CO327: Operating Systems – 2024
Assignment 2

E/19/166
Jayathunga W. W. K.

1. System call enable user-level processes to request services from the operating system.

2. It reads commands from the user or a file of commands and executes them, typically by converting them into one or more system calls. It is usually not part of the kernel because the command interpreter is prone to modifications.

3. System programs can be viewed as collections of useful system calls. These offer basic functionality, eliminating the need for users to create their own programs to solve common problems.

4. Designing an operating system in a modular manner has numerous benefits. Changes to the system are easier to debug and adjust as they only affect certain portions rather than the entire system. Bugs affecting data must be constrained to a certain module or layer, as it is only stored and accessed inside a defined and restricted area. The fundamental downside of the layered method is poor performance as a result of the overhead associated with traversing through the many layers to get an operating system service.

5. The operating system must be stored in firmware for some devices, such as embedded systems, for which a disk with a file system may not be available.

6. Enforcing protection amongst several processes that are working concurrently in the system is one type of service that an operating system provides. The memory regions that correspond to a process's address spaces are the only ones it can access. Furthermore, it is forbidden for processes to corrupt files linked to other users. Additionally, a process cannot directly access a device without the operating system's help.
   The provision of new functionality not directly enabled by the underlying hardware is the second class of services offered by an operating system. Two instances of novel services offered by an operating system are virtual memory and file systems.

7. **Register** – pass the parameters in registers
   Block – Parameters stored in a block, or table, in memory, and address of block passed as a parameter in a register
   Stack – Parameters placed, or pushed, onto the stack by the program and popped off the stack by the operating system.

8. Every device is accessible just like any other file within the filesystem. It is very simple to add a new device driver by developing the hardware-specific code to support this abstract file interface, as this file interface is used by the majority of the kernel to interact with

devices. Consequently, this helps with the development of device driver code, which can be developed to support a well-defined API, as well as user program code, which can be written to access files and devices in the same way.
Using the same interface has the drawback that it could be challenging to include some devices' functionality within the parameters of the file access API, which could lead to a loss of functionality or a decrease in performance.
Using the ioctl operation, which offers a general-purpose interface for processes to call operations on devices, could help with some of this.

9. A shared memory model and a message-passing model are the two types of models.
   Strengths and drawbacks of message-passing: messages can be sent directly or indirectly between processes via a shared mailbox. It is simpler to implement for intercomputer communication and useful for sending lesser amounts of data. Its speed isn't as fast as a shared memory approach, though.
   The advantages and disadvantages of shared memory include maximum speed and convenience in communicating. Nonetheless, there are several issues with process synchronization and protection.

10. For a system to be flexible, the mechanism and policy must be kept apart. Only a few parameters may be impacted by a policy change if the interface between the mechanism and the policy is clearly stated. However, if the interface between these two is ambiguous or poorly defined, the system may need to be changed much more deeply.
    To make sure that systems are simple to change, mechanisms and policies need to be kept distinct.

11. Advantages:
    - **Security and Stability**: Since only the most essential services run in kernel space, the attack surface of the operating system is reduced, making it more difficult for an attacker to exploit vulnerabilities. If a user-level process crashes, it will not affect the stability of the entire system.
    - **Modularity and Flexibility**: Services are implemented as user-level processes, making it easier to add, remove, or replace services without affecting other parts of the system. This makes it easier to customize the operating system to meet specific requirements.

    Interaction between User Programs and System Services: In a microkernel architecture, user programs and system services interact via message passing. The client program and service never interact directly. Rather, they communicate indirectly by exchanging messages with the microkernel.

    Disadvantages:

- **Performance:** Message passing between user-level processes can be slower than direct system calls in a monolithic kernel. This can affect the performance of the operating system, especially in high-performance applications
- **Complexity:** The modular design of a microkernel can lead to increased complexity, which can make it more difficult to develop and maintain the operating system.
- **Memory Usage:** Microkernel systems may have higher memory usage compared to monolithic kernel.

12. Each is loadable as needed within the kernel with more flexible.
    - **Optimized System Resources**: LKMs optimize system resources by loading modules on demand. This reduces memory usage and system footprint, enhancing performance by unloading unused modules.
    - **Easy Debug & Testing**: Kernel modules facilitate debugging and testing of specific features or drivers.
    - **Memory Efficiency**: Using modules can save memory, because they are loaded only when the system is actually using them.
    - **Reduced Kernel Rebuilds**: We often have the choice between installing a module in a kernel by uploading it as LKM or committing to it. base kernel. LKM has many benefits in addition to binding to the basic kernel. Another advantage is that we do not need to rebuild your kernel often.
    - **Faster Maintenance and Debugging**: Modules are much faster to maintain and debug.

13. There are several reasons:
    - **Compatibility:** The standard Java API and virtual machine are primarily designed for desktop and server systems, and they are not fully compatible with mobile devices.
    - **Optimization:** Google has created a different API and virtual machine for mobile devices known as the Dalvik virtual machine. The Dalvik virtual machine is optimized for low power handheld devices, i.e., mobile devices. It optimizes the mobile system for battery life, memory, and performance in general.
    - **Execution Process:** Programs intended for Android are first written in Java and later compiled to bytecode and stored in .dex and .odex files. These are Dalvik executable first respectively.
    - **Portability:** The purpose of a virtual machine is to be able to abstract hardware by simulating it. If we make a virtual machine and compile it to work on every possible hardware, we get what originally made Java rise to popularity: write once run anywhere portably.