2017073

Assignment 6

1.
- yield_on_return is set after every TIMER_SLICE timer interrupts. For PINTOS, TIMER_SLICE is 4. ∴ yield_on_return is set after every 4 timer interrupts.
- thread_yield() routine yields the CPU to the scheduler. It puts the current running thread back into the ready list and calls the schedule() routine to pick a new thread to run. This new thread could be the previously running thread too.
- During a system call, the kernel executes on behalf of the thread, for which it uses a separate kernel stack for each thread (per-thread kernel stack). If we have just one kernel stack, kernel pre-emption will be a problem. If a timer interrupt arrives while the kernel is executing a system call, the scheduler might pre-empt the currently executing user space thread while it is still executing in kernel space. The next time the process is scheduled, it must begin execution where it left off. If there is a single kernel stack, other threads could corrupt the stack preventing threads to resume execution after being pre-empted.
- Yes, it is possible to operate PINTOS as a single kernel stack architecture. Due to the loss of the state stored in the stack across a context switch, function return addresses and local state

that will be needed after the context switch must be stored in an alternate location such as the thread control block explicitly. Context switches modelled as a function call must never return.

— x —