1. a) No. Trap instruction occurs at user level, and it is responsible for moving process from user mode to kernel mode.

   Correct Solution

   No. Trap instruction must also needs to executed called at user level (e.g. system call), to move process from user mode to kernel mode

   b) This question is omitted. It is not covered in class

   c) No. If the type of access is read for both threads then there will be no concurrency error

   Correct Solution

   No. If both threads are reading the shared variable then it's fine, and furthermore, it's also fine when only one of two threads are updating the shared variable and reading whichever shared variable, before and after update, by the other thread doesn't matter

   d) No. Limited direct execution means running a process in CPU but with limited permission, and it can be thought as baby proofing CPU, so bad code won't harm the system.

   e) No. Indexed based system uses block pointers in inode, and block pointer can be pointing data blocks in the data region.

   f) No. Extent-based file system requires only extent + length to get to a particular byte in file, and this differs from indexed-based system which uses many indirect pointers in between (which adds disk access).

2. a)
   - Process counter
   - Process state
   - Process ID
   - Process Register

   b) Before going from user to kernel mode, the following must be saved

   - User/Process Register
   - Stack pointer
   - Frame pointer
   - I/O Information
   - Process ID
   - Process State

   so that upon executing return-from-trap instruction, the process can resume where it has left.

   c) This question is omitted. It is not covered in class