Introduction

1. What are the two main functions of an operating system?

Ans; An operating system must provide the users with an extended machine, and it must manage the I/O devices and other system resources. To some extent, these are different functions.

1. In Section 1.4, nine different types of operating systems are described. Give a list of applications for each of these systems (one per operating systems type).

Ans: Obviously, there are a lot of possible answers. Here are some. Mainframe operating system: Claims processing in an insurance company. Server operating system: Speech-to-text conversion service for Siri. Multiprocessor operating system: Video editing and rendering. Personal computer operating system: Word processing application. Handheld computer operating system: Context-aware recommendation system. Embedded operating system: Programming a DVD recorder for recording TV. Sensor-node operating system: Monitoring temperature in a wilderness area. Real-time operating system: Air traffic control system. Smart-card operating system: Electronic payment.

5. On early computers, every byte of data read or written was handled by the CPU (i.e., there was no DMA). What implications does this have for multiprogramming?

Ans: The prime reason for multiprogramming is to give the CPU something to do while waiting for I/O to complete. If there is no DMA, the CPU is fully occupied doing I/O, so there is nothing to be gained (at least in terms of CPU utilization) by multiprogramming. No matter how much I/O a program does, the CPU will be 100% busy. This of course assumes the major delay is the wait while data are copied. A CPU could do other work if the I/O were slow for other reasons (arriving on a serial line, for instance).

6. Instructions related to accessing I/O devices are typically privileged instructions, that is, they can be executed in kernel mode but not in user mode. Give a reason why these instructions are privileged.

Ans: Access to I/O devices (e.g., a printer) is typically restricted for different users. Some users may be allowed to print as many pages as they like, some users may not be allowed to print at all, while some users may be limited to printing only a certain number of pages. These restrictions are set by system administrators based on some policies. Such policies need to be enforced so that userlevel programs cannot interfere with them.

9. There are several design goals in building an operating system, for example, resource utilization, timeliness, robustness, and so on. Give an example of two design goals that may contradict one another.

Ans: Consider fairness and real time. Fairness requires that each process be allocated its resources in a fair way, with no process getting more than its fair share. On the other hand, real time requires that resources be allocated based on the times when different processes must complete their execution. A realtime process may get a disproportionate share of the resources.

10. What is the difference between kernel and user mode? Explain how having two distinct modes aids in designing an operating system.

Ans: Most modern CPUs provide two modes of execution: kernel mode and user mode. The CPU can execute every instruction in its instruction set and use ev ery feature of the hardware when executing in kernel mode. However, it can execute only a subset of instructions and use only subset of features when executing in the user mode. Having two modes allows designers to run user programs in user mode and thus deny them access to critical instructions.

16. When a user program makes a system call to read or write a disk file, it provides an indication of which file it wants, a pointer to the data buffer, and the count. Control is then transferred to the operating system, which calls the appropriate driver. Suppose that the driver starts the disk and terminates until an interrupt occurs. In the case of reading from the disk, obviously the caller will have to be blocked (because there are no data for it). What about the case of writing to the disk? Need the caller be blocked awaiting completion of the disk transfer?

Ans: Maybe. If the caller gets control back and immediately overwrites the data, when the write finally occurs, the wrong data will be written. However, if the driver first copies the data to a private buffer before returning, then the caller can be allowed to continue immediately. Another possibility is to allow the caller to continue and give it a signal when the buffer may be reused, but this is tricky and error prone.

17. What is a trap instruction? Explain its use in operating systems.

Ans: A trap instruction switches the execution mode of a CPU from the user mode to the kernel mode. This instruction allows a user program to invoke functions in the operating system kernel.

18. Why is the process table needed in a timesharing system? Is it also needed in personal computer systems running UNIX or Windows with a single user?

Ans: The process table is needed to store the state of a process that is currently suspended, either ready or blocked. Modern personal computer systems have dozens of processes running even when the user is doing nothing and no programs are open. They are checking for updates, loading email, and many other things, On a UNIX system, use the ps -a command to see them. On a Windows system, use the task manager.

19. Is there any reason why you might want to mount a file system on a nonempty directory? If so, what is it?

Ans: Mounting a file system makes any files already in the mount-point directory inaccessible, so mount points are normally empty. Howev er, a system administrator might want to copy some of the most important files normally located in the mounted directory to the mount point so they could be found in their normal path in an emergency when the mounted device was being repaired.

27. Modern operating systems decouple a process address space from the machine’s physical memory. List two advantages of this design.

Ans: This allows an executable program to be loaded in different parts of the machine’s memory in different runs. Also, it enables program size to exceed the size of the machine’s memory

28. To a programmer, a system call looks like any other call to a library procedure. Is it important that a programmer know which library procedures result in system calls? Under what circumstances and why?

Ans: As far as program logic is concerned, it does not matter whether a call to a library procedure results in a system call. But if performance is an issue, if a task can be accomplished without a system call the program will run faster. Every system call involves overhead time in switching from the user context to the kernel context. Furthermore, on a multiuser system the operating system may schedule another process to run when a system call completes, further slowing the progress in real time of a calling process.

31. Explain how separation of policy and mechanism aids in building microkernel-based operating systems.

Ans: As far as program logic is concerned, it does not matter whether a call to a library procedure results in a system call. But if performance is an issue, if a task can be accomplished without a system call the program will run faster. Every system call involves overhead time in switching from the user context to the kernel context. Furthermore, on a multiuser system the operating system may schedule another process to run when a system call completes, further slowing the progress in real time of a calling process.

32. Virtual machines have become very popular for a variety of reasons. Nevertheless, they have some downsides. Name one.

Ans: The virtualization layer introduces increased memory usage and processor overhead as well as increased performance overhead.

Processes and Threads

18. What is the biggest advantage of implementing threads in user space? What is the biggest disadvantage?

Ans: . The biggest advantage is the efficiency. No traps to the kernel are needed to switch threads. The biggest disadvantage is that if one thread blocks, the entire process blocks.

26. In Sec. 2.3.4, a situation with a high-priority process, H, and a low-priority process, L, was described, which led to H looping forever. Does the same problem occur if roundrobin scheduling is used instead of priority scheduling? Discuss.

Ans: With round-robin scheduling it works. Sooner or later L will run, and eventually it will leave its critical region. The point is, with priority scheduling, L never gets to run at all; with round robin, it gets a normal time slice periodically, so it has the chance to leave its critical region.

29. The producer-consumer problem can be extended to a system with multiple producers and consumers that write (or read) to (from) one shared buffer. Assume that each producer and consumer runs in its own thread. Will the solution presented in Fig. 2-28, using semaphores, work for this system?

ans: Yes, it will work as is. At a given time instant, only one producer (consumer) can add (remove) an item to (from) the buffer.

36. A fast-food restaurant has four kinds of employees: (1) order takers, who take customers’ orders; (2) cooks, who prepare the food; (3) packaging specialists, who stuff the food into bags; and (4) cashiers, who give the bags to customers and take their money. Each employee can be regarded as a communicating sequential process. What form of interprocess communication do they use? Relate this model to processes in UNIX.

Ans: The employees communicate by passing messages: orders, food, and bags in this case. In UNIX terms, the four processes are connected by pipes.

File Systems

8. A simple operating system supports only a single directory but allows it to have arbitrarily many files with arbitrarily long file names. Can something approximating a hierarchical file system be simulated? How?

Ans: Use file names such as /usr/ast/file. While it looks like a hierarchical path name, it is really just a single name containing embedded slashes.

11. Contiguous allocation of files leads to disk fragmentation, as mentioned in the text, because some space in the last disk block will be wasted in files whose length is not an integral number of blocks. Is this internal fragmentation or external fragmentation? Make an analogy with something discussed in the previous chapter.

Ans: Since the wasted storage is between the allocation units (files), not inside them, this is external fragmentation. It is precisely analogous to the external fragmentation of main memory that occurs with a swapping system or a system using pure segmentation.

17. For a given class, the student records are stored in a file. The records are randomly accessed and updated. Assume that each student’s record is of fixed size. Which of the three allocation schemes (contiguous, linked and table/indexed) will be most appropriate?

Ans: For random access, table/indexed and contiguous will be both appropriate, while linked allocation is not as it typically requires multiple disk reads for a given record.

27. Oliver Owl’s night job at the university computing center is to change the tapes used for overnight data backups. While waiting for each tape to complete, he works on writing his thesis that proves Shakespeare’s plays were written by extraterrestrial visitors. His text processor runs on the system being backed up since that is the only one they have. Is there a problem with this arrangement?

Ans: Ollie’s thesis may not be backed up as reliably as he might wish. A backup program may pass over a file that is currently open for writing, as the state of the data in such a file may be indeterminate

Memory Management

5. What is the difference between a physical address and a virtual address?

Ans: Real memory uses physical addresses. These are the numbers that the memory chips react to on the bus. Virtual addresses are the logical addresses that refer to a process’ address space. Thus a machine with a 32-bit word can generate virtual addresses up to 4 GB regardless of whether the machine has more or less memory than 4 GB

9. What kind of hardware support is needed for a paged virtual memory to work?

Ans: There needs to be an MMU that can remap virtual pages to physical pages. Also, when a page not currently mapped is referenced, there needs to be a trap to the operating system so it can fetch the page.

17. Suppose that a machine has 38-bit virtual addresses and 32-bit physical addresses. (a) What is the main advantage of a multilevel page table over a single-level one? (b) With a two-level page table, 16-KB pages, and 4-byte entries, how many bits should be allocated for the top-level page table field and how many for the next level page table field? Explain.

Ans: Consider,

(a) A multilevel page table reduces the number of actual pages of the page table that need to be in memory because of its hierarchic structure. In fact, in a program with lots of instruction and data locality, we only need the toplevel page table (one page), one instruction page, and one data page.

(b) Allocate 12 bits for each of the three page fields. The offset field requires 14 bits to address 16 KB. That leaves 24 bits for the page fields. Since each entry is 4 bytes, one page can hold 212 page table entries and therefore requires 12 bits to index one page. So allocating 12 bits for each of the page fields will address all 238 bytes

45. Explain the difference between internal fragmentation and external fragmentation. Which one occurs in paging systems? Which one occurs in systems using pure segmentation?

Ans: Internal fragmentation occurs when the last allocation unit is not full. External fragmentation occurs when space is wasted between two allocation units. In a paging system, the wasted space in the last page is lost to internal fragmentation. In a pure segmentation system, some space is invariably lost between the segments. This is due to external fragmentation.

48. Can you think of any situations where supporting virtual memory would be a bad idea, and what would be gained by not having to support virtual memory? Explain.

Ans: General virtual memory support is not needed when the memory requirements of all applications are well known and controlled. Some examples are smart cards, special-purpose processors (e.g., network processors), and embedded processors. In these situations, we should always consider the possibility of using more real memory. If the operating system did not have to support virtual memory, the code would be much simpler and smaller. On the other hand, some ideas from virtual memory may still be profitably exploited, although with different design requirements. For example, program/thread isolation might be paging to flash memory

“Describe the process in figure 3-29 in detail”

Diagram

Description automatically generated