Quiz 2A (10 pts) – CS503 Spring 2019

Answer the questions in the spaces provided. If you run out of room, continue on the back.
Provide clear answers and justify statements where required.
Unless otherwise noted, assume that questions refer to C, AT&T assembly, and x86.

1. (2 points) Name:

2. (6 points) State whether the following statements are true of false. **Explain the false statements.**

   (a) (1 point) The assembly statement "`push %eax`" moves the value of EAX to an implicit register.

   ◯ True
   ◯ False. Explanation:

   Solution: False. Reads value from the EAX register and puts it in the stack. (No need to discuss the stack pointer.)

   (b) (1 point) The Xinu system call "create()" creates a new conditional variable.

   ◯ True
   ◯ False. Explanation:

   Solution: False. Creates a new process.

   (c) (1 point) Xinu types only encompass the size of variables.

   ◯ True
   ◯ False. Explanation:

   Solution: False. Size and purpose.

   (d) (1 point) Only a suspended process can be resumed in Xinu.

   ◯ True
   ◯ False. Explanation:

   Solution: True.

   (e) (1 point) The process table stores information about all processes in the system. In Xinu this table includes the null process.

   ◯ True
   ◯ False. Explanation:

   Solution: True.

   (f) (1 point) In Xinu, only processes in the ready list are eligible to run and can be chosen by the resched() function.

   ◯ True

◯ False. Explanation:

Solution: False. The current process is also eligible and can be chosen.

3. (2 points) When creating a new process the operating system needs to assign a new process ID to the new process. Explain the high-level consideration when assigning an ID to a new process in the context of ensuring good OS semantics. Describe a simple but effective approach to address such problems.

Solution: OSs should allocate PIDs such that the time between PID reuse by different processes is generally high (+0.5 points); this avoids races when invoking system call operations based on PIDs (+0.5 points). Use a large PID space and iterate over this space when allocating new PIDs, making sure to resume iteration on the PID after the previously allocated one. (+1 points)

4. (2 points (bonus)) Xinu uses a structure named "queuetab". Explain its purpose. Provide two examples of lists used by Xinu that leverage this structure. Describe how this structure is conceptually organized. Explain how the "insert()" operation is implemented. You can assume knowledge of a standard list implementation.

Solution: Efficiently implements a double linked list of processes (wrong/missing -1). Ready list, semaphore wait list (wrong/missing -1). Fixed size array with each entry representing a node; first part is for the process nodes and the second part is for the start and end nodes (wrong/missing -1). Iterates over the list until it finds the last element with a key lower or equal than the key of the new node; adds the new element after that node (wrong/missing -1). [Bogus answers get -2 on the question.]