CS 347 QUIZ 3 (Aug 4, 2016)

Name: _____ ***Roll No.*** _____

Please write your answer in the space next to the question.

For all questions, assume that the processes are running on x86 hardware with the xv6 OS.

1. Suppose a process P1 is executing on a CPU, when a disk interrupt with data that unblocks another process P2 occurs. Below are several steps that happen while P1 services the interrupt. For each step below, specify whether the action is implemented by the CPU in hardware, or by the kernel code in software. Your answer below should be either "CPU" or "kernel".

A. EIP (program counter in x86) shifts from the user code of the process to the kernel code that handles interrupts.

B. EIP of the user code instruction where execution stopped is pushed on to the kernel stack.

C. Several general purpose registers are pushed onto the kernel stack.

D. ESP (stack pointer) moves from pointing to the user stack to pointing to the kernel stack.

E. The process P2 that is unblocked by the interrupt is marked as ready/runnable in the global process table data structure.

2. Consider a parent process P that has executed a fork system call to spawn a child process C. Suppose that P has just finished executing the system call code, but has not yet returned to user mode. Also assume that the scheduler is still executing P and has not context switched it out. Below are listed several pieces of state pertaining to a process in xv6. For each item below, answer if the state is identical in both processes P and C. Answer Yes (if identical) or No (if different) for each question.

A. Contents of the PCB (struct proc). That is, are the PCBs of P and C identical? (Yes/No)

B. Contents of the memory image (code, data, heap, user stack etc.).

C. Contents of the page table stored in the PCB.

D. Contents of the kernel stack.

E. EIP value in the trap frame.

F. EAX register value in the trap frame.

G. The physical memory address corresponding to the EIP in the trap frame.

H. The files pointed at by the file descriptor table. That is, are the file structures pointed at by any given file descriptor identical in both P and C?

3. Suppose the kernel has just created the first user space "init" process, but has not yet scheduled it. Answer the following questions.

A. What does the EIP in the trap frame on the kernel stack of the process point to?

B. What does the EIP in the context structure on the kernel stack (that is popped when the process is context switched in) point to?