1a.

2. An interrupt is the signal emitted at a hardware level on a processor in order for anything to happen. The purpose of an interrupt is to get certain parts of the processor to perform certain tasks. An interrupt is different than a trap because a trap comes from the software level. Traps are most commonly used to switch the processor mode from “user” to “kernel.” Because of this, user programs are unable to generate traps.

3.

4. One mechanism for enforcing memory protection is restricting user processes to only be allowed to write to memory in their address space. The address space is determined by the operating system and kept track of in two registers on the processor. When the kernel gives control to a user process it loads two addresses onto the processor. The addresses are stored in a “base” and “limit” register and these addresses dictate where a user process is allowed to write in memory.

5. Set value of a timer: **kernel**

Read the clock: **user**

Clear memory: **kernel**

Issue a trap: **kernel**

Mask interrupts: **kernel**

Modify entries in device-status table: **kernel**

Switch from user to kernel: **kernel**

Access: **user**