Short Discussion on Mid-Term:

Question 1:
For function `do_switch(int *sp1, int *sp2)`, what it does is:
1. Save the stack pointer (sp) to the memory location pointed to by sp1: sp -> sp1
2. Load sp from the memory location pointed to by sp2: sp2 -> sp
Question 2) This question was about implementing a function using some variable and condition or semaphores. Here we count the variable rendezvous to determine the if it is thread 1 or thread.

Question 3 and 4  → Refer to solution:  
http://www.cs.nmsu.edu/~pfeiffer/classes/473/notes/intelvm.html
Hardware Virtualization and Virtual Machine

Virtualization

Systems like VM Ware and Virtual Box let us run unmodified OS on top of another one. We can also run multiple copies of OS on the same machine.

Question: Why cannot we run OS on top of an OS?
ANS: Every application runs on the user mode. But to run OS, to do privileged task, it require to run in supervisor mode. Consider for example if the linux is running on the top of MS OS and if linux try to do system call then it will go to windows kernel instead of linux kernel. Thus linux need to work in supervisor mode for its kernel to be called when system call is made.

1) One of the way to overcome this situation is using emulation.

Software emulation: Here we write a program to emulate the hardware: variables for each of the registers in CPU, a loop to read instructions, code for instructions like MOV, JUMP and POP

Virtual machines can also perform the role of an emulator, allowing software applications and operating systems written for another computer processor architecture to be run.

Some virtual machines emulate hardware that only exists as a detailed specification. For example:

- One of the first was the p-code machine specification, which allowed programmers to write Pascal programs that would run on any computer running virtual machine software that correctly implemented the specification.
- The specification of the JVM
- The Common Language Infrastructure virtual machine at the heart of the Microsoft .NET initiative.
- Open Firmware allows plug-in hardware to include boot-time diagnostics, configuration code, and device drivers that will run on any kind of CPU.
This technique allows diverse computers to run any software written to that specification; only the virtual machine software itself must be written separately for each type of computer on which it runs.

Disadvantage
It works slower. It can work fast only if we emulate slower machine with faster machine.

**Hardware Virtualization**

One way to solve above problem is to use hardware virtualization.

Here the basic method is to run entire machine such that both application and operating system work in the user mode.
The diagram on the left below illustrates the structure of a single machine. On the right, it shows how the machine that runs virtual machine is structured. Each guest OS runs directly on top of the hypervisor. They are like processes and threads to the hypervisor. Therefore, we are able to run multiple copies of OS at the same time and these copies can be switched pretty much like context switching.

Question: How do we fully virtualize?
Ans: 1) Run the guest OS in user mode.
   2) Privileged instructions trap.
   3) We emulate them.

If the hypervisor gets the interrupt, it goes to 2 and sees who the kernel or the OS and then to its interrupt handler

If the OS is not aware of the supervisor, what we can do for an instance is

We see the page fault at p=0, then the hypervisor sees the page fault. Page fault can be used to access read by write fault from this memory access.

*Another Example:* Example of simple virtual drive adaptor.
If we use page fault to trap to trap all these memory location.

Question: How do we have to virtualize?
Answer: 1) Trap Handler
   Any sort of trap that goes into the supervisor mode has to be emulated by the hypervisor.

2) Page Tables
   We need to control page tables when the guest kernel is running.

3) System Calls
   Any sort of system calls goes to supervisor mode has to be emulate by the hypervisor

4) Interrupts

5) Page Faults
6) Memory
   Memory map in virtual machine
To map the virtual memory address base to the real memory address base, we make sure that the page tables that the hardware sees are not the same as the guest OS sees. The general idea is we need to keep different copies of everything for the hardware and the virtual machine. On real R/W the page table that hardware see area not same page table that virtual machine sees.

**Question And Answer**

1) Compare to fully virtualization, is there any other kind of virtualization? Para virtualization, In this virtualization the hypervisor the modified version of the underlying physical h/w. This exported machine is of has same architecture, instead target modification are introduced to make it simpler and faster to support multiple guest operating system.

2) Is there anything like memory sharing between virtual machines? There is no reason why we cannot share memory between 2 virtual machines similar to sharing of processes.