

Question 1. (20 points) State whether each of the following actions requires the active process to be in kernel mode or whether user mode is sufficient. Briefly justify.

1. Adding two numbers stored in registers together.
2. Halting (i.e., shutting down) the machine.
3. Creating a new process.
4. Writing data to a file stored on a disk.
5. Invoking a system call to read a file stored on a 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) Consider the following statement: using an OS that supports multiprogramming, multiple programs can be run concurrently on the machine. Is this statement fully, partially, or not at all accurate? Explain.

Question 4. (10 pts) Suppose you are designing an OS in which your primary objective is to make the OS as low-overhead as possible. What type of OS kernel organizational approach would you choose and why? How would your answer change if your primary objective was to minimize the chance of critical OS failure (e.g., due to buggy code in the OS)?

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. (10 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 has to execute.

```
1  int main() {
2      int i = 1;
3      while (i < 5) { i++; }
4      printf("%d ", i);
5      printf("%d ", i - 1);
6      return 0;
7  }
```

Question 7. (25 pts) Consider the following piece of code:

```
1  int c = 5;
2  pid_t pid = fork();
3  if (pid == 0) {
4      c += 5;
5  } else {
6      pid = fork();
7      c += 10;
8      if (pid) {
9          c += 10;
10     }
11 }
12 fork();
13 printf("%d\n", c); // print the value of c
```

- (10 pts) Assuming all calls to `fork` are successful, how many processes are created by running this program? What is a possible output of running the program (i.e., what are the values that are printed by the processes)? Note: you should be able to answer this question without running the program on a computer!
- (10 pts) Describe the hierarchical process tree that is created from running this program, assuming that all processes have been created but not yet exited. You may (but are not required to) draw a picture of the process tree to illustrate.
- (5 pts) What would the process tree be if you inserted an `exec` system call in between lines 2 and 3 that executes the `pwd` program?