Instructions: Answer all six questions. **Computer Sciences @ Wisconsin – OS Qualifier: Fall 2013**

1. Nucleus

Nucleus was an early example of an extensible system in which a hierarchy of internal and external processes could be created and controlled.

(a) The designers of Nucleus rejected semaphores as a basic form of synchronization between cooperating processes. Why were semaphores not a good match for Nucleus?

(b) Nucleus instead uses messages as the basic means of process communication. Specifically, Nucleus provides four related primitives: send_message, wait_message, send_answer, and wait_answer. Briefly describe how a "client" process that wishes to read from disk might interact with a "server" process responsible for performing the I/O. You may show pseudo-code to make the interaction clearer.

(c) What are some of the advantages of using messages for synchronization and communication in this environment?

(d) What new complexities are raised when using messages for synchronization? How did Nucleus ensure that new problems would not occur?

2. Disk Scheduling

Modern disk drives have an internal scheduler that reorders and merges requests to improve disk efficiency. In this question, we'll seek to understand this scheduling behavior in more detail.

(a) Disk scheduling exists because of the nature of how disk requests are performed, and their overheads. Describe the typical overheads encountered during a single disk request.

(b) Of all the overheads encountered in servicing disk requests, only some are the focus of modern disk schedulers. Which overheads can a disk scheduler help minimize? Which costs are not reduced by scheduling?

(c) Simple schedulers like shortest-seek-time-first (SSTF) optimize disk performance, but not perfectly. Describe how shortest-seek-time-first works, and what its limitations are. What kind of workloads are not particularly well served by SSTF?

(d) Given that much of the storage market is moving to Flash-based SSDs, which generally support random accesses quite well, is there still a role for in-device scheduling?

3. Virtual Memory

Virtual memory support was designed in older systems, with much smaller physical memories than exist today. In this question, we'll examine whether the old designs still make sense for modern systems.

(a) One important aspect of a modern paged virtual memory system is page size. What are current page sizes? Why might modern systems wish for different size pages than older systems?

(b) Another important aspect of paged virtual memory systems is the page table structure used to support virtualization. Very early systems used linear page tables (i.e., arrays). Describe why linear page tables are not appropriate for modern systems.

(c) To address the limitations of linear page tables, more modern designs introduced multi-level page tables. Describe how such an approach works; as address spaces get larger, does the structure of a multi-level page table need to change?

(d) A third aspect fundamental to traditional paged virtual memory is its ability to swap pages back and forth from disk. In modern systems, with large amounts of physical memory available, do VM systems still need to swap pages to disk? Why or why not?

4. **Authentication**

A common facility needed in computer networks is means for one party to set up a secure, authenticated communications channel with another.

(a) The Needham and Schroeder key distribution algorithm allows a process to get credentials from a Key Distribution Center, which it can then use to establish a connection with another process on the network. Describe how this algorithm works for the private key (symmetric) case.

(b) The Kerberos authentication system is based on the Needham/Schroeder scheme. An additional feature of Kerberos is that the credentials contain IP network addresses, and part of the verification of credentials involves checking that packets come from where they claim to come from. One drawback to this design decision is that it prevents Kerberos "tickets" from being used as capabilities. Explain why.

(c) Kerberos made an important improvement over Needham and Schroeder's protocol in the first message exchange. This improvement eliminated a possible replay attack. Describe the weakness in Needham and Schroeder and how Kerberos fixed it.

5. **Bayou**

This question is about the Bayou protocol.

(a) What is eventual consistency?

(b) The basic anti-entropy protocol in Bayou helps servers to agree on the set of writes that have been conducted. Briefly describe this basic anti-entropy protocol in Bayou.

(c) Describe an extension to the basic anti-entropy protocol in Bayou that can provide eventual consistency.

6. **Resource Location**

Locating resources is an important task in distributed systems.

(a) GrapeVine assigns several servers to host each individual user's mail box. How does a GrapeVine message server locate the server that contains the mail box of a particular user Alice.CS?

(b) Dynamo is a key-value storage system, where every key-value pair is stored on multiple nodes. How does a coordinator node in Dynamo locate the nodes that handle a particular read or write operation?

(c) The Google File System (GFS) stores file chunks on chunkservers. How does GFS locate the chunkservers that are holding a particular file chunk?

(d) Please compare the reliability and performance impact of the resource-location schemes used by the above three systems.