Collaboration

You must work on this quiz with a partner. You should have listed a partner on the Google form available on Moodle *at class time*, and received a confirmation message regarding your partner from the professor.

Deliverables

You should deliver a **PDF** file containing the answers to the questions.

1. [3 pts] When performing context switches, the hardware saves instruction pointer, stack, and the CPU control register (EFLAGS), while the OS saves the remaining registers (`eax`, `ebx`, ...). Why do we *need* that the hardware saves those registers while the OS can save the rest?

   - A short paragraph (about 4 lines) should be enough for the answer. Answer in the assigned reading of Sep 19 (look at Moodle).

2. [3 pts] In theory, upon context switch, we could save the process registers in the process' own stack. Instead, OSs save the process registers in an interrupt stack located in the kernel. Give two reasons for *why* the latter approach is taken.

   - A short paragraph (about 4 lines) should be enough for the answer. Answer in the assigned reading of Sep 19 (look at Moodle).

3. [4 pts] Chapter 5, exercise 4. Tell me how (1) a thread can "pass" information to another thread; (2) if a thread "passes" a pointer to a stack variable, *how* and *why* could chaos immediately ensue.

   - A short paragraph (about 4 lines) should be enough for the answer. Answer in the assigned reading of Sep 19 (look at Moodle).

4. [7 pts] Chapter 7, exercise 4. Please provide a graphic view of the times each process is allocated similar to the slides given in class.

   - Answer in the assigned reading of Oct 15 (look at Moodle).

5. [3 pts] Chapter 7, exercise 11.

   - Answer in the assigned reading of Oct 15 (look at Moodle).