betabit {...}

samen bijzonder maken;

Een .NET-besturingssysteemtoolkit

Discovering Cosmos

Sijmen J. Mulder

Agenda

Boek 1
Cosmos: a very short introduction

Boek 2
Modern Operating Systems

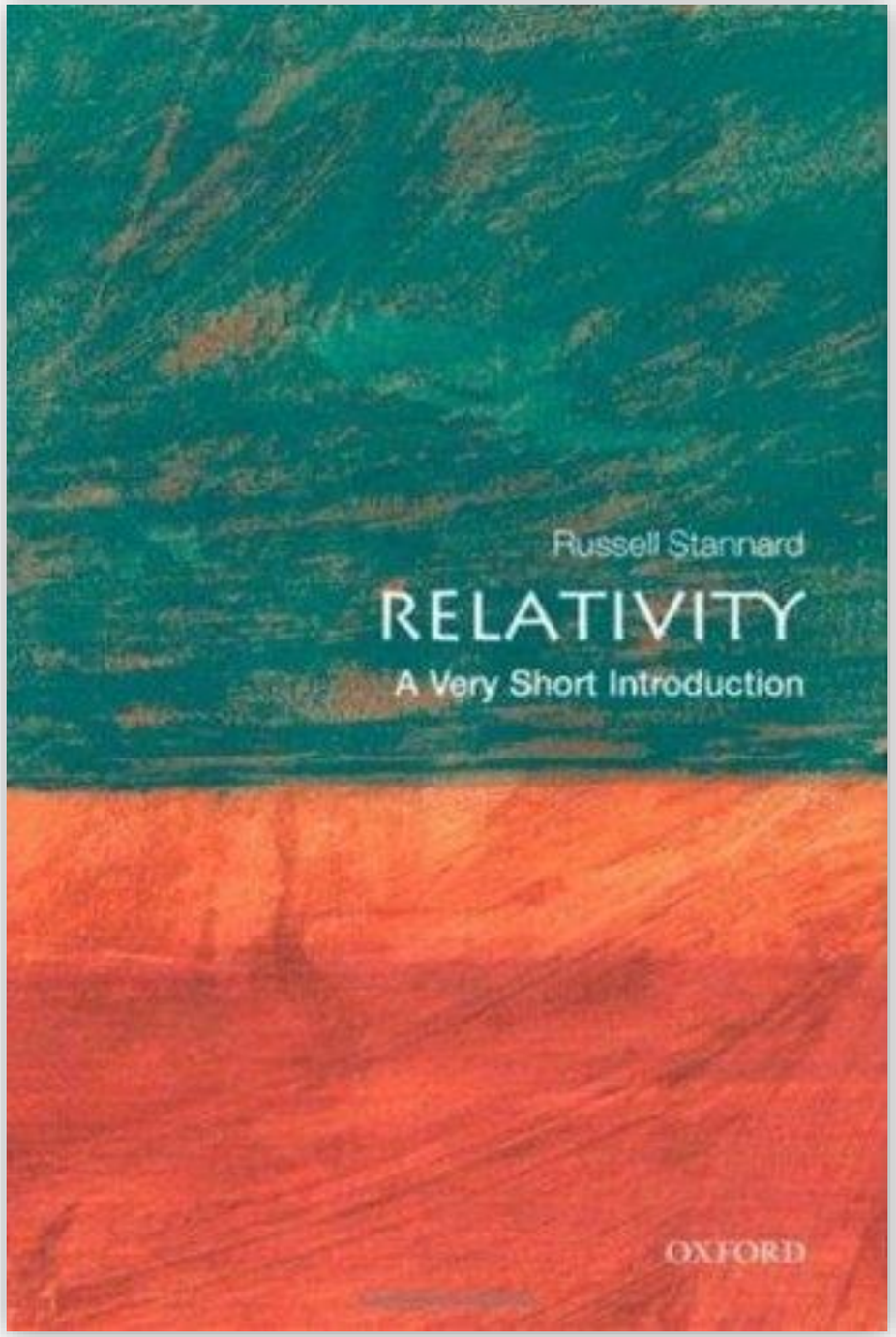
Pauze

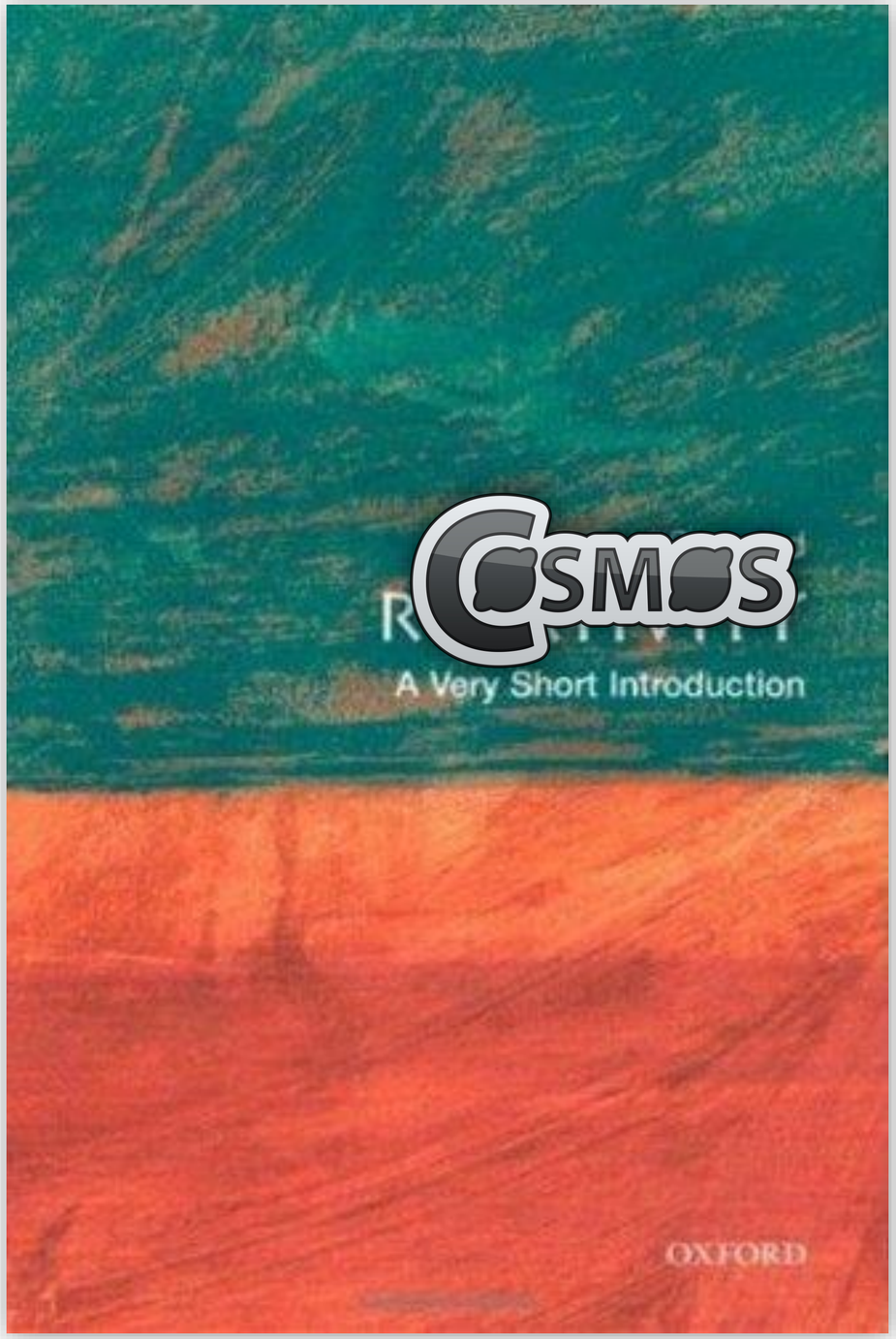
Boek 3
The Design and Implementation of the Cosmos Operating System

Boek 4
Cosmos in Context

betabit {...}

samen bijzonder maken;


Russell Stannard
RELATIVITY
A Very Short Introduction
OXFORD


COSMOS
A Very Short Introduction
OXFORD

Cosmos

- Toolkit voor besturingssystemen
- (Bijna) compleet .NET
- Open source (BSD-3)

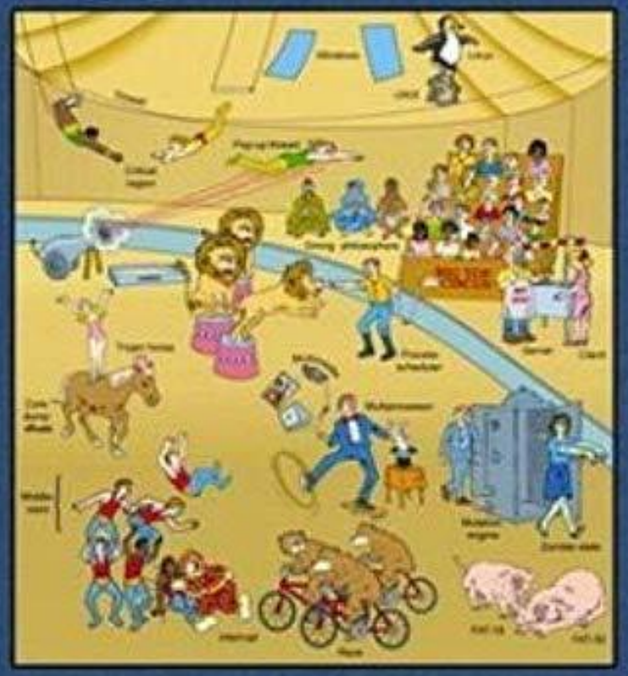
https://www.gocosmos.org

Demo

“Waarom”

MODERN OPERATING SYSTEMS

SECOND EDITION



Andrew S. Tanenbaum

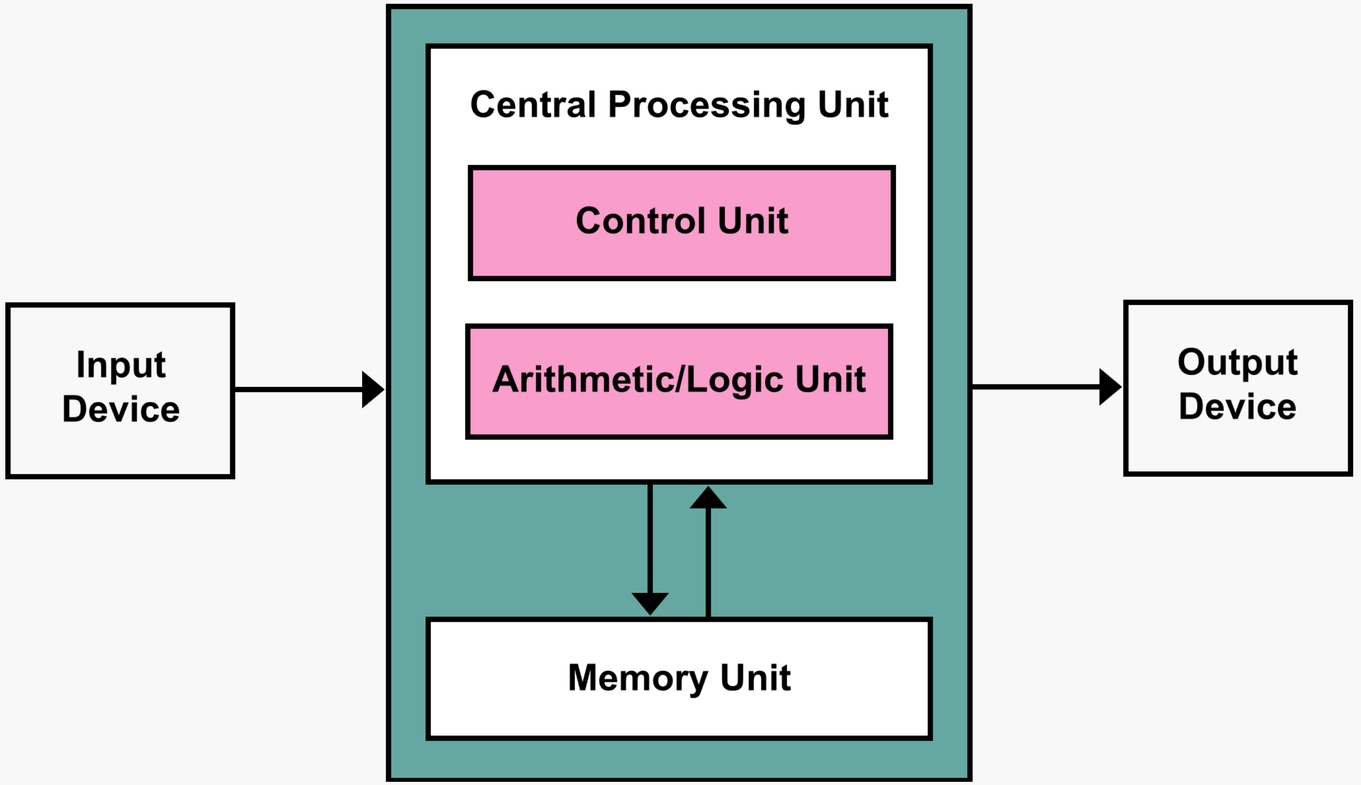
Besturingssysteem

- **Drivers** voor hardware en protocollen
- **Interface** voor gebruiker
- Voert **applicaties** uit
- **Diensten** voor applicaties

Computer

- **Processor** voert instructies uit
- **Geheugen** bevat data (o.a. instructies)

Welke instructies?



https://commons.wikimedia.org/wiki/File:Von_Neumann_Architecture.svg

betabit

Firmware

- Op PC: **UEFI**
- Vindt en start besturingssysteem

Hoe dan?

UEFI

- **Drivers** (FAT32, WiFi, enz.)
- **Interface** voor systeemconfiguratie
- Voert **besturingssystemen** uit
- **Diensten** voor het besturingssysteem

~~UEFI~~ Besturingssysteem

- **Drivers** (FAT32, WiFi, enz.)
- **Interface** voor systeemconfiguratie
- Voert ~~**besturingssystemen**~~ applicaties uit
- **Diensten** voor ~~het besturingssystemen~~ applicaties

Diensten

- Procesbeheer
- Geheugenbeheer
- Communicatie

...en nog veel meer

Uitdagingen procesbeheer

- Multitasking (*scheduling*)
- Isolatie tussen processen
- Coördinatie gedeelde bronnen

System calls

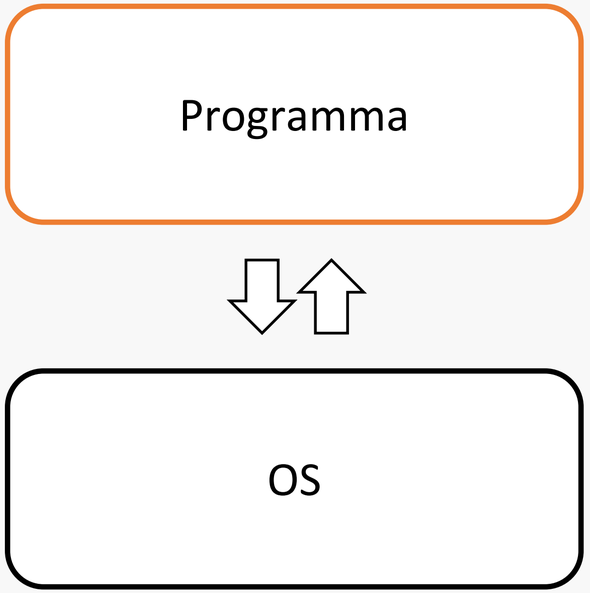
- OS en software in eigen domeinen
- Speciale functiecalls naar OS-domein

Voordelen:

- Sterke isolatie
- Privileges OS bewaakt door processor

Nadelen:

- Dure contextswitch



OS as library

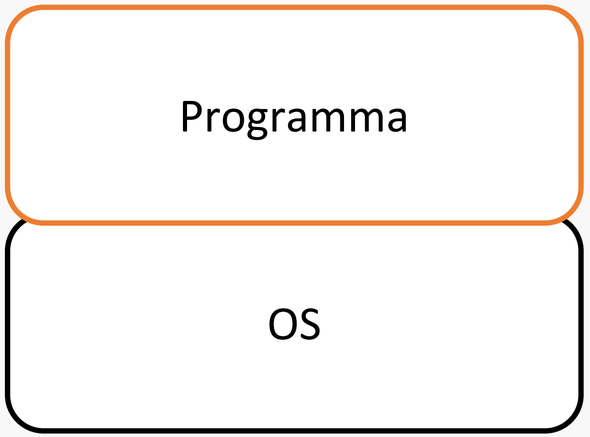
- OS en software in zelfde domein
- Reguliere functiecalls

Voordelen:

- Eenvoud
- Snelheid*

Nadelen:

- Geen harde scheiding OS en programma's



Uitdagingen geheugenbeheer

- Meerdere programma's tegelijk laden
- Gote werksets
- Isolatie tussen processen

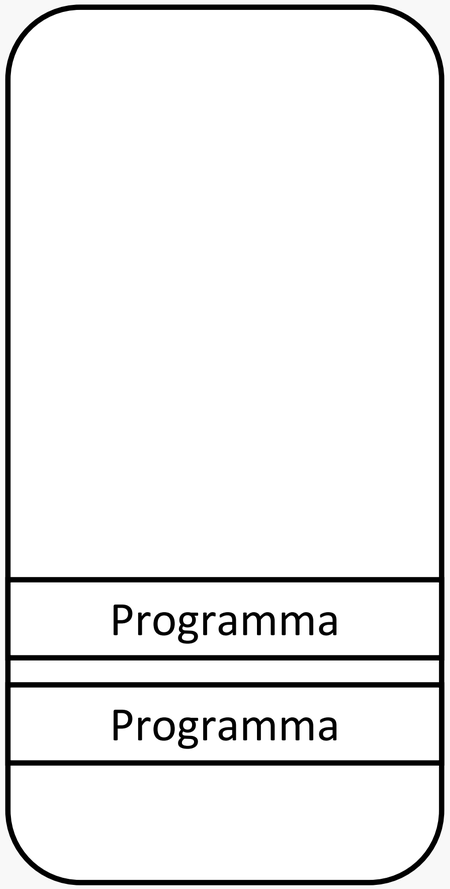
Lineair geheugen

Voordelen:

- Eenvoudig
- Geen indirectie, dus snel

Nadelen:

- Vereist positie-onafhankelijke code
- Geen isolatie tussen programma's
- Geen werksets groter dan fysiek geheugen



Toegewezen geheugen

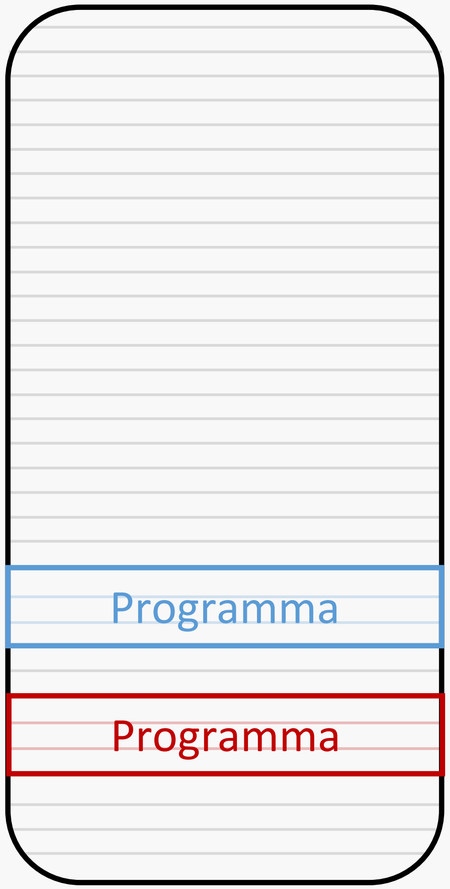
- Paginering
- Elke pagina heeft een eigenaar
- Sleutel huidig programma in beveiligd register

Voordelen:

- Geen indirectie
- Isolatie

Nadelen:

- Vereist positie-onafhankelijke code
- Geen werksets groter dan fysiek geheugen



Virtueel geheugen

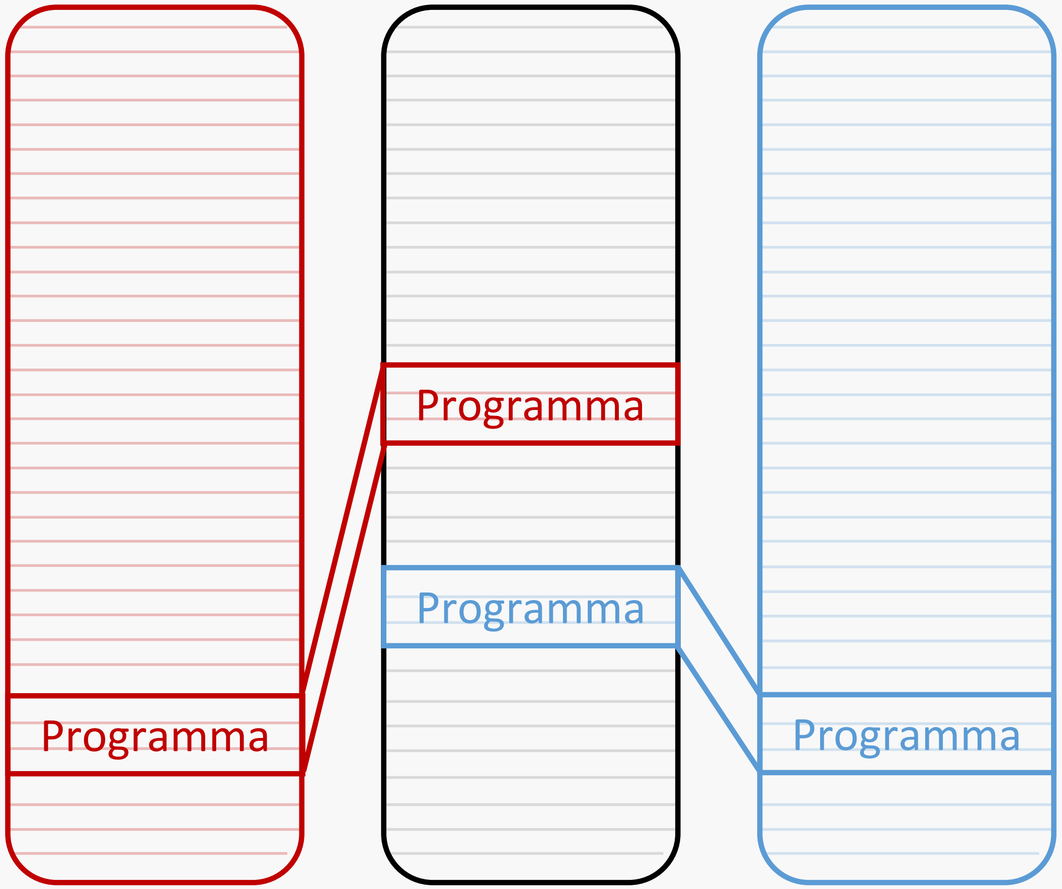
- Paginering
- Virtuele adressen

Voordelen:

- Eigen ruimte
- Grote werksets

Nadelen:

- Indirectie
- Complexiteit



Architectuur

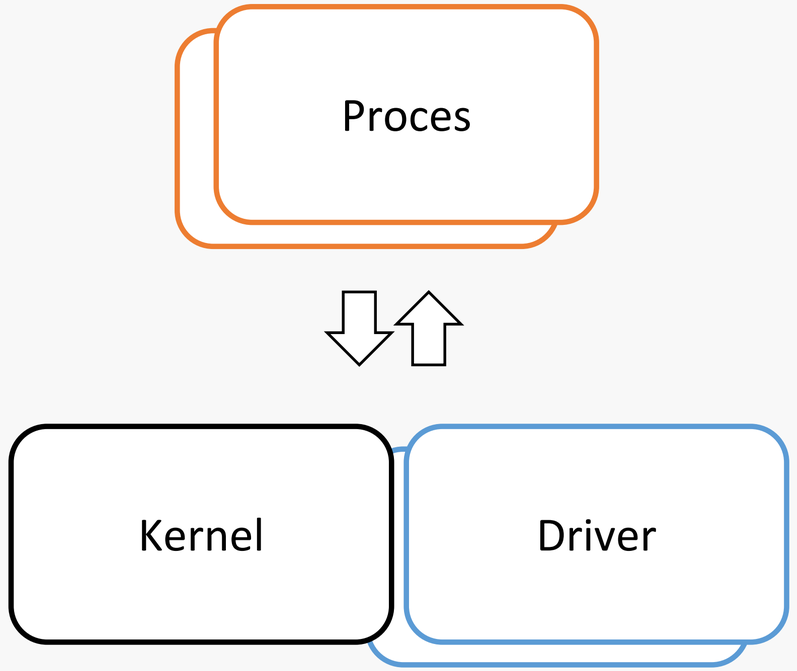
- Monolitisch
- Microkernel
- Unikernel

Monolitisch

- Processen in eigen domeinen
- Drivers in kerneldomein

Voorbeelden:

- Windows
- Linux
- BSD

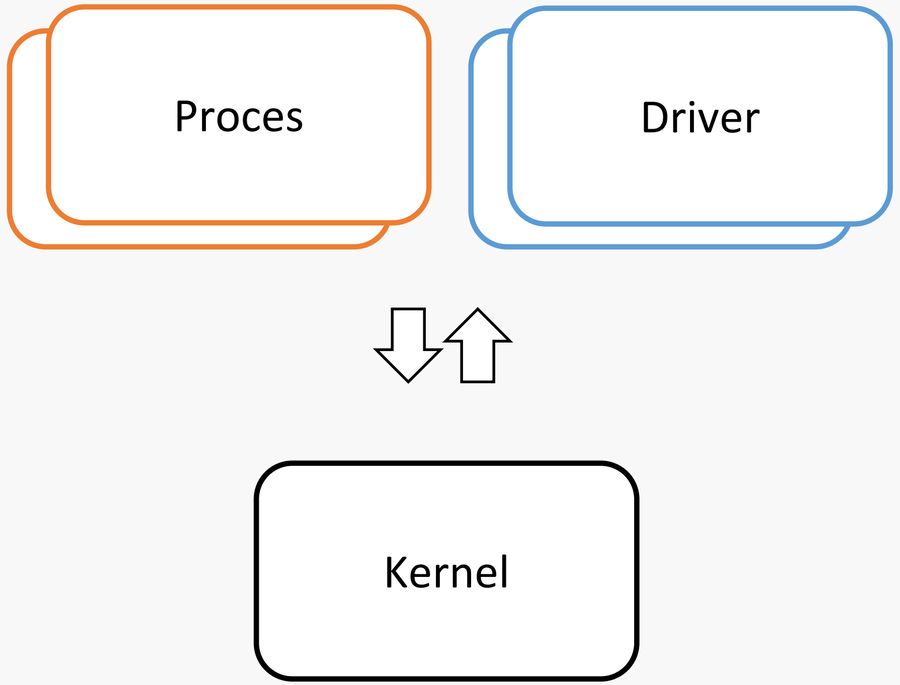


Microkernel

- Minimale kernel
- Drivers zijn processen

Voorbeelden:

- Mach
- MINIX
- L4

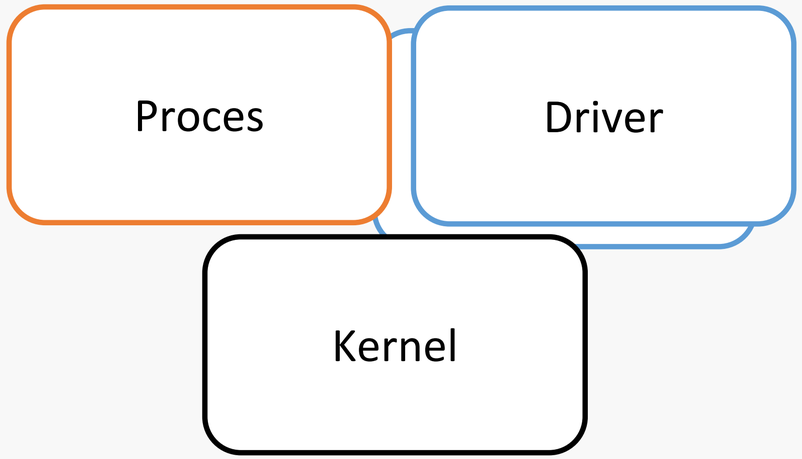


Unikernel

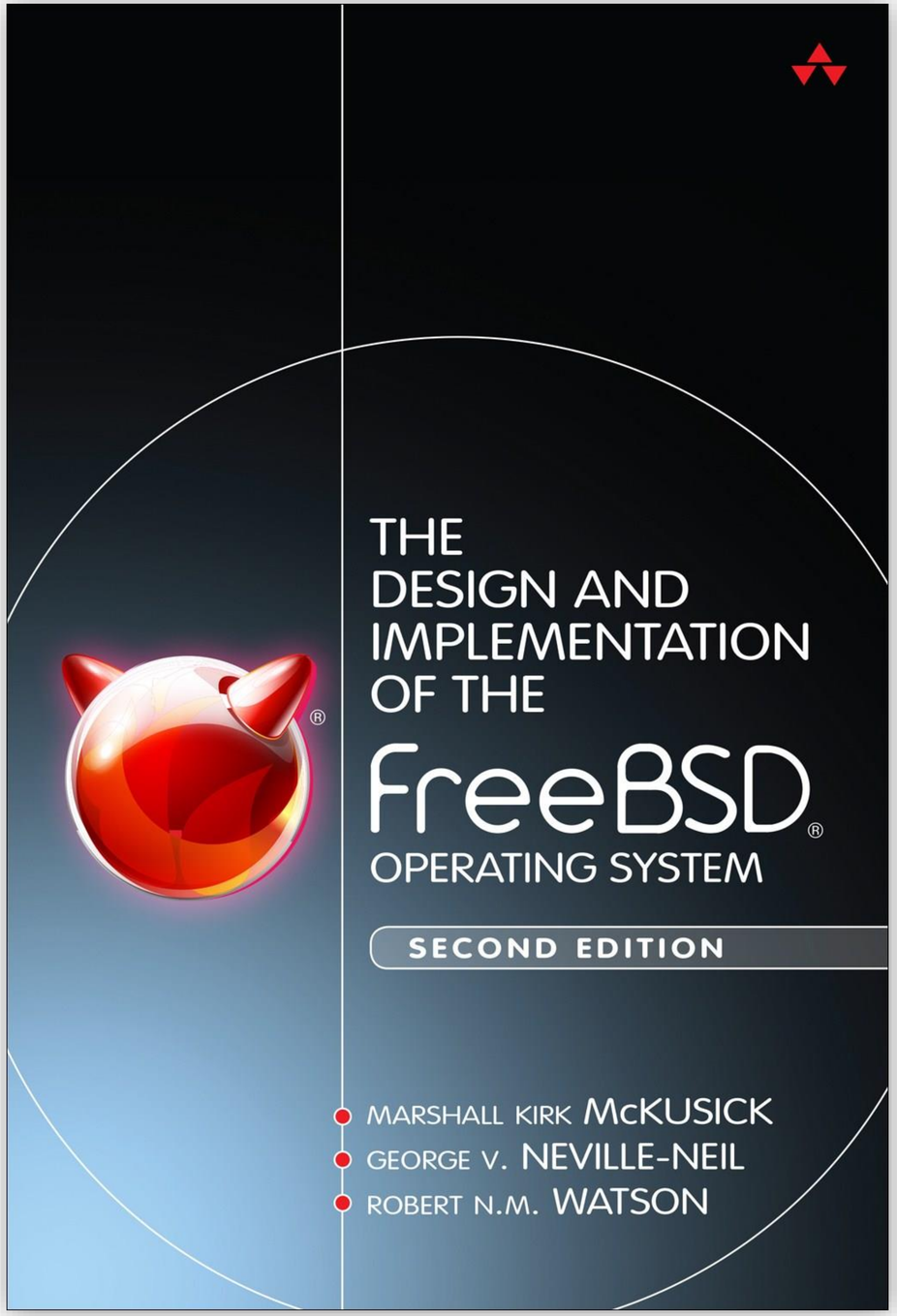
- Eén proces
- Eén domein

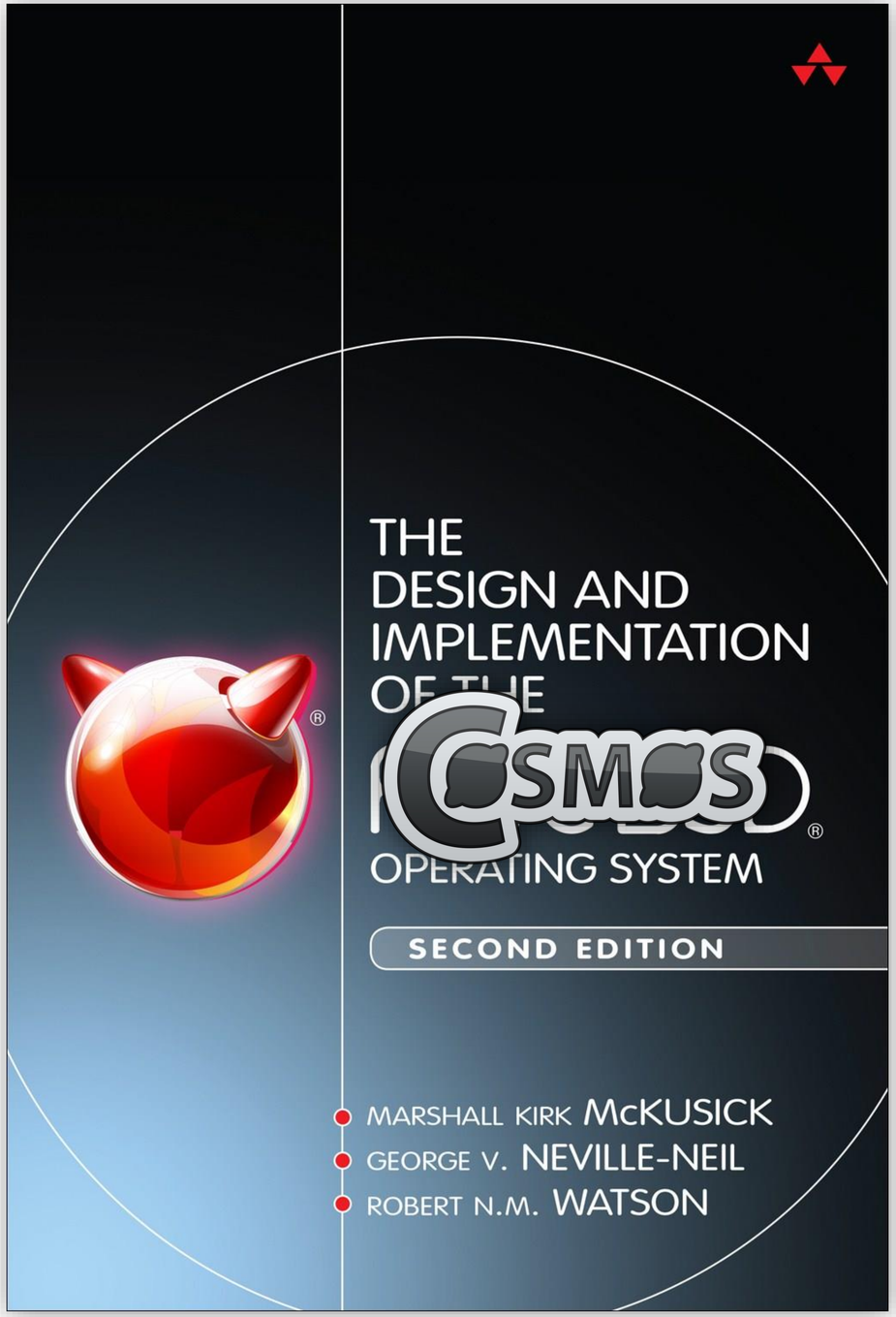
Voorbeelden:

- Cosmos
- IncludeOS
- MirageOS



Demo


THE
DESIGN AND
IMPLEMENTATION
OF THE
FreeBSD®
OPERATING SYSTEM
SECOND EDITION
MARSHALL KIRK McKUSICK
GEORGE V. NEVILLE-NEIL
ROBERT N.M. WATSON

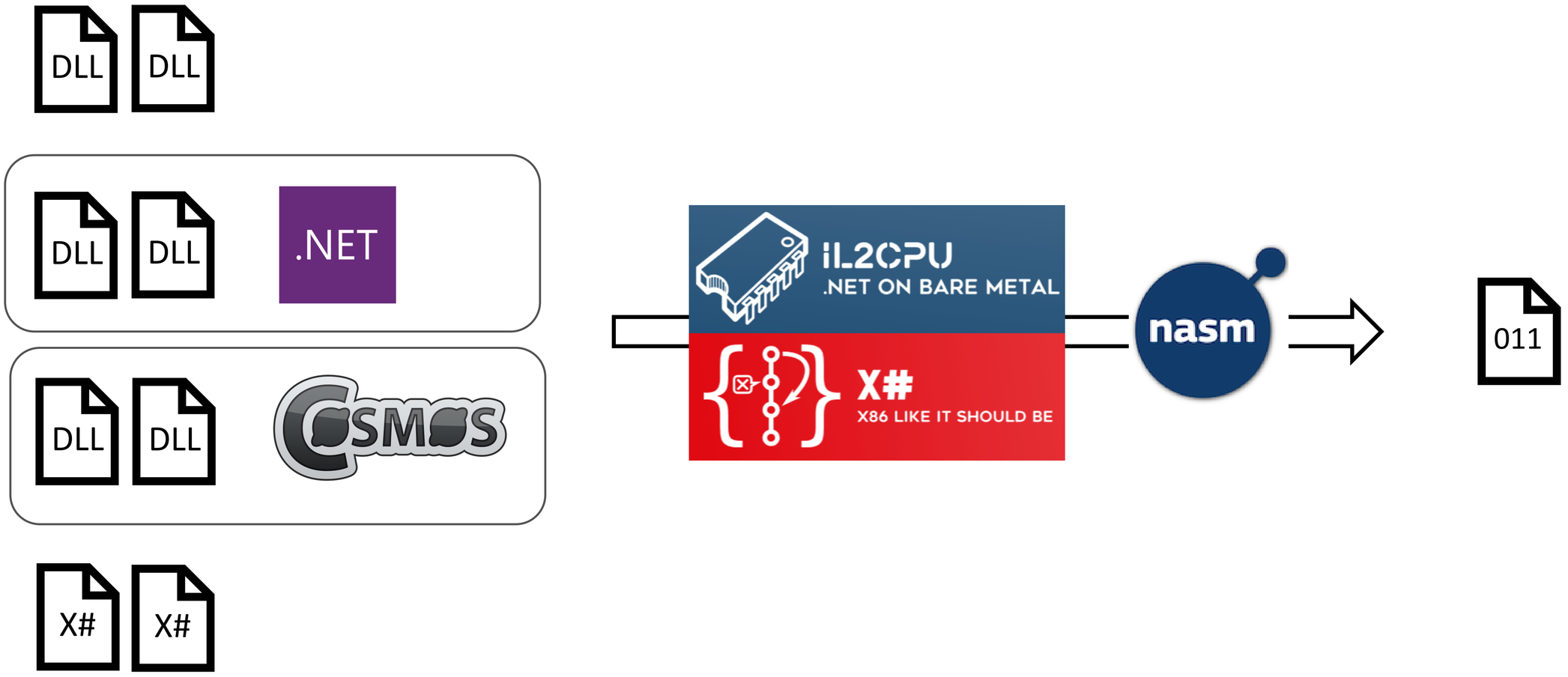

THE
DESIGN AND
IMPLEMENTATION
OF THE
FreeBSD
OPERATING SYSTEM
SECOND EDITION
MARSHALL KIRK McKUSICK
GEORGE V. NEVILLE-NEIL
ROBERT N.M. WATSON

Ontwerpkeuzes

- C#/etc gecompileerd naar machinecode
- Unikernel
- Lineair geheugen

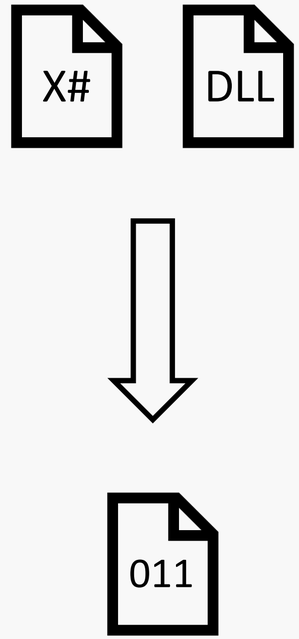
Maar die nadelen?!

betabit


DLL
DLL
DLL
DLL
.NET
DLL
DLL
COSMOS
X#
X#
IL2CPU
.NET ON BARE METAL
X#
X86 LIKE IT SHOULD BE
nasm
011

IL2CPU

- Compileert MSIL naar assembly
- (Voor nu) alleen x86
- *Plugs* om implementaties te vervangen
- Gebuikt X# API



X# API

- Genereer assembly vanuit .NET
- (Voor nu) alleen x86

```
XS.Comment("Arraytype: " +
    aOpType.StackPopTypes.Last().FullName);
XS.Comment("Size: " + aElementSize);

// calculate element offset into array
// memory (including header)
XS.Pop(EAX);
XS.Set(EDX, aElementSize);
XS.Multiply(EDX);
XS.Add(EAX, (uint)(ObjectUtils.FieldDataOffset + 4));

// pop the array now
XS.Add(ESP, 4);
XS.Pop(EDX);

XS.Add(EDX, EAX);
XS.Push(EDX);
```

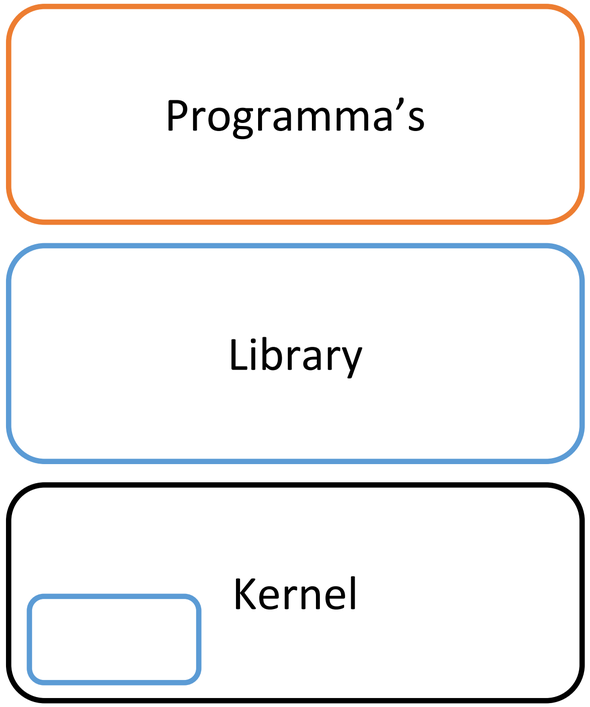
IL2CPU/source/Cosmos.IL2CPU/IL/Ldelema.cs

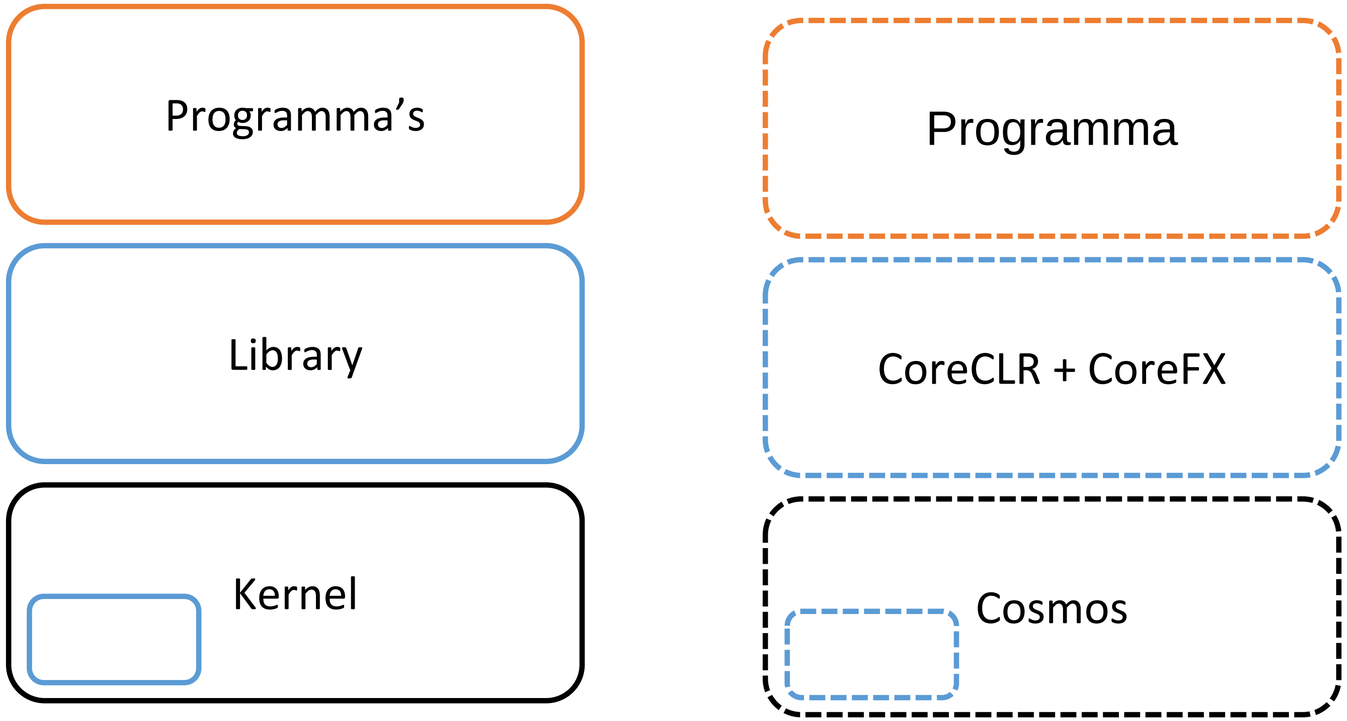
X# taal

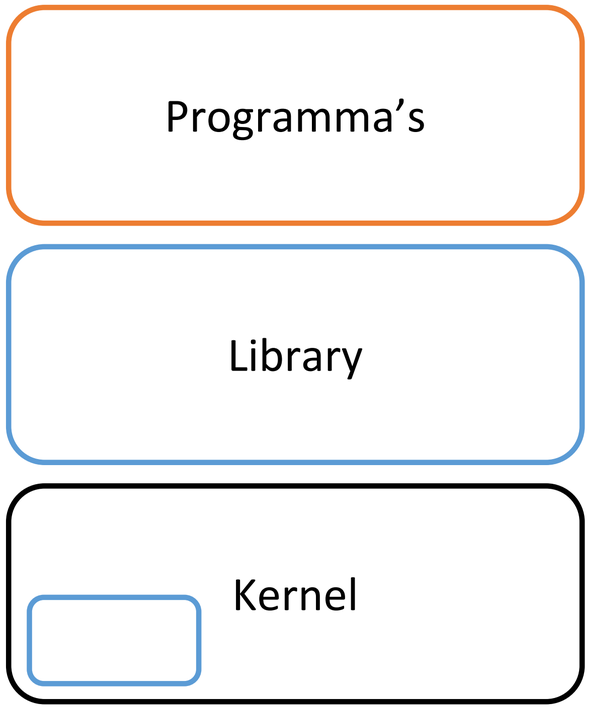
- High level assembler
- Via NASM
- Op basis van X# API
- Niet gebruikt in Cosmos

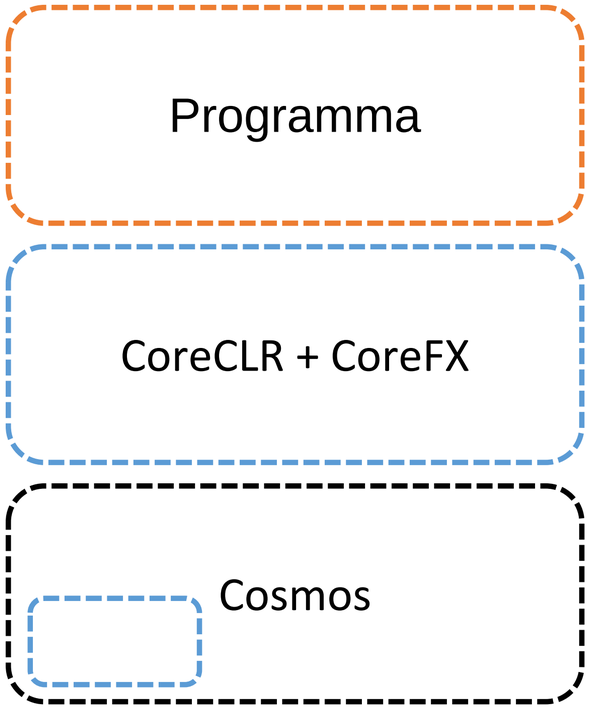
```
function strlen {
    // get pointer to string passed as first argument
    ESI = ESP[4]
    // clear ECX
    ECX ^ ECX
    Loop:
        AL = ESI[ECX]
        if AL = 0 return
        ECX++
        goto Loop
}
```

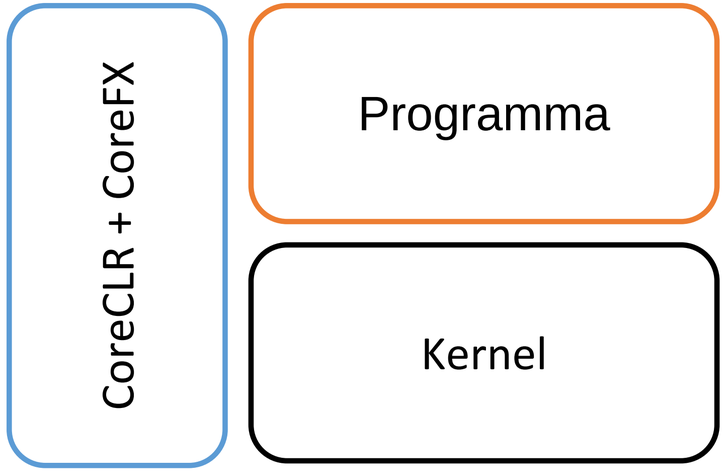
https://www.x-sharp.net/docs/


Programma’s
Library
Kernel


Programma’s
Library
Kernel
Programma
CoreCLR + CoreFX
Cosmos

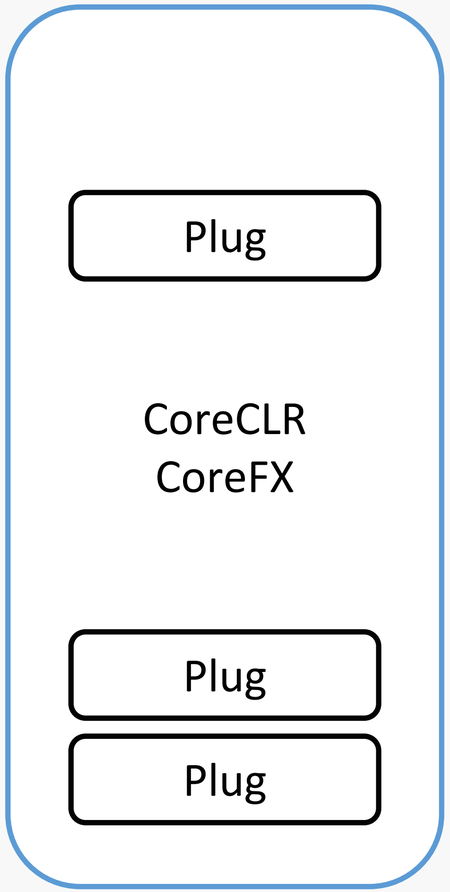

Programma’s
Library
Kernel


Programma
CoreCLR + CoreFX
Cosmos

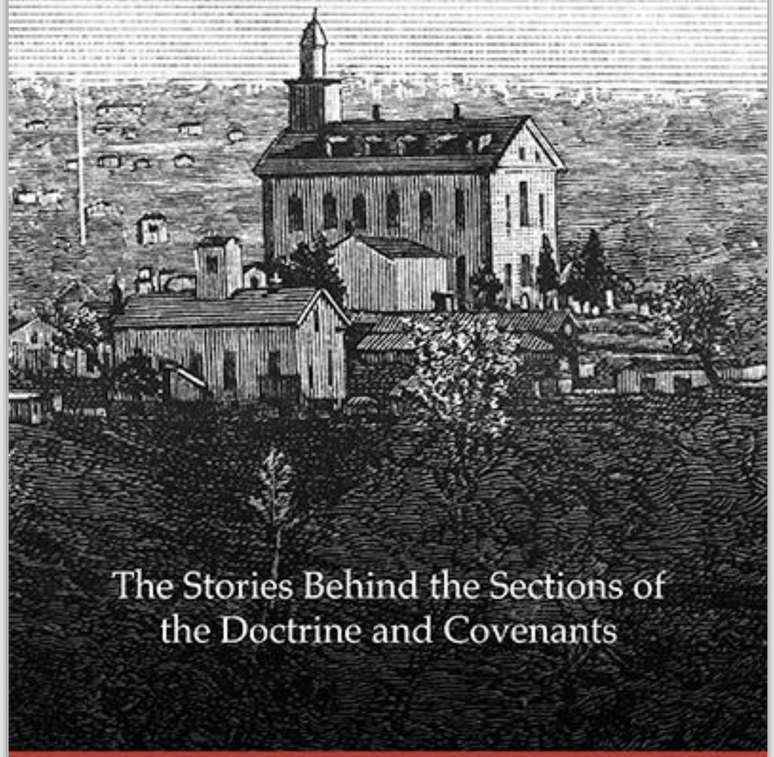

CoreCLR + CoreFX
Programma
Kernel

Plugs

- Vervangen delen bestaande assemblies
- .NET of X#
- Bijvoorbeeld: Console, File, …
- Zo kan reguliere .NET Core worden gebruikt

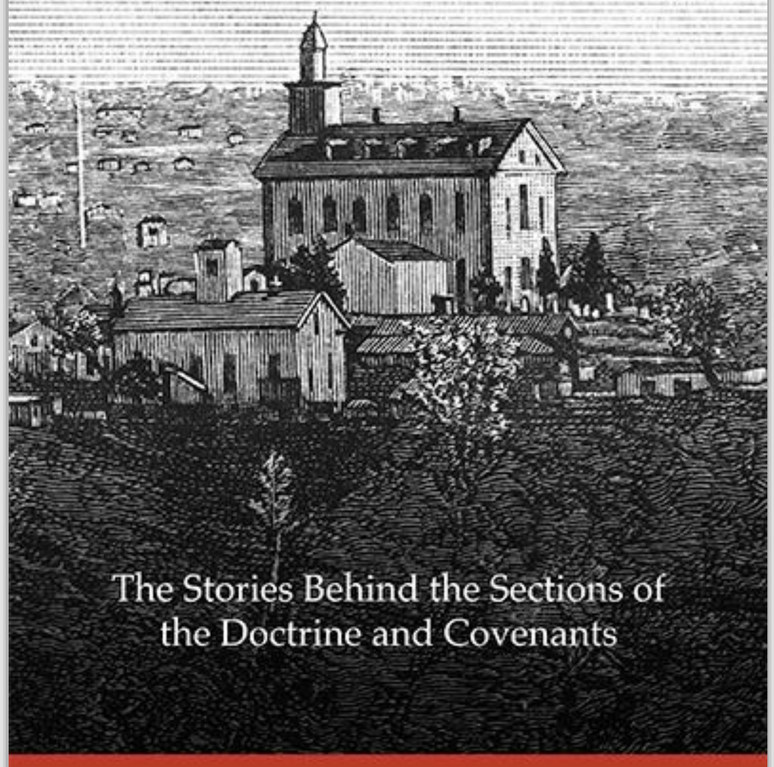


Revelations in Context



The Stories Behind the Sections of the Doctrine and Covenants

Revelations in Context



The Stories Behind the Sections of the Doctrine and Covenants

“Waarom”

- Alternatief voor andere unikernels
- **Geheugenveiligheid** door de talen
- **Privilegescheiding** door het typesystem
- **Scheduling**-trucs via de compiler

.NET-besturingssystemen

- Singularity OS
- MOSA
- Meadow

Singularity

- Onderzoeksproject Microsoft (2003-2010)
- Microkernel
- Lineair geheugenmodel met garanties door taal (zoals Cosmos)

https://www.microsoft.com/en-us/research/project/singularity/

MOSA

- *Managed Operation System Alliance*
- Standaardisatie interfaces .NET operating systems
- Cross platform

https://github.com/mosa/MOSA-Project/wiki

Meadow

- IoT platform
- Vergelijkbare benadering als Cosmos
- Realtime OS
- Van (deel) Mono/Xamarin team

https://www.kickstarter.com/projects/meadow/meadow-full-stack-net-standard-iot-platform

Unikernels

- IncludeOS (http://www.includeos.org)
- Mirage OS (https://mirage.io)

Bijzondere vermeldingen

- OpenBSD
- seL4
- Redox

OpenBSD

- Conventioneel (evolutionair)
- Focus op veiligheid in de diepte
- pledge(), retpoline, …

https://www.openbsd.org

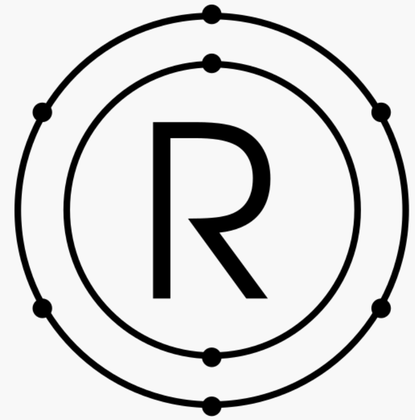
seL4

- L4 microkernel zonder bugs*
- Geschreven in Haskell en C
- *Compleet formeel geverifieerd!*

http://sel4.systems

* Terms and conditions apply

Redox



- Unix-achtig
- Geschreven in Rust

https://www.redox-os.org

Andere richtingen

- Containers
- Virtuele machines (Erlang, Java, WebAssembly, ...)
- Hybride oplossingen

betabit {...}

samen bijzonder maken;

Bedankt voor je aandacht