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 executeonly**

**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 timeconsuming**

**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.**