Developers Conference #DevConMU
@ the Voila Hotel & Flying Dodo

An introduction to The seL4 MicroKernel

Learn about the secure and performant MicroKernel

Pritvi Jheengut @zcoldplayer

13th April 2019

Copyleft License Attribution

Made with love using beamer, LaTeX and git.

You can view at Introduction to seL4

Who Am I

Geek@Slackware
twitter @zcoldplayer
zcoldplayer xmail Website
Work : SMTT@Meteorological.Services.mu
Active in many User group LUGM, MMC, MSCC, FECM, GDG_MU
and several other hackathons
Passionate about how and why things work.
Fervour Advocate of Free Libre and Open Source Software.

Who are you

Would you mind tell me who you are?

Some hints:

1. @twitter_handle
2. where you work
3. email you want to share
4. Hobbies
5. purpose and expectations of this session

Community Groups in Mauritius

Healthy growth of Community Groups in Mauritius

This turn of the century has seen an uprising of Community groups in Mauritius in the field of the Digital World. The diversity has helped the exchange and sharing of innovative ideas, experience, bleeding edge technology, upcoming events, conferences in the Digital Island of the Republic of Mauritius.

This is a list of some of the active communities in Digital Mauritius.

- Linux User Group Meta, LUGM
- Mauritius Software Craftsmanship Community, MSCC
- Mauritius Makers Community, MMC
- Front-End Coders Mauritius, FECM
- PHP User Group of Mauritius, phpMauritiusUG
- Symfonymu
- Google Developers Group Mauritius, GDG_M
- Digital Marketing Mauritius

Underlying OS basics

What Do You Understand By An Operating System ?

- Provide The User Utilities, Programs to perform jobs
- Manage the Hardware on which it is running
- Manage resources automatically
- The OS and the hardware together is called a Computer

An Operating System contains a Kernel, the Kernel is not an OS.

Underlying OS basics

What Do You Understand By An Operating System ?

- Provide The User Utilities, Programs to perform jobs
- Manage the Hardware on which it is running
- Manage resources automatically
- The OS and the hardware together is called a Computer

An Operating System contains a Kernel, the Kernel is not an OS.

Underlying OS basics

What Do You Understand By An Operating System ?

- Provide The User Utilities, Programs to perform jobs
- Manage the Hardware on which it is running
- Manage resources automatically
- The OS and the hardware together is called a Computer

An Operating System contains a Kernel, the Kernel is not an OS.

Underlying OS basics

What Do You Understand By An Operating System ?

- Provide The User Utilities, Programs to perform jobs
- Manage the Hardware on which it is running
- Manage resources automatically
- The OS and the hardware together is called a Computer

An Operating System contains a Kernel, the Kernel is not an OS.

The importance of the Kernel

In Computer science

the Kernel is the bridge between
the hardware and user space Software
as shown in a hierarchical protection domain system.

Multi functionality of the Kernel

The Kernel in general manages resources such as :

- Processing and Memory units
- Process Capabilities and Virtual addressing
- Input as well as Output Devices
- Virtualisation and by extension Containers
- Access Control lists, Security and Cryptographic Mechanisms

Hardware abstraction layers of the Kernel

The Kernel provides hardware abstraction

for the underlying user space software using hierarchical protection domains.

Hardware abstraction is crucial in cases such as

- portability
- security frameworks
- code reuse
- forward and backward version compatibility
- fault tolerance

A Kernel once built, can be run on several different types of platform but of similar architectures.

The Multi Tasking Approach using a Kernel

Scheduling processes

In a multi tasking environment, process scheduling is primordially managed by the Kernel.

A process contains both

- Data
- Instruction

Mechanisms provided by the Kernel manage processes using

- Overall performance or system load
- CPU and Memory as well as Networking usage
- Input and Output Device Isolation
- ACL's and Capabilities

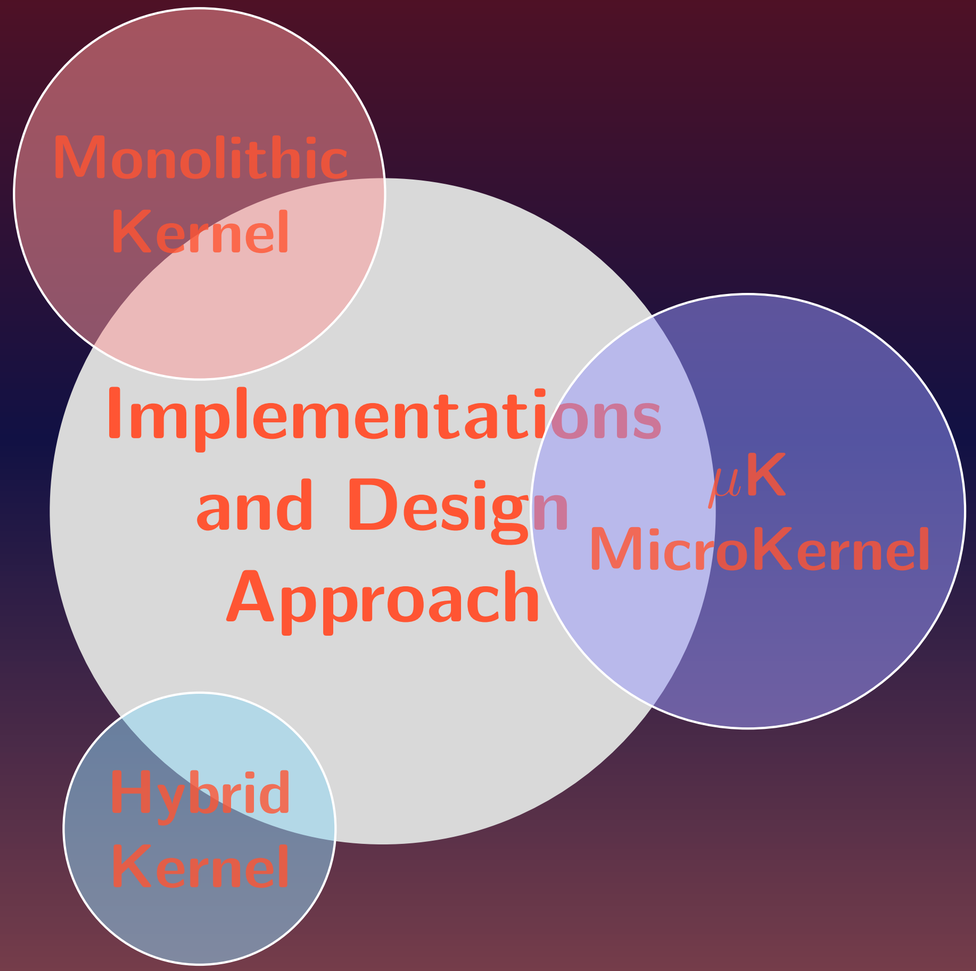
Example of Scheduling Processes

A prime example of the Kernel's role in this context is to separate processes resources so that

- one process do not overwrite or read another's process data either in
  1. Volatile Memory or
  2. Persistent Storage
- one process is not utilising greedily the CPU, network, memory continuously at the expense of other processes
- regulate Input and Output Devices with the respective process
- The overall performance of the system is within the permissive set limits

There exist several implementations of all these Mechanisms, hierarchical protection domains, Capabilities set with the hardware or provided by the Kernel or inherited by the processes through software libraries provided by the Operating System as a whole.

The major Kernel Design Implementations



The Most famous Monolithic Kernel in the world

`https://www.Kernel.org` : The home of the Linux Kernel

Linux is GPL'ed Freely licensed Monolithic Kernel developed by Linus Torvalds who also is the creator of git.

Inspired by the μKernel Minix as a Unix Clone,

Linux was initially publicly released in 1991 primed for the relatively cheap x86 based IBM clone PC.

Being Monolithic, Linux contains all the previously mentioned Capabilities and Mechanisms inside the Kernel all running within the Kernel Space

The Most famous Monolithic Kernel in the world

Linux is Everywhere

Our Benevolent Dictator, Lord Linus dreamed of conquering the world and he achieved it.

- Most Smart
  1. Phones
  2. TV
  3. Cars
  4. Guns
  5. Camera

  runs Tizen, Android and Firefox OS/KaiOS or in another Form.
- All of Top 500 Supercomputers runs Linux since two years
- Ported to most architectures
- Powers 80% - 90% of the Internet and most of the cloud
- Intensive and highly complex CGI are produced by Linux Clusters

Except the Desktop

even steam accounts for almost 1%

less famous Monolithic Kernels

Other dominant Monolithic Kernels

- The BSD Family :: FreeBSD, NetBSD, OpenBSD
- MS-DOS / FreeDOS
- Unix System V based OS :: AIX, HP-UX, Solaris

FreeBSD is Open Licensed as such as the other BSD's

Therefore, it can used in closed source hardware such as

- Routers, Switches, Firewalls
- PS4
- Windows Network stack
- Components of them are used in other software

The Hierarchical protection Domain Design of Monolithic Kernel

Use case of the Hierarchical protection Domain

Improves granularity in addition to the robustness of protective mechanisms by separating all Kernel services in Kernel space and the remaining unprivileged services in user space.

Inside the Kernel space or Supervisor mode, everything is possible

The Kernel provides System Calls for user software in order to permit them to perform privileged operations

Inside the user space, the protected domain permits only safe operations and unprivileged actions are permitted

When a user space process tries to directly access a Kernel space process, a protection fault exception is reported by the Kernel.

If Monolithic Kernels Work, then where is the Issue?

In general, Monolithic Kernels are

- very big, sometimes in the dozens of Megabytes
- difficult to Maintain
- licensing issues with third party partners, vendors
-
- hard to detect and eliminate bugs in general
- not much extensible
- resource and time consuming for new compilations

To address these issues, one solution major Operating System vendors have adopted is Hybrid Kernels.
The other solution is μKernels.

Some common Hybrid Kernels in the Market

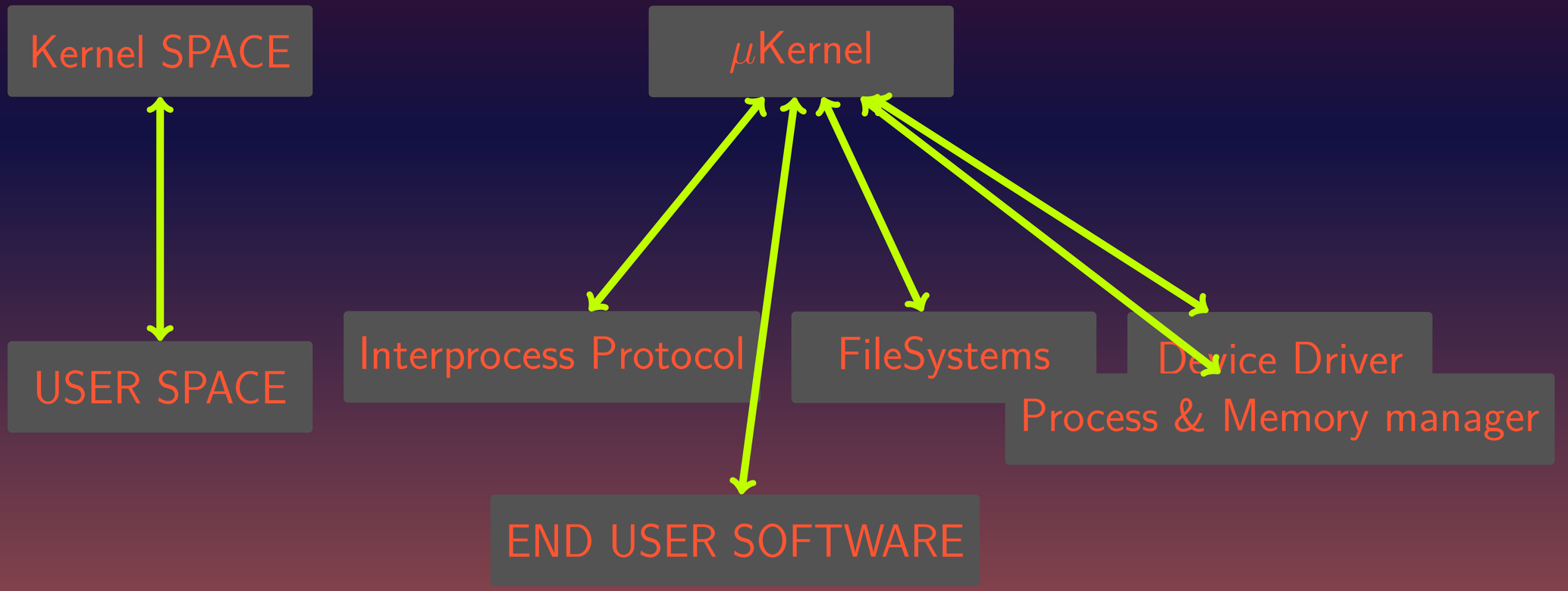
Some Hybrid Kernels include

- Apple Kernel
- Microsoft Windows NT kernel

The μKernel

What is a μKernel ?

By moving all non-essential services, protocol stack, file system management, device driver stack out of the Kernel space into the user space, the size of the Kernel code reduces dramatically such that certain Kernels can fit into the Level one cache of CPU's.



The μKernel

The μKernel advantages with respect to Monolithic Kernels

This brings some advantages such as ::

- smaller size means code gets to run more rapidly
- security is enhanced by having minimal dependencies and code review is improved too
- Maintenance becomes easy on the long run
- Feature extension is not an issue at the user space
- easy to have third party vendor components with complying licenses
- small code means bug detection is not that hard at all
- compilation of a new Kernel involves minimal resources and takes a small time to fulfil

The Mach μKernel Family

Mach is a kernel developed at Carnegie Mellon University

to support operating system research, primarily distributed and parallel computing.
Mach is often mentioned as one of the earliest examples of a microkernel

IPC overhead is a major issue for Mach 3 systems

This issue has been one of the reason μKernel has a bad reputation in the industry.
A single-side of a syscall took 20μs under BSD and 114μs on Mach running on the same system.

The L4 μKernel Family

L4 created by Jochen Liedtke is a family of second-generation microkernels

- useful to implement Unix-like Operating Systems
- designed from the start for
  1. high performance
  2. improving security
  3. isolation
  4. robustness

About seL4

sel4 is a μKernel.

http://sel4.systems/

The world's first operating-system kernel with an end-to-end proof of implementation correctness and security enforcement is available as open source.

sel4 has been proved with

- Functional correctness
- Integrity and information flow security
- Compilation correctness
- Verified worst-case execution time

More information about seL4 is available here and here.

github and seL4

seL4 Howto and repositories

github

- How to start with seL4
- utils for seL4
- Userland Components and Drivers
- CAmkES

CAmkES

CAmkES is a component architecture for microkernel-based embedded systems is a software development and runtime framework for quickly and reliably building microkernel-based multiserver systems.

seL4 in practice

Hardware support of seL4

- Odroid XU4
- Raspberry Pi
- Beagle Board

Data61's Trustworthy Systems Research Group is one of the companies who actively develop seL4.

Questions

QUESTIONS!