CSCI 3500 – Midterm Reinforcement

This assignment may add up to 15 points to your midterm exam score. **You must complete the entire assignment to receive any credit.** You must show your work to receive credit. This is an individual assignment- do not discuss it with other class members. Please see the instructor about questions you may have.

Match each statement to the most appropriate choices. More than one choice may be appropriate. (10 points)

Choices:

A – System Call

B – Library Function

C – User Program

Statements:

Enters the OS kernel via a trap instruction \_\_\_\_\_\_

Code is stored in user space \_\_\_\_\_\_

Code is stored in kernel space \_\_\_\_\_\_

Switches the processor to kernel (supervisor) mode \_\_\_\_\_\_

Encapsulates (wraps) other functions \_\_\_\_\_\_

Always starts at a fixed physical memory address \_\_\_\_\_\_

Returns control to user space once finished \_\_\_\_\_\_

Pushes library function arguments onto program stack \_\_\_\_\_\_

Calling convention is highly machine dependent \_\_\_\_\_\_

Implements the POSIX standard \_\_\_\_\_\_

Consider the following program:

int main( int argc, char\* argv[] ){

 while(1){

 func\_1 ();

 func\_5 ();

 func\_10 ();

 }

}

Suppose it’s important for func\_1() to execute every 1ms, for func\_5() to execute every 5ms, and func\_10() every 10ms.

What problem occurs if func\_10() needs to load a lot of data from an ancient floppy drive and blocks for 4ms? (1.5 points)

What mechanism would let these three functions execute and be scheduled independently? (1.5 points)

What class of schedulers would be suitable for ensuring that func\_1() executes at least once per millisecond? (1 point)

Supposing each function is scheduled independently, does the completely fair scheduler (CFS) guarantee that func\_1() would have higher priority than func\_10()? (1 point)