

TÖL401G - Stýrikerfi

1. Identify services provided by an operating system.

- Error detection
 - Detect errors in the CPU and memory hardware
- Program execution
 - Load a program into memory and run it
- I/O operations
 - Transfer data to and from I/O devices
- File-system manipulation
 - Read, write, create, delete and search files and directories
- Communications
 - Exchange information between processes executing either on the same computer or on different systems tied together by a network
- Resource allocation
 - Allocate resources to multiple users or multiple jobs running at the same time
- Accounting
 - Keep track of which users use how much and what kinds of computer resources
- Protection and security
 - Protect the computer and its data from unauthorized use, either by ensuring that only authorized users are allowed access to the system or by protecting individual files and other system resources against unauthorized access. It also makes sure that concurrent processes running in the system do not interfere with each other.

2. Illustrate how system calls are used to provide operating system services.

The system call API invokes intended system call in the kernel, by passing number (and additional parameters) using a trap assembly instruction, which then performs the requested service and returns control to the caller.

3. Compare and contrast monolithic, layered, microkernel, modular, and hybrid strategies for designing operating systems.

- Monolithic
 - The kernel is a single program that provides all of the services of the operating system. It executes in kernel mode and has access to all of the hardware and data structures of the kernel. It is a single static binary file.
- Layered
 - The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface. With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers. If a layer is changed, the layers above it are unaffected. The major difficulty with the layered approach involves defining the layers.
- Microkernel
 - Moves as much from the kernel into user space. Communication takes place between user modules using message passing. Benefits include: easier to extend a microkernel; easier to port the

operating system to new architectures; more reliable (less code is running in kernel mode); more secure (less code is running in kernel mode). Disadvantages include: performance overhead of user space to kernel space communication; increased size of operating system.

- Modular
 - Instead of having a single, monolithic kernel, the kernel is broken down into separate processes, known as servers. Some of the servers are:
 - file server,
 - process server,
 - and memory server.

The servers invoke system calls as needed by sending messages to other servers. The kernel is not a separate entity, but is a set of cooperating processes in user space.

Benefits include:

- easier to extend a modular operating system;
- easier to port the operating system to new architectures;
- more reliable (less code is running in kernel mode);
- more secure (less code is running in kernel mode).

Disadvantages include:

- performance overhead of user space to kernel space communication;
- increased size of operating system.

- Hybrid
 - Combines the speed of a microkernel with the modularity of a modular kernel. The kernel consists of a microkernel, but the servers are divided into modules, each running in user space. The microkernel provides minimal process and memory management, interprocess communication, and basic synchronization primitives. The kernel modules provide the file system, device drivers, networking, and other operating system functions. The kernel modules can be loaded and unloaded dynamically, making it easier to extend the kernel.

4. Illustrate the process for booting an operating system.

- The BIOS is located in ROM on the motherboard. It is the first code that is executed at start-up and is responsible for locating and loading the operating system kernel software.
- The BIOS performs a power-on self-test (POST) to ensure that all of the hardware components are present and operational.
- The BIOS then loads the first sector of the boot device (usually a hard disk) into memory and transfers control to that code.
- This code is known as the master boot record (MBR).
- The MBR locates the active partition on the hard disk and loads a copy of its first sector into memory.
- This code is known as the volume boot record (VBR).
- The VBR loads the operating system kernel into memory and transfers control to it.
- The kernel initializes the rest of the operating system.
- The kernel creates a process for the init program, which is the first user-level process.
- The init program then starts other processes, such as daemons, which are background processes that provide services to the system.
- The init program then waits for the system to shut down or reboots the system if instructed to do so.