Faculty of Engineering & Technology

Electrical & Computer Engineering Department

OPERATING SYSTEMS - ENCS3390

Essay for assignment 1 : Operating System Overview and Structures Analysis

---

Prepared by : Sondos Farrah **ID :** 1200905

Instructor: Bashar Tahayna

Section : 1

Date : 14/4/2023

Assignment Objective:

To familiarize yourself with the basic organization of computer systems and the major components of operating systems, and to provide an overview of the different types of computing environments. To understand the services an operating system provides to users, processes, and other systems, and to explore the various ways of structuring an operating system.

Table of Contents

Table of Figures

Essay:

Computer system can be divided into four components, First Hardware that's the basic organization of a computer system is the processing unit, memory unit, and input-output devices, The processing unit controls all the functions of the computer system. It is the brain of the computer e.g. CPU. The memory unit consists of two units. One is an arithmetic unit and the other is a logic unit. Input devices are those devices through which end-users can send messages to computers e.g. keyboard, mouse, etc. Output devices are those devices through which end-users get output from computers e.g. monitors.[1]. Second, Operating system that's Controls and coordinates use of hardware among various applications and users. Third, Application programs – define the ways in which the system resources are used to solve the computing problems of the users. Final , users such like people , machines or other computers[2].

Operating system provide an environment for execution of programs and services to programs and users such as provides a user interface, including graphical user interfaces (GUIs) and command-line interfaces (CLIs), to enable users to interact with the system and applications. Manage program execution and threads. Provides a file system that allows users and applications to organize and store data on disk drives. Resource allocation when multiple users or multiple jobs running concurrently, resources must be allocated to each of them and many other services[2].

The main components of an operating system are: kernel (The one program running at all times on the computer) , API (Application Programming Interface which system calls Mostly accessed by) , user interface & file system, hardware devices (provides basic computing resources CPU, memory, I/O devices) and device drivers (for each device controller to manage I/O)[2].

kernel is a fundamental component of an operating system (OS). The kernel is responsible for managing hardware resources such as memory, CPU, input/output devices, and networking. It allocates memory to programs, manages processes and threads, and schedules tasks to be executed on the CPU. The kernel also provides a layer of security by controlling access to system resources and enforcing permissions. In addition, the kernel provides system calls, which are a set of programming interfaces that allow applications to interact with the operating system. System calls provide access to file systems, inter-process communication, and other OS services[2].

Computing environments refer to the technology infrastructure and software platforms that are used to develop, test, and run software applications. There are several types of computing environments, including: Multiprogramming and Timesharing[3]. Multiprogramming (Batch system): Single user cannot keep CPU and I/O devices busy at all times. A subset of total jobs in system is kept in memory so Multiprogramming organizes these jobs (code and data) so CPU always has one to execute. Timesharing (multitasking): CPU executes multiple jobs by switching them frequently that the users can interact with each program while it is running. This environment provides direct

communication between user and system and allows many users to share the computer simultaneously[2].

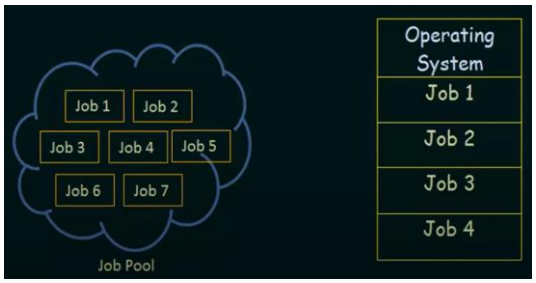

Figure 1: Multiprogramming

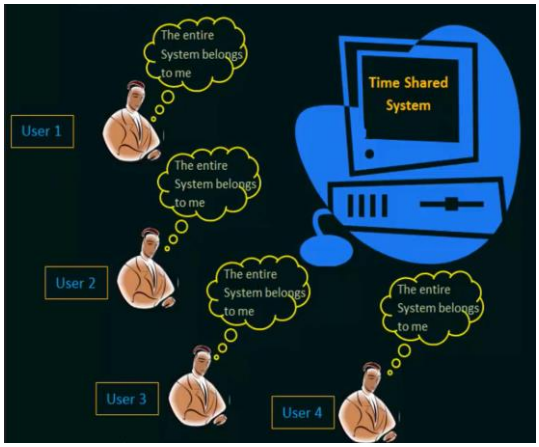

Figure 2:Timesharing

There are various ways to structure OS:

- **Simple structure (Monolithic):** Such operating systems do not have well defined structure and are small, simple and limited systems. The interfaces and levels of functionality are not well separated. MS-DOS is an example of such operating system. In MS-DOS application programs are able to access the basic I/O routines. These types of operating system cause the entire system to crash if one of the user programs fails[4].

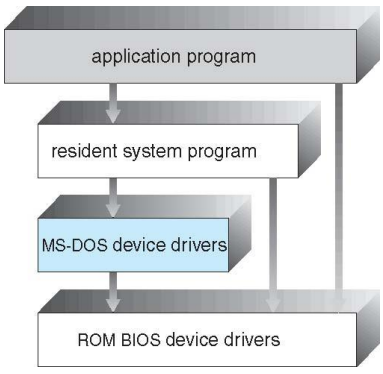

Figure 3: simple structure

- **More complex:** limited by hardware functionality, the original UNIX operating system had limited structuring. The UNIX OS consists of two separable parts: Systems programs and The kernel which Consists of everything below the system-call interface and above the physical hardware and Provides the file system, CPU scheduling, memory management, and other operating-system functions; a large number of functions for one level[2].

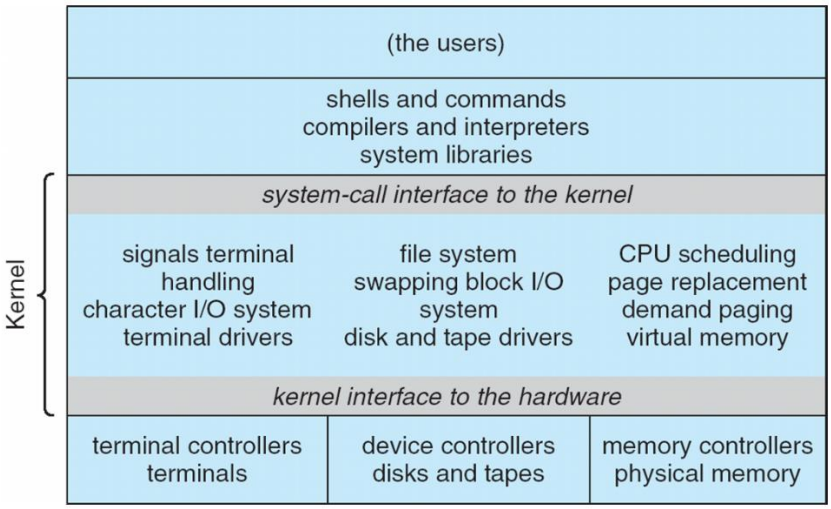

Figure 4: Non simple structure

- **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. The functionalities of each layer are separated in this method, and abstraction is also an option. Because layered structures are hierarchical, debugging is simpler, therefore all lower-level layers are debugged before the upper layer is examined. As a result, the present layer alone has to be reviewed since all the lower layers have already been examined[2].

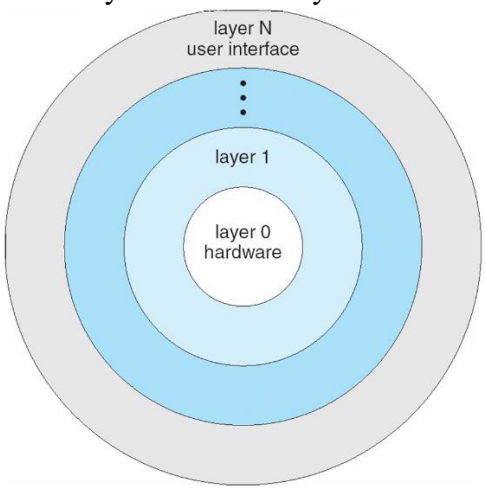

Figure 5: Layered structure

- **Microkernel:** This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs. This result in a smaller kernel called the micro-kernel. Advantages of this structure are that all new services need to be added to user space and does not require the kernel to be modified. Thus, it is more secure and reliable as if a service fails then rest of the operating system remains untouched. Mac OS is an example of this type of OS[2].

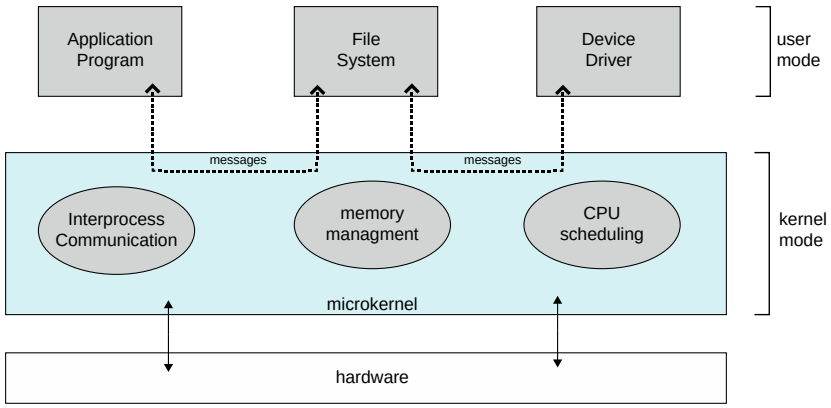

Figure 6: Microkernel structure

- **Module:** It is considered as the best approach for an OS. It involves designing of a modular kernel. The kernel has only set of core components and other services are added as dynamically loadable modules to the kernel either during run time or boot time. It resembles layered structure due to the fact that each kernel has defined and protected interfaces but it is more flexible than the layered structure as a module can call any other module. For example Solaris OS is organized as shown in the figure below[4].

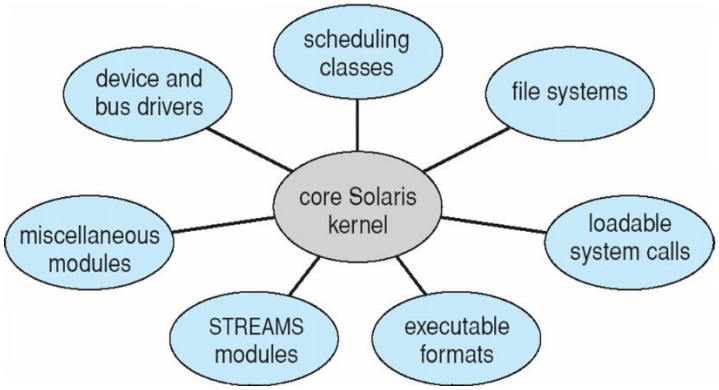

Figure 7: Module structure

- **Hybrid:** In practice, very few operating systems adopt a single, strictly defined structure. Instead, they combine different structures, resulting in hybrid systems that address performance, security, and usability issues. For example, both Linux and Solaris are monolithic, because having the operating system in a single address

space provides very efficient performance. However, they are also modular, so that new functionality can be dynamically added to the kernel. Windows is largely monolithic as well (again primarily for performance reasons), but it retains some behavior typical of microkernel systems, including providing support for separate subsystems (known as operating-system personalities) that run as user-mode processes. For example **Mac OS X Structure, IOS, Android**[5].

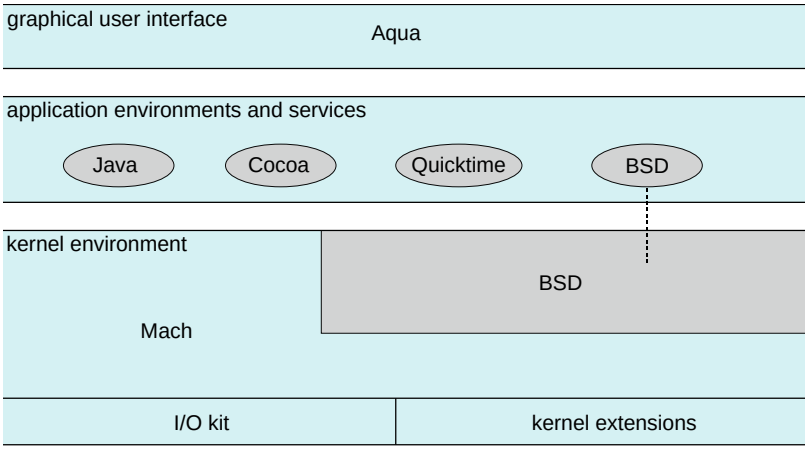

Figure 8: Mac OS X structure

Installing an operating system involves preparing the installation media, such as a CD/DVD, USB flash drive, or network share. Once the media is ready, the computer is booted from it and the user selects the language, time zone, and other basic settings. They also select the disk or partition on which the OS will be installed. The installer then copies the necessary files from the installation media to the disk, and configures system files and registry settings. After the installation, the system is rebooted and the user can log in and customize the OS. To make the OS available for use, it must be booted by loading the kernel into memory and executing it with a bootstrap program, which initializes the system, runs diagnostics, and loads the OS. Some systems store the entire OS in ROM, while others store the bootstrap loader in firmware and the OS on disk. The bootstrap loader reads a block from disk and executes it, which then loads the OS kernel into memory and starts its execution[5].

References:

1. What are the Basic Organization of a Computer System? (unacademy.com)
2. Slides for chapter 1 and 2
3. Computing Environments - GeeksforGeeks
4. Different approaches or Structures of Operating Systems - GeeksforGeeks
5. 9th edition of the OS Concepts book