1. The implementation given of `thread_switch` in class is as follows:

   ```c
   void thread_switch( ) {
     thread_t NextThread, OldCurrent;
     NextThread = dequeue(RunQueue);
     OldCurrent = CurrentThread;
     CurrentThread = NextThread;
     swapcontext(&OldCurrent->context, &NextThread->context);
     // We’re now in the new thread’s context
   }
   ```

   Suppose we don’t have a `swapcontext` routine to call. Instead, we have to use `setjmp` and `longjmp`. Recall that `setjmp` saves a thread’s registers (including the stack pointer) in a data structure of type `jmp_buf` and returns 0. The routine `longjmp` restores the registers from its first argument (of type `jmp_buf`); the instruction pointer is set so that control continues at the instruction following the original `setjmp`, which now returns whatever `int` value is given as the second argument to `longjmp`. Show how to implement `thread_switch` using `setjmp` and `longjmp`. If changes to the thread control block (of type `thread_t`) are required, indicate what they are.

2. In the x86 architecture, interrupts are processed using the current thread’s kernel stack. However, on the VAX architecture, there was a per-processor interrupt stack on which interrupts were processed. Thus, when an interrupt occurred, the current context was saved on the processor’s interrupt stack and the interrupt handler was run on that interrupt stack.
   a. We discussed in class how Windows uses its DPC facility to implement preemption: when the current thread is to be preempted (perhaps because the current thread’s time slice has expired) a DPC (deferred procedure call) is requested. The DPC interrupt handler calls `thread_switch` to switch to the context of the next thread. Explain why this approach doesn’t work on the VAX architecture.
   b. Instead, the VAX architecture had a feature called `asynchronous system traps` (ASTs), which is similar, but not identical, to the Windows APC (asynchronous procedure call). Among the processor registers is a special control register containing various flags, one of which is the `AST-request flag`: when this is set, a trap occurs and the AST handler is called on the kernel stack of the current thread. This register is among those saved when an interrupt occurs and is restored when an interrupted thread is resumed. When an interrupt handler is entered, a copy of this register is loaded that has the AST-request flag cleared. Explain how ASTs might be used to implement preemption on the VAX architecture, achieving the same effect as the DPC approach on the x86 architecture.
3. Consider an architecture for which all sensitive instructions are privileged instructions (the IBM 360 is an example of such an architecture; the x86 is not). On such architectures it is possible to run an operating system known as a virtual machine monitor (VMM): it supports virtual machines in which operating systems designed to run on the real machine can run, unmodified. The OS running in the virtual machine is called a guest OS. Suppose we run a copy of the VMM itself as the guest OS, and it, in turns, runs a guest OS in one of its virtual machines. This is known as nested virtualization.

Let’s say that OS_0 is running on the real hardware, OS_1 is running as guest OS on OS_0, and, in general, OS_{i+1} is running as a guest OS on OS_i. Suppose, for some i>0, a sensitive instruction (which, of course, is also a privileged instruction) is executed within OS_i. Thus, since the real machine is running in user mode, a privileged-instruction trap occurs and the (real) hardware invokes the trap handler of OS_0. Assuming the virtual machine running OS_i is running in (virtual) privileged mode, explain the sequence of events that must occur to process this instruction. In particular, explain what happens in each of the OS_k’s, for k<i. Your answer should not be long or complicated; there is no need to supply any code. You don’t need to supply the details of what happens when, say, a virtual machine monitor emulates the effect of a trap in one of its virtual machines; you may simply say that the VMM effects the trap in the virtual machine. You may assume that each VMM is supporting exactly one virtual machine.

Hint: a virtual machine is either in virtual privileged mode or virtual user mode, i.e., it should behave as if it is in either privileged mode or user mode. The virtual machine monitor implementing it keeps track of which mode the virtual machine “thinks” it’s in, but, as far as the VMM is concerned, the virtual machine is always in (real) user mode. Thus if a privileged instruction is executed in the virtual machine, the VMM is notified by a privileged-instruction trap. If the VMM is running on the real machine, the trap is generated by the real hardware. But if the VMM is, itself, running in a virtual machine, what causes the trap to happen? When the VMM responds to the trap, its response depends upon whether the virtual machine it is supporting thinks it is in privileged mode or user mode.

4. Explain how renaming a file can be done using the consistency-preserving approach so that a crash will not result in the loss of the file. Be sure to take into account the link count stored in the file’s inode. Note that there may have to be a temporary inconsistency; if so, explain why it will not result in lost data.

If you do all of the following correctly, you’ll get an A regardless of how well you do on the first four problems. If you miss any of the following, your grade will be based solely on how well you do on the first four problems.

5. Who was president of Brown when the first current CS faculty members were hired? Who was president of Brown when the professor of this course was hired?

6. Whose name is in large letters on the front of the CIT building? Who was this person and what was his or her relationship with Brown?

7. Whose name is in large letters on the outside of CIT 477? In addition to being a Brown alum, this person also had a relationship with the CS department because of a computer company he or she was associated with. What company was this?

8. Two prominent computer-company executives (and alums) and a famous CS academic (not an alum) were featured speakers at the dedication of the CIT building in 1988. Who were they?

9. The Brown Mathematics Department currently resides in Kassar House, at 151 Thayer Street. Prior to its being the Mathematics building, it was the CS building. It was named by Raymond Kassar ’48 in honor of his father. What was Ray Kassar’s job at the time he pledged money for the renovation of Kassar house? How is his last name pronounced (in particular, which syllable is stressed)?

10. What is the name of the female CS concentrator (class of 1983) who is now a member of the Brown Board of Trustees?