Sheet 6 - Sol

I

- **3.1**
  File manager, device drivers, memory manager, scheduler, and dispatcher.
- **3.3**
  R, S, T, X, Y, Z (The items are removed in the same order they were placed in the queue.)
- **3.5**
  An operating system that allows several activities to execute "at the same time".
- **3.8**
  a) The user interface of an operating system handles the communication with the operating system's users.
  b) The kernel of an operating system performs the fundamental tasks of the system.
- **3.14**
  Virtual memory size in bytes = 2 × 512 = 1024 MB = 1048576 KB
  Virtual memory size in pages = 1048576 / 2 = 524288 Page
- **3.16**
  Application software performs tasks that are unique to the use of the particular computer system like spreadsheets, database systems, desktop publishing systems, accounting systems, program development software, and games.
  System software performs tasks that are required as the software infrastructure of any computer system like operating systems and utility software
- **3.18**
  The machine begins by executing a program, called the bootstrap, at a predetermined location in memory. This program directs the machine to load a program (the operating system) from mass storage into main memory. The original program tells the machine to transfer its attention to the program just loaded.
- **3.19**
  Since most of a computer's main memory is volatile, the operating system must be reloaded each time the machine is turned on.
- **3.21**
  Number of instructions per nanosecond = 5
  Number of instructions per millisecond = $5 * 10^6$
  Number of instructions per time slice = $5 * 10^6 \times 10 = 5 * 10^7$
  The point is that a modern machine can do a lot in a single time slice.
- **3.27**
  1. Save the current process' state;
  2. Select another process from the process table;
  3. Load that process' state;
  4. Start the next time slice.
- **3.28**
  A process's state includes the values in the CPU's registers (including the program counter) as well as the contents of its associated memory cells.
- **3.33**
  The test-and-set instruction is often used to implement semaphores. Since its task is executed as a single instruction, no interrupt signal can interfere.
- **3.42**
  The point of this problem is as much to introduce students to this piece of computer science folklore as it is to pose the problem itself. Issues include the problem of each philosopher obtaining possession of one fork (deadlock) as well as the problem of a philosopher's neighbors obtaining possession of the forks available to him and never releasing them (starvation).
- **3.43**
  As the length of time slices become smaller, the ratio of time spent swapping processes compared to the time spent executing them increases. Thus, a point is reached where the efficiency of the system becomes quite low. On the other hand, if time slices are too long, the illusion of simultaneous operation is lost.

II

1.

a) A *program* is an algorithm written using a compute programming language or machine language.
b) A *job* is a program to be executed.
c) A *process* is a program in execution.
d) A *task* is executing a program or a part of it.
e) *Batch processing* is the execution of jobs by collecting them in a single batch, then executing them without further interaction with the user. Jobs wait for execution in a job queue.
f) *Interactive processing* is allowing a program being executed to carry on a dialogue with the user. Interactive processing requires execution of programs in real-time.
g) *Multiprogramming* is dividing processor time into intervals and restricting the execution of each job to only one interval at a time. At the end of each interval, the current job is temporarily set aside and another is allowed to execute during the next interval. By rapidly shuffling the jobs back and forth in this manner, the illusion of several jobs executing simultaneously is created.
h) A *semaphore* is a flag used by the operating system for controlling access to machine resources.
i) A *deadlock* is the condition in which two or more processes are blocked from progressing because each is waiting for a resource that is allocated to another.

2.

1. Competition for non-sharable resources
2. Resources requested on a partial basis
3. An allocated resource can not be forcibly retrieved