Question 1. *(20 points) State whether each of the following actions requires the OS to use kernel mode or whether user mode is sufficient. If kernel mode is required, explain why.*

1. *Reading the current time from the hardware clock.*
2. *Starting a new process.*
3. *Multiplying two numbers stored in an application's memory.*
4. *Writing the results of the multiplication to a temporary log file on disk.*

Question 2. *(10 points) Explain the purpose of a system call. Describe how control passes to the kernel when executing a system call.*

Question 3. *(10 pts) Give an example of an architectural feature that is necessary for* **correctness** *and another feature that is only for* **performance**. *Explain why.*

Question 4. *(10 pts) Compare and contrast a monolithic kernel design with a microkernel design. What are the relative advantages and disadvantages of each approach?*

Question 5. *(15 pts) What happens during a context switch? Do we want our OS to perform many or few context switches? Explain your answer.*

Question 6. *(15 pts) Write down the list of process state transitions that occur during the following program. You may assume that this is the only process that the CPU is executing.*

```
int i = 1;
while (i < 100) { i++; }
printf("%d ", i);
while (i > 0) { i--; }
printf("%d ", i);
```

Question 7. *(20 pts) Consider the following piece of code:*

```
int child = fork();
int c = 5;
if (child == 0) {
    c += 5;
} else {
    child = fork();
    c += 10;
    if (child) {
        c += 10;
    }
}
// do something with c
```

a. *How many copies of the variable c are created by this program? What are their values at the end of this code?*

b. *Describe the hierarchical process tree that is created from running this program. You can assume that all processes have not yet exited.*