

# MicroKosm Operating System

Filippo Mutta

2023-09-01

# Contents

<b>1</b>	<b>Introduction to MicroKosm</b>	<b>2</b>
1.1	What is a MicroKosm? . . . . .	2
1.1.1	What is a microkernel? . . . . .	2
1.1.2	What is a multitasking kernel? . . . . .	3
1.1.3	What is a real-time kernel? . . . . .	3
1.1.4	What is a virtualizing kernel . . . . .	4
<b>2</b>	<b>Kernel structure</b>	<b>5</b>

# Chapter 1

## Introduction to MicroKosm

### 1.1 What is a MicroKosm?

MicroKosm, also called MicroK for short, is a real-time multitasking virtualizing microkernel created in such a way to be compatible with almost any machine and any usage.

It is meant to be expandable with modules developed along the MicroKosm Driver Interface (MKMI). For more information about the topic, please refer to the MKMI specification.

#### 1.1.1 What is a microkernel?

A microkernel is the near-minimum amount of software that can provide the mechanisms needed to implement an operating system. These mechanisms usually include low-level address space management, thread management and inter-process communication. Traditional operating system functions, such as device drivers, protocol stacks and filesystems, are typically removed from the microkernel itself and are instead run in user space.

Microkernels were a very hot topic in the 1980s, but slowly died down in the decades since, mostly because of their inferior performance. Major work on microkernels had largely ended by the early 2000, with only minor projects being developed in the following years.

Microkernels have the major advantage of being far more flexible and secure than normal monolithic or hybrid kernels because of their strict separation of powers. This allows them to withstand most crashes and continue to operate normally.

Some examples of microkernels, outside of MicroK, are: QNX, L4, Minix and Zircon. Apart from Zircon, the others have started development in the 1990s or earlier.

### 1.1.2 What is a multitasking kernel?

A multitasking kernel is an operating system that supports the concurrent execution of multiple tasks called processes. Multitasking operating systems have existed since the early days of modern computing and have become mainstream in the 1980s.

There are however two distinct types of multitasking: cooperative multitasking and preemptive multitasking. In cooperative multitasking, a process voluntarily yields control to the next process in line, while in preemptive multitasking the kernel itself manages when a switch occurs. Cooperative multitasking has largely been abandoned in favour of the other more reliable version, but it's still important in situations where speed is crucial.

MicroKosm supports preemptive multitasking for user tasks and modules, and tries to eliminate cooperative multitasking in the kernel itself to avoid situations where a thread does not yield control and hangs the whole system.

### 1.1.3 What is a real-time kernel?

A real-time kernel is an operating system that supports real-time computing. Real-time computing is the computer science term for hardware and software systems subject to a "real-time constraint". Real-time programs must guarantee response within specified time constraints, often referred to as "deadlines". Most real-time programs are used in safety-critical and mission-critical situations, and must be executed flawlessly.

There are various types of real-time systems, categorized by their response when missing a deadline:

- Hard: missing a deadline is a total system failure.
- Firm: infrequent deadline misses are tolerable, but may degrade the system's quality of service. The usefulness of a result is zero after its deadline
- Soft: the usefulness of a result degrades after its deadline, thereby degrading the system's quality of service.

MicroK, with its real-time scheduler, is capable of managing all types of real-time programs. It is also possible to run normal programs and modules together with soft or firm real-time programs, but not with hard real-time programs.

Because of their strict timing requirements, their execution is only possible in kernel-mode with only specialized real-time kernel-mode drivers. A hard real-time MicroK system morphs the microkernel in what basically is a monolithic kernel, allowing for maximum performance.

Instead, with the other two types of real-time software the system executes those programs alongside normal programs, giving them the RT (Real Time) priority in the scheduler, effectively creating a hybrid kernel. This even allows to run some real-time programs in user-mode, executing what is almost a true microkernel, with all the obvious security benefits brought by this model.

#### **1.1.4 What is a virtualizing kernel**

A virtualizing kernel is an operating system that supports the execution of hardware-accelerated virtual machines.

## Chapter 2

# Kernel structure