1. For each term below indicate the number of (human) users, CPUs, and operating systems. You answers should be one of the following: 0 (none), 1 (exactly one), >=1 (one or more), or >1 (more than one). Be ready to justify or explain each answer. OS's should be interpreted as complete copy of an OS (rather than type of OS)

<table>
<thead>
<tr>
<th>users types</th>
<th>CPUs copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>uniprogrammed uniprocessor</td>
<td>___</td>
</tr>
<tr>
<td>multiprogrammed uniprocessor</td>
<td>___</td>
</tr>
<tr>
<td>timeshared uniprocessor</td>
<td>___</td>
</tr>
<tr>
<td>multiprogrammed multiprocessor</td>
<td>___</td>
</tr>
<tr>
<td>(parallel processor)</td>
<td>___</td>
</tr>
<tr>
<td>network of workstations</td>
<td>___</td>
</tr>
</tbody>
</table>

2. Consider the various definitions of **operating system**. Consider whether the operating system should include applications such as web browsers and mail programs. Argue both pro and con positions, and support your answer.

3. Protecting the operating system is crucial to ensuring that the computer system operates correctly. Provision of this protection is the reason for dual-mode operation, memory protection and the timer. To allow maximum flexibility, however, you should also place minimal constraints on the user.

The following is a list of instructions that are normally protected. What is the minimal set of instructions that must be protected? Why?

- **a)** Change to user mode.
- **b)** Change to monitor mode.
- **c)** Read from monitor memory.
- **d)** Write into monitor memory.
- **e)** Fetch an instruction from monitor memory.
- **f)** Turn on timer interrupt.
- **g)** Turn off timer interrupt.
4. When an interrupt occurs, the PC gets loaded with a new value. For each of the answers below, write "+" if it describes a value that could be loaded into the PC just after the interrupt occurs. Write "0" if it describes a value that would not be loaded into the PC just after the interrupt occurs. Assume interrupts are enabled (not masked).

_____ address of the next instruction in memory
_____ address of the first instruction in the interrupt handler
_____ address in the user's program
_____ address in the operating system kernel
_____ an address that it has loaded in the past

5. What are the three major activities of an operating system in regard to memory management? What are the three major activities of an operating system in regard to secondary-storage management?

6. Which of the following information is stored in the Process Control Block for each process?

_____ PID (process ID)
_____ PC (program counter) - what memory address is in the saved PC?
_____ link to next PCB
_____ the code for this process
_____ register contents
_____ status/condition code flags
_____ CPU state (new, ready, running, blocked/wait, zombie)
_____ scheduling priority
_____ name of user who created this process

7. Describe the actions taken by a kernel to switch context between processes.