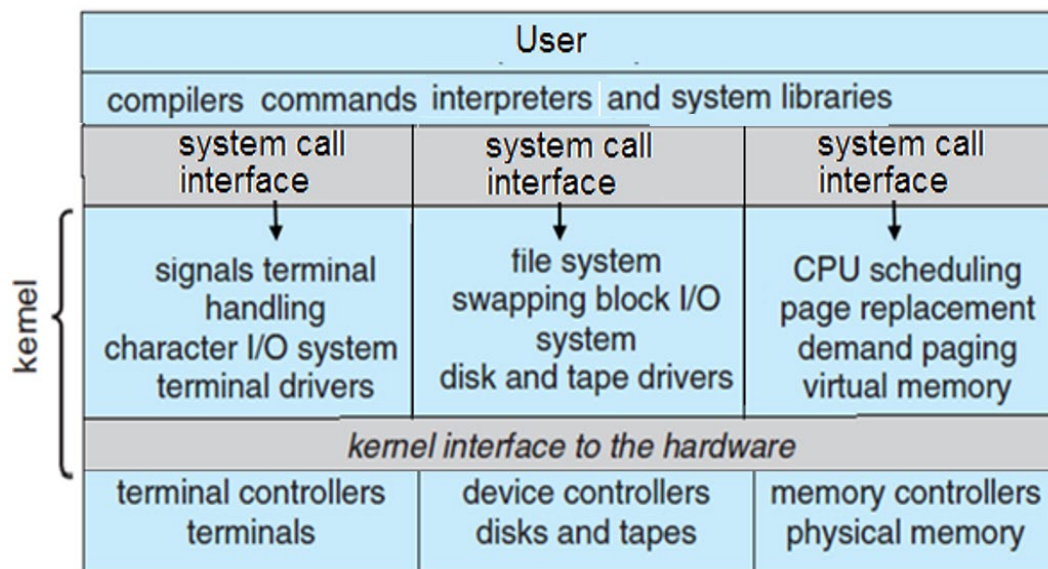## Week2 Questions

1. What is the purpose of system calls?

2. What are the five major activities of an operating system for process management?

3. What are the three major activities of an operating system for memory management?

4. What is the purpose of the command interpreter? Why is it usually separate from the kernel?

5. A command interpreter or shell must execute what the system calls to start a new process.

6. What is the purpose of system programs?

7. Answer the following based on the OS architecture shown in the **Figure** below:
   7.1. Identify the OS architecture.
   7.2. What are the advantages and disadvantages of this architecture?

User		
compilers commands interpreters and system libraries		
system call interface	system call interface	system call interface
↓	↓	↓
signals terminal handling character I/O system terminal drivers	file system swapping block I/O system disk and tape drivers	CPU scheduling page replacement demand paging virtual memory
kernel interface to the hardware		
terminal controllers terminals	device controllers disks and tapes	memory controllers physical memory

(kernel spans the rows from the system call interface down through the kernel interface to the hardware)

8. Answer the following based on the OS architecture shown in the **Figure** below:

   8.1 Identify the OS architecture.
   8.2 What are the advantage and disadvantages of this method?
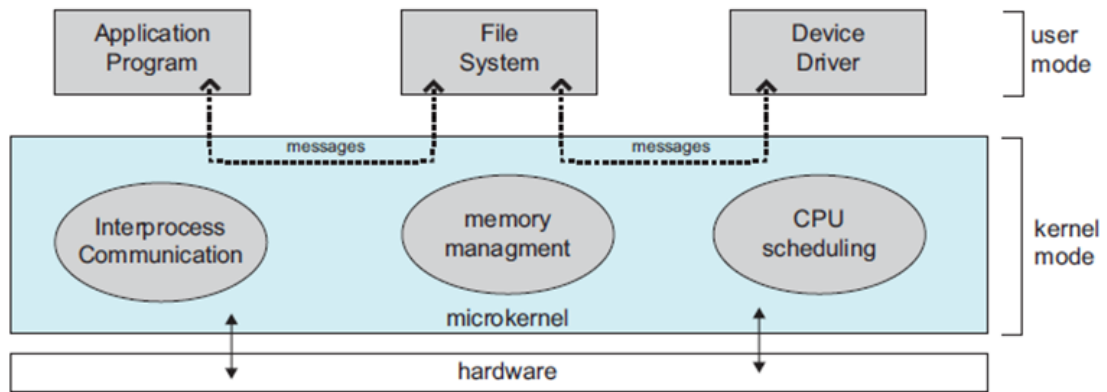
9. What is the main advantage of the layered approach to system design?

10. List **five services** an operating system provides and explains how each creates user convenience. In which cases would it be impossible for user-level programs to provide these services? Explain your answer.

11. What are the two models of inter-process communication (IPC)? What are the strengths and weaknesses of the two approaches?

12. What are the advantages of using loadable kernel modules?

13. How could a system be designed to allow a choice of operating systems from which to boot? What would the bootstrap program need to do?

14. Would it be possible for the user to develop a new command interpreter using the system-call interface provided by the operating system?