

Worksheet 2.1: OS Structures, Multiple-Choice Questions

In the questions below, circle the right answer. There is only one correct answer.

1. What is/are the advantage(s) of using a standard API, like POSIX, relative to using system calls directly?
- a. That provides faster access to OS services.
 - b. That better resolves conflicts among processes requesting access to the same I/O device.
 - c. That makes it possible for one process to access I/O while another process is running on the CPU.
 - d. That makes the application more portable.
 - e. Both c and d are correct.
 - f. Both b and c are correct.
 - g. Both a and c are correct.

Explanation: Using a standard API like POSIX makes applications more portable because the same code can run on different operating systems that support the POSIX standard. Additionally, it allows for better multitasking by enabling one process to access I/O while another process runs on the CPU.

2. How does writing an OS in a high-level language compare with writing it in assembly?
- a. An optimizing compiler *a/ways* generates more efficient code than an assembly programmer.
 - b. An assembly programmer *a/ways* generates more efficient code than an optimizing compiler.
 - c. Writing an OS in assembly makes it easier to develop and debug.
 - d. Writing an OS in a high-level language makes it more portable.
 - e. Both b and d are correct.
 - f. Both a and d are correct.
 - g. a, b and d are correct.

Explanation: High-level languages make the OS more portable because they are less dependent on the hardware. Optimizing compilers can often generate efficient code that approaches or matches the efficiency of hand-written assembly code, making high-level languages a practical choice.

3. How does the loadable module (Mod) structure compare with the layered structure?
- a. In layered all services are loaded in one address space, while in Mod each service is loaded in a separate address space.
 - b. In Mod there is more context switching than in layered.
 - c. In Mod there is more message passing than in layered.
 - d. In layered, there is a hierarchy of services, while in Mod there is no hierarchy.
 - e. Both a and b are correct.
 - f. Both b and d are correct.
 - g. a, b and d are correct.

Explanation: The loadable module structure tends to have more context switching because modules can be loaded and unloaded as needed. The layered structure has a clear hierarchy of services, while the module structure does not necessarily follow a hierarchy, allowing for more flexibility.

4. How does the microkernel structure (Micro) compare with the monolithic (Mono) structure?
- a. Micro is easier to extend than Mono.
 - b. Mono is more secure than Micro, because there is less code running in kernel mode.
 - c. In Mono, if one service crashes, it will crash the whole kernel, but in Micro it won't necessarily crash the whole kernel.
 - d. Micro gives better performance (speed) than Mono.
 - e. Both a and b are correct.
 - f. Both a and c are correct.
 - g. a, b and c are correct.

Explanation: Microkernel architectures are easier to extend because new services can be added without modifying the kernel. They also improve reliability because the failure of a single service does not necessarily crash the entire system. However, microkernels typically have a performance overhead compared to monolithic kernels.