1. True or false:
   ___ T   _X_ F   Lightweight processes were invented to provide an alternative to the exec() system call. (They provide a faster alternative to fork().)
   ___ T   _X_ F   Processes waiting for the CPU are in the waiting state. (No, they are in the ready state.)
   _X_ T   ___ F   The pthread library implements user-level threads.
   ___ T   _X_ F   The number of processes that can be simultaneously in the running state is hardware-specific. (It depends on the number of system processing units.)

2. How many lines of output will be printed by the following program?
   main(){
      int pid;
      if ((pid = fork()) == 0) {
         printf("How are you?\n");
      }
      printf("Fine. Thank you.\n");
   } //main

   Answer: The program will print exactly _3_ lines.

3. Give two major disadvantages of user-level threads over kernel supported threads.
   • User-level threads cannot take advantage of multiprocessor architectures and let multiple threads of the same task run at the same time on different processing units.
   • When a user-level thread does a blocking system call, all threads sharing its address space are blocked.

4. Which processes can be safely swapped out and why?
   Processes that have been in the waiting state for a long time can be safely swapped out as they are likely to remain for some time in the waiting state.

5. What would be the result of the execution of the following piece of code?
   int pipedesc[2];
   ...
   pipe(pipedesc);
   close(0);
   dup(pipedesc[0]);

   Standard input is now redirected to pipedesc[0].