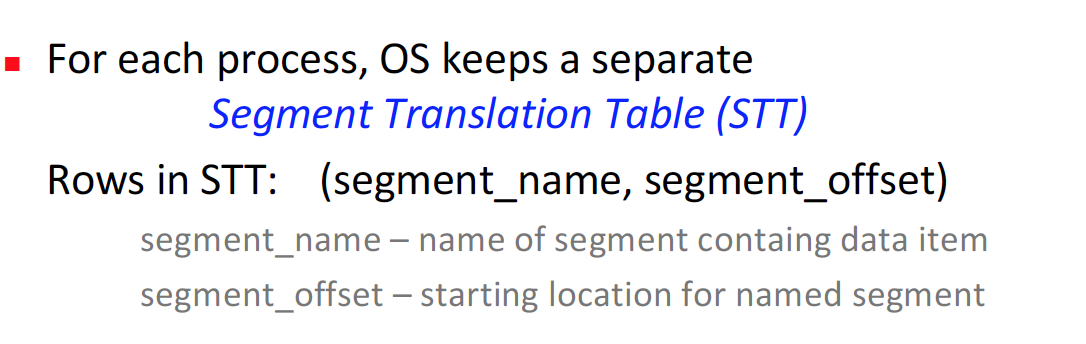
**-** So in other words between the section from if a user executes a program, and that program generates data – that would be read and write only program? That is not a case when the user executes a program, in which creates another program, then it is not read-write-only anymore.  
Even in the PPT there is no mention about read-write-only.

**8. A design using tag bits presupposes that adjacent memory locations hold dissimilar things: a line of code, a piece of data, a line of code, two pieces of data, and so forth. Most programs do not look like that. How can tag bits be appropriate in a situation in which programs have the more conventional arrangement of code and data?**This is how the tagged solution looks like, so in the question. The tagging system will need codes that are strangely arranged. The only way I can think of for a tagging system to have a proper rearrangement on traditional code is to have a system, in which the original can be reassembled.

**9. What are some other modes of access that users might want to apply to code or data, in addition to the common read, write, and execute permission?**

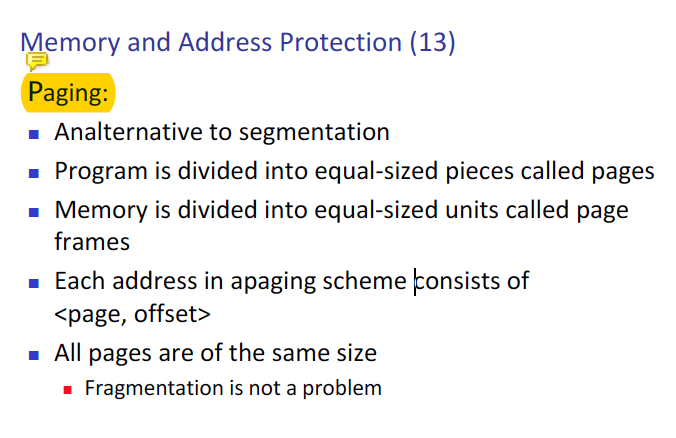
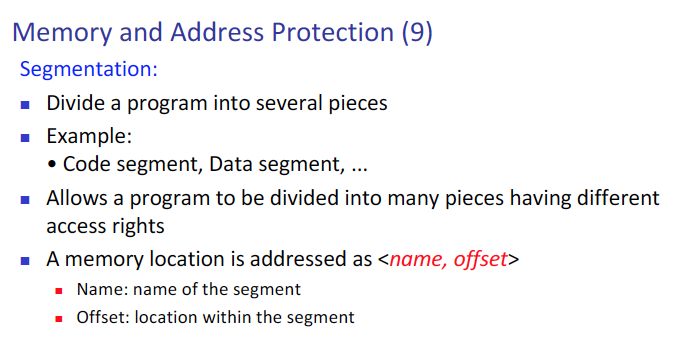
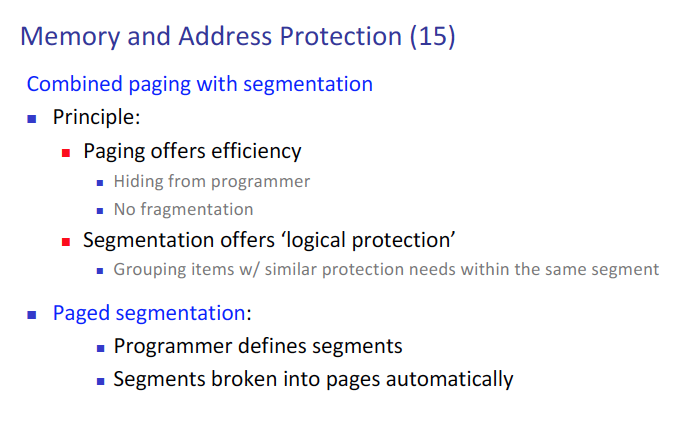
- This is a great question. For me when coding in c, I hate the fact that we need to first compile the code and then run it. It would be nice to have a mode of access, where the c file get automatically complied.   
  
**Some of other possible answers on this one, is the  
1. See the ownership of the file – yes a user might not have the access to view the file, but if there is a way for the normal use get hold of the root user, then the normal user can shoot an email asking for a grant access.  
  
2. Ownership – along to read, write and execute it would be much easier for a user to have ability to have an access mode to give owner ship to another user, in that way. Granting user access would be much more simplier.**

**10. If two users share access to a segment, they must do so by the same name. Must their protection rights to it be the same? Why or why not?**

- No they can be different, one of the benefit of using segmentation is that it is capable of handling complex access rights. Different users can share the same segment but the protection level for each can be different.  
**More concrete reason why different users can have different protection level is because of Segment Translate Table (STT)  
A user must access this table with the same name, however each user is associated with different rights in the table, and this is managed by the OS. And that is the reason why, each user can have different protection rights when accessing the same segment.**

**11. A problem with either segmented or paged address translation is timing. Suppose a user wants to read some data from an input device into memory. For efficiency during data transfer, often the actual memory address at which the data are to be placed is provided to an I/O device. The real address is passed so that time consuming**

**address translation does not have to be performed during a very fast data transfer. What security problems does this approach bring?**

* First let’s review what is paging and what is segment are.   
  it seems like, both method are, something that divides a program and store them in separate places. But one is dividing in equal amount, and the other is not. Also, it is a good idea to look at what it looks like to combine both of them.  
  Good, now we understand what paging and segmentation are, so when the actual address is given. One problem that I can think of is, if the attacker gets the address, then they might be able to harm the program directly. Since they will know the actual address of the program. Rather having to translate.

The answers are similar. If an IO device such as printer can access the program address directly, they might corrupt the data, if an attacker hacks the printer, and somehow that printer can corrupt the program. Or another problem might be buffer overflow, the IO device can overflow the memory size.

**12. A directory is also an object to which access should be controlled. Why is it not appropriate to allow users to modify their own directories?**

**-** One reason that I can think of is because, the user them self can change the root access directory to normal user’s access level? – (This is actually stupid since the question said ‘own directories’).  
Another reason maybe the fact that if an attacker is a user, and they create some kind of virus in their directory. And set the access, so only they can access that directory. Then the root user cannot do anything about it. So that hardware is gone for sure.

**The answer was something similar – if a user is allow to change access to their own directory, then they might change the rights provided by the directory holders, which will create conflict in rights. And that is very bad news LOL**

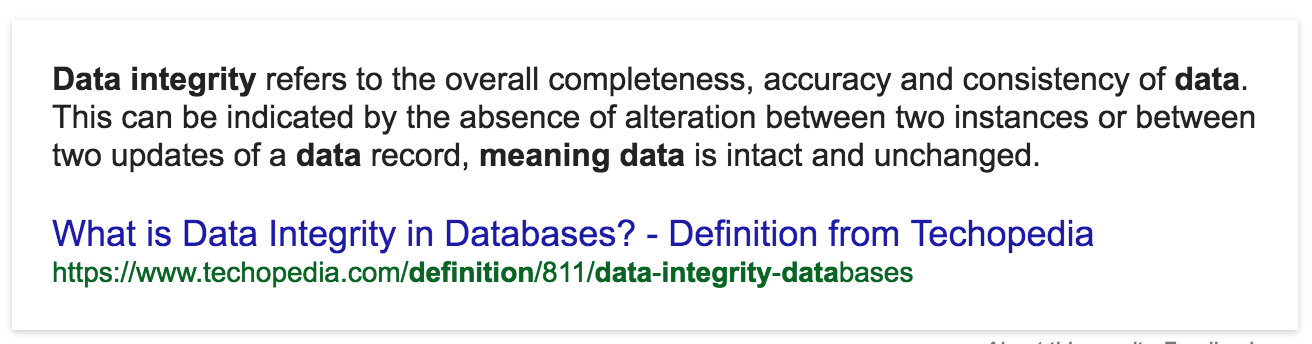
**13. Why should the directory of one user not be generally accessible to other users (not even for read-only access)?**

**-** Generally speaking, that directory is their private data. We do not go into other people home for no reason. It does not make sense to go into someone’s home without no permission.

In other words, to keep the minimal confidentiality, directory of one user should not be granted access to another users’ directory.

Main concern is passing chain of rights. If user A have access right to file F and user B access that file, just because they can. And give access permission to user C. Now if user A wants to delete access permission from user B, then there arises conflict. Since User C’s permission access is not known to user A.

**14. File access control relates largely to the secrecy dimension of security. What is the relationship between an access control matrix and the integrity of the objects to which access is being controlled?**

**-** First let us review what is integrity is. Integrity – only trusted users are permission to change the data.  
Now, access control is way to control, who access what with what access level. Then the relationship between access control and data integrity can be defined as if the access level is read only, there is nothing to worry about since even the users who can access the data are only able to read the data.

Access control list can acts as a mean of forcing data integrity. The main function of access control list would be prevented of data change from unauth user.

**15. One feature of a capability-based protection system is the ability of one process to transfer a copy of a capability to another process. Describe a situation in which one process should be able to transfer a capability to another.**

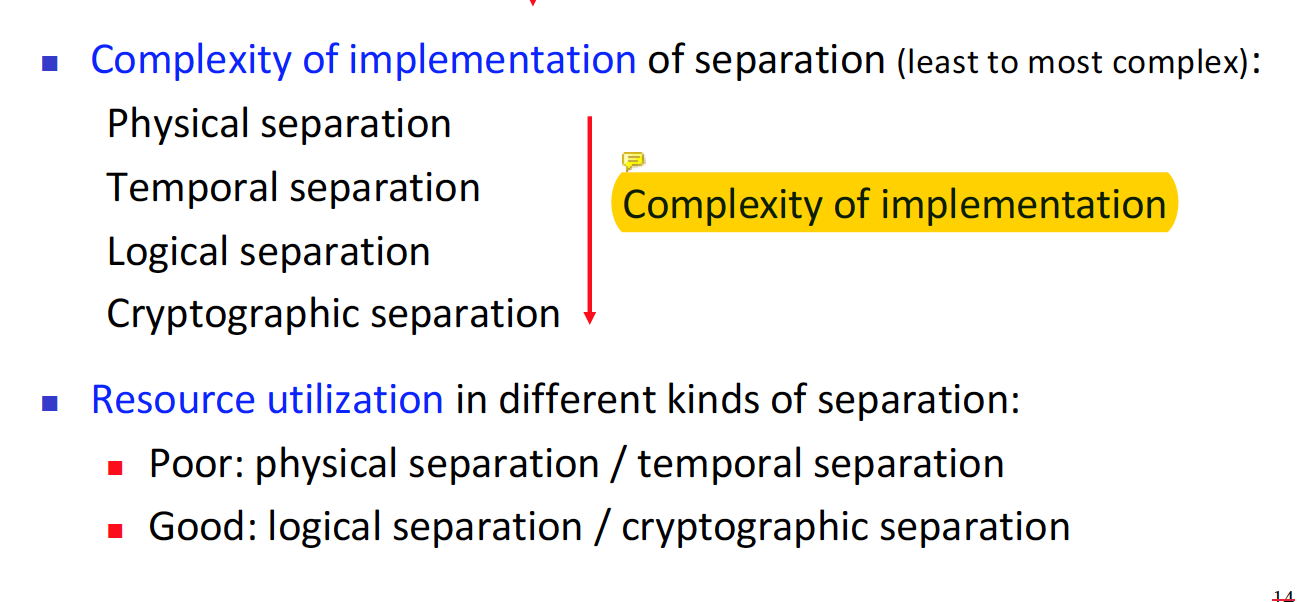
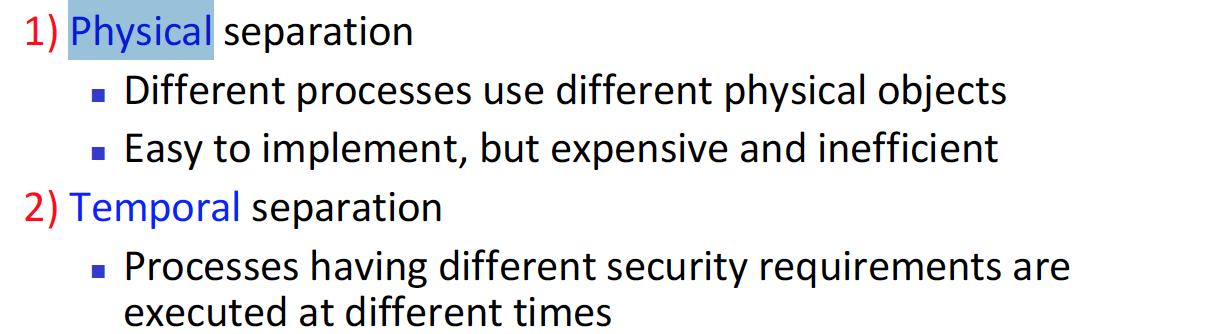
- Capability list, this is user oriented control access list. So one user can give access right to another user, this is dangerous but some situation where this is beneficial might be when, time is running OUT! Is the root user, or the super user is out of town, so the file permission cannot be granted via the super user. It is much faster if the file owner can give access to other users.

If process A is using a printer and grants permission to process B. The acceptable time for A to transfer permission to process B is when it is not using the printer. Something like this?

**16. Describe a mechanism by which an operating system can enforce limited transfer of capabilities. That is, process A might transfer a capability to process B, but A wants to prevent B from transferring the capability to any other processes. Your design should include a description of the activities to be performed by A and B, as well as the activities performed by and the information maintained by the operating system.**

**-** Skipped, since not very plausible to be on the final.

**17. List two disadvantages of using physical separation in a computing system. List two disadvantages of using temporal separation in a computing system.**

**-** One of disadvantage of physical is expense, and another can be uncomfortable of use.  
We can already see here, that resource utilization is poor for physical separation.  
Now lets move to temporal – first it is complex to implement. And second level of security is lower than physical.

Physical 1) execution time is longer 2)decrease utilization   
temporal 1) If one user is facing crucial condition, have to wait until another user is done with they task. 2) Some OS is not compatible with how temporal separation works.

**18. Explain why asynchronous I/O activity is a problem with many memory protection schemes, including base/bounds and paging. Suggest a solution to the problem.**

**-**

**19. Suggest an efficient scheme for maintaining a per-user protection scheme. That is, the system maintains one directory per user, and that directory lists all the objects to which the user is allowed access. Your design should address the needs of a system with 1000 users, of whom no more than 20 are active at any time. Each user has an average of 200 permitted objects; there are 50,000 total objects in the system.**

**- Skipped, not expected in final.**

**20. A flaw in the protection system of many operating systems is argument passing. Often a common shared stack is used by all nested routines for arguments as well as for the remainder of the context of each calling process.**

**(a) Explain what vulnerabilities this flaw presents.**

**(b) Explain how the flaw can be controlled. The shared stack is still to be used for passing arguments and storing context.**