CHAPTER 7

Operating Systems

(Solutions to Practice Set)

Review Questions

1. An operating system is a program that facilitates the execution of application programs.
2. Components of an operating systems are user interface, memory manager, process manager, device manager, and file manager.
3. In monoprogramming, only a single program is in memory at any time. In multiprogramming, several programs are in memory at a time, but the resources of the computer are only assigned to the program that is running.
4. In partitioning, memory is divided into variable-length sections, each of which holds one complete program. In paging, memory is divided into much smaller fixed-length sections as is the program itself; the program does not have to be contiguous in memory.
5. In regular paging the entire program must be in memory at the same time in order for the program to execute. With demand paging, only some pages of a program can be in memory. This means that, in demand paging, more programs can use the computer's resources at any given time.
6. A program is a non-active set of instructions stored on a disk and does not become a job until it is selected for execution. A job is a program that is scheduled for execution and does not become a process until it actually gets loaded into memory and starts executing.
7. A process resides, at least in part, in main memory. Programs and jobs reside on a disk.
8. The job scheduler moves a job from the hold state to the ready state or from the running state to the terminated state. The process scheduler moves a process from one state to another.
9. An operating system needs to use queues because there can be many jobs and processes active at the same time. In order to share all of the resources, queues are

necessary to make sure that jobs and processes all get access to the resources that they need.

10. Deadlock happens when processes are all waiting for resources held by other processes: they are all waiting for each other. This happens when the operating system does not put resource restrictions on processes. Starvation happens when the operating system puts too many resource restrictions on a process. If a process must wait until it can get all of the resources that it needs before it starts to execute, it may never start.

Multiple-Choice Questions

11. a	12. a	13. b	14. c	15. a	16. a
17. a	18. a	19. d	20. c	21. c	22. c
23. a	24. d	25. a	26. b	27. d	28. a
29. d	30. c				

Exercises

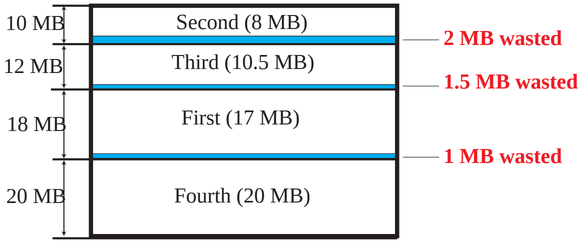
31. 64 – 4 = 60 MB.

32. 64 – (10 + 4) = 50 MB.

33. 70 / (70 + 10) × 100 = 87.5%.

34. Figure S7.34 shows the partitions and memory used by each program.

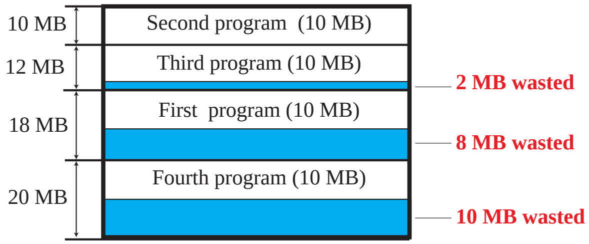
Figure S7.34 *Exercise 34*



Total memory used = 17 + 8 + 10.5 + 20 = 55.5 MB.
Total memory wasted = 2 + 1.5 +1 = 4.5 MB.
Percent memory wasted = 4.5 / 60 × 100 = 7.5%.

35. Figure S7.35 shows the partitions and memory used by each program. The total memory wasted = 2 + 8 + 10 = 20 MB. Memory wasted = 40 / 60 × 100 = 33.3 percent. The total memory used = 10 + 10 + 10 + 10 = 40 MB.

Figure S7.35 *Exercise 35*



36.

a. 13 / 4 = 3.25 → 4 pages.

b. 12 / 4 = 3 pages.

c. 27 / 4 = 6.75 → 7 pages.

d. 15 – (4 + 3 + 7) = 1 frame.

e. Not considering memory lost inside each frame, one frame (4 MB) is unused.

f. (4 / 60) × 100 = 6.66%.

37.

Total memory = 1000 + 100 = 1100 MB. Number of program = 1100 / 10 = 110.

38.

a. running

b. ready

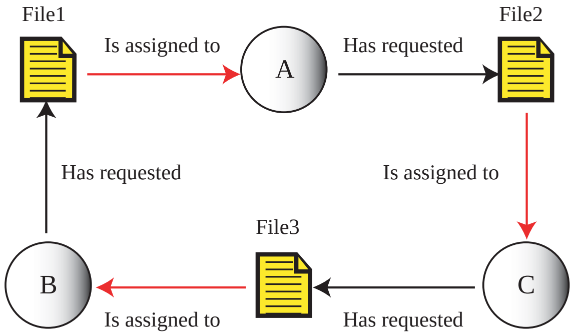
c. ready

d. waiting

e. waiting

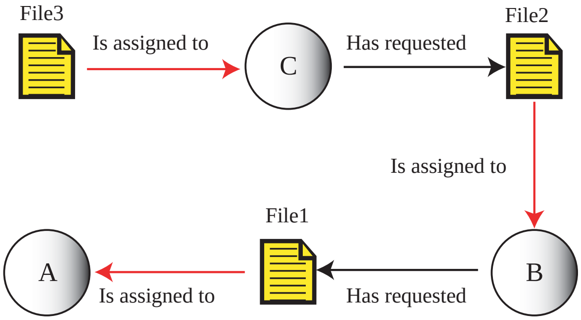
39. This is a deadlock situation (see Figure S7.39) because all four conditions of deadlock (mutual exclusion, resource holding, no preemption, and circular waiting) are all present.

Figure S7.39 *Exercise 39*



40. This is not a deadlock situation (see Figure S7.40) because one of the necessary condition (circular waiting) is not present.

Figure S7.40 *Exercise 40*


File3
Is assigned to
C
Has requested
File2
Is assigned to
File1
A
Is assigned to
Has requested
B