Derek Lee, Shine Li Computer Operating Systems Fall 2020

Professor Hakner PSET #7

Problem 1

a) First, the kernel pushes the %eax register onto the stack. It also pushes all of the registers that will likely be used. It allocates a stack area for each user-level thread. Then, it adds the thread_info struct at the lowest memory address of the stack. It masks the stack pointer to find the beginning of the thread_info struct. Next, the bitwise flags word is examined. After that, data validation takes place. Since the system call is negative, the syscall_badsys code is jumped to, which places an error code into the stack where the eax register was placed. Once the system call handler is exited, the value is popped into %eax. Since an error occurred, a negative value is returned. In this case, the value returned is -ENOSYS. Because of the UNIX API, the system call returns -1, and errno is set to ENOSYS.

b) The handler terminates by executing the special iret instruction, which resets the privilege level to the previous value. This is accomplished through the %eflags register, which has been restored from the stack.

c)

i) In a fully pre-emptive Linux kernel, once the interrupt handler completes, a context switch takes place. The execution of Process 123 is temporarily suspended and Process 999 gets the CPU. Some time later, Process 123 resumes later.

ii) In a non pre-emptive Linux kernel, once the interrupt handler completes, Process 123 continues execution. Then, once the system call finishes, upon return to user mode, pre-emption takes place. Now, Process 999 gets the CPU.