

Technical university of Košice  
Faculty of Electrical Engineering and Computer Science  
Department of Computers and Information Sciences

# Operating systems structure

# 1 Introduction

In past few decades, there was a huge increase in a computer segment. Strange machines with silicon brains that were before used only inside huge companies and by homemade enthusiasts, called geeks.

Over the last 30 years, industry has changed so massively, that nowadays, almost everyone owns a “high-end” computer (compared with the machines from 5-10 years ago) that can fit in your pocket, called the smartphones.

A huge minority of people are using a computer (desktop or laptop) on a daily basis, simply to get things done, or for an entertainment and fun.

In this paper, I’m going to look at the very key element of a computer software, that changes hardware from a piece of silicon to a useful “intelligent” machine - the operating system.

## 2 What is operating system?

A computer is a massively complicated electronic device. It’s designed and created to be a hard worker.

But on its own, it’s just a stupid dumb machine that does nothing on its own.

To become a really powerful machine capable of “everything” it has to have instructions, to tell it what to do.

So people, called programmers, are experts, that know how to program a computer. By writing and instructions (code), we can use the power of a computer to do what we want.

In a meantime, some instructions repeated over and over, so it was reasonable to extract those instructions to its own separate abstraction layer which started to form the first instance of an operating system. That meant, that we no longer have to type everything from scratch. Instead, we can have an operating system to implement all the general things, and build our programs on top of it.

It’s hard to define and operating system in one general definition, but there are two main ones, used to describe it.

### 2.1 The operating system as an extended machine

An operating system is a program that controls the execution of application programs and acts as an interface between applications and the computer hardware. [2]

### 2.2 The operating system as a resource manager

An operating system is a software that manages the computer hardware. It provides appropriate mechanisms to ensure correct operation of the computer system. [1]

So with a bit of an exaggeration, we could write that operating system is a software that ensures that applications running on the computer on top of it are running correctly.

### 3 Marketshare and categorization

In those modern days that we are living, computers and operating systems are everywhere around us. Thus, we got used to them, so we don't take them as something unique as it was before 10-20 years ago.

If we want to somehow classify every device, we can come up with three categories:

- **Desktop** operating systems (used on desktops, laptops and servers)
  - Designed for mouse and keyboard experience, architected for the heavy load of work. They had been there for more than 20 years, and they underwent huge development change from both, hardware and software perspective. Every year, there is some increase in speed, space size and raw power in general, so in comparison with older machines, today's machines are beasts. Which opens up possibilities for software, since it has more resources to use to its advantage. Years of development and usage results in a stable robust piece of software.
- **Mobile** operating systems (used on smartphones, tablets, and mobile TVs)
  - Devices (and operating systems) that are specialized on touch experience. Simple and easy to use user interface is the key. They are designed for everyday general use, like web browsing and media consuming. The biggest player in the game is currently Android (supported by Google) followed by iOS (from Apple) and small market percentage remained for Windows Phone (Microsoft) and other small players.
- **Specialized** operating systems (used on e.g. game consoles and virtualization host machines)
  - Operating systems, that are intended and tuned for "special" hardware, when performance is priority number one. In game consoles market, there are three main players. Playstation, XBox, and Nintendo.

### 4 Operating system architecture

Since the operating system is a huge project (software) with a lot of responsibilities and functions it has to support, it needs to have a number of sections/layers/abstractions to achieve its goals. At the bottom, there is the most hardware-specific layer, and every other layer is built on top of previous ones.

- Hardware depend layer
  - Layer, that is strongly specific to a concrete hardware architecture it's running on. Every different hardware has to have its own layer, that knows how to communicate with it. That layer is converting an "ugly" and complicated hardware specific commands and operations to a clean interface, which higher layers know how to talk to.
- Kernel (core)
  - Layer, that is the core of an operating system. Its responsibility is to achieve, that everything will work as expected, without any issues. It manages the hardware resources and provides an ability for higher layers to access and use them an easier way, without the need to actually know how things are done.

- Device drivers
  - Since there are dozens of hardware accessories, it’s not possible for the core of an operating system to know about everything. So we need some way to teach it, on the go. To fulfill that need, drivers were created. When a new hardware device is connected to the computer, an operating system doesn’t know how to talk to it. In that situation, if a concrete driver is present, it’ll use it as a “translator” to communicate with a device.
- Service layer
  - We, humans, prefer to think in abstractions. For example, we like to have directories and files and have an easy way to tell how many spaces is left. We don’t really care about how it looks under the hood and how that files and directories look like on the disk. For that purpose, service layer was created. It changes a command like “write 25 bytes to a disk at position 23462 from the beginning” to “create a file for me with name XX and with that content”. There are lot’s of services here, from file system mentioned in the example to a virtual machine manager, job manager and so on.
- API (syscall layer)
  - Since talking straight to a device is mostly an overkill for a general application, another layer was created to simplify that needs. Since the majority of applications are using files (and other general operations with directories, networking, processes and so on), a lot of uselessly low-level code is saved, resulting in simpler applications that can be created in a shorter period of time. Also, that means that programmers can specialize whether they want to create low-level stuff or high-level applications for the consumer.
- Command line interface and graphical user interface
  - In order to tell the computer what to do (from the user perspective), we need to have some way to give it our commands. In older days, computers were running in text modes, so it was done by so-called command line interface. It is straightforward and fast, but for nontechnical people not so easy. So it naturally evolved into something easier, more intuitive, designed for everyone. The graphic user interface, or GUI in short, is designed to be controlled not only with the keyboard but also with mouse and touch.
- Applications
  - Those layers are representing applications, that we are using every day. Web browser, word processor, music player and so on. They are designed to do a concrete thing. To get things done, entertain or just to kill boredom (like some games).

## 5 Conclusion

Computers are enjoying huge popularity not only among the young people but also older generations found it’s useful features. Almost everyone from the modern civilizations owns some kind of computer device. We use them daily for work, entertainment or education and for fun. We joined together our “real” lives with those “virtual” ones and we’ve learned to benefit from our devices.

Since we use them a lot, it's useful to have some kind of general knowledge how it looks like under the hood. The operating system is a fairly complicated device, but with a bit of simplification, it's easy to describe it, without much overwhelming.

## References

- [1] *Modern Operating Systems: Global Edition*. Pearson, 4th revised edition edition, 1823.
- [2] William Stallings. *Operating Systems: Internals and Design Principles (7th Edition)*. Prentice Hall, 7 edition, 3 2011.