Quo vadis Linux?!?

The rise of new (cloud) operating systems

Dr. Udo Seidel

Agenda

- Introduction
- A few actors
- Time to think
- Summary

Me :-)

- Teacher of mathematics and physics
- PhD in experimental physics
- Started with Linux/Opensource in 1996
- With Amadeus since 2006
- Before:
  - Linux/UNIX trainer
  - Solution Engineer in HPC and CAx environment
- Now: Architecture & Technical Governance

Introduction

Linux questioned/challenged

- Role of the O/S
  - Container
  - Compute as a Service
- Platform as a Service
- Cloud Native Applications

- … not only Linux

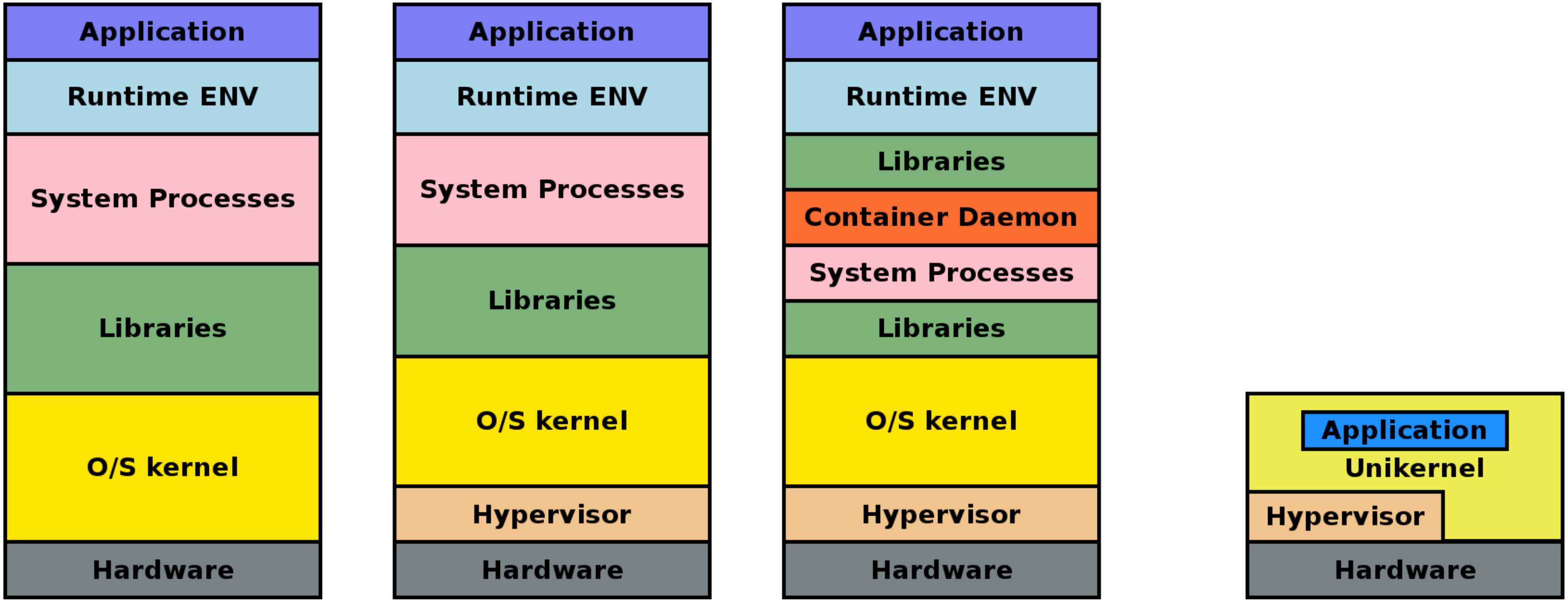
The cloud

- Era/hype/age
- Trigger of paradigm shift
- Demand of new 'compute'

About unikernels

- Not new
- Just enough Operating system
- See also
  - Exokernel
  - Anykernel

Unikernel idea in a picture



Opportunities & Consequences

- Green field
- Coding framework
- Re-design

Why did it not fly so far ...

- Hardware drivers/support
- Brown field
- 'Silver bullet' business case

Some actors

Selection

- Technical baseline
  - General purpose O/S
  - Linux
  - Opensource
- Non-technical baseline
  - Known suspects
  - New players

In and out

- In:
  - OSv
  - MirageOS
  - Rumprun
- Out:
  - ClickOS
  - Clive
  - Drawbridge
  - HaLVM
  - ...

Approach

- History & facts
- Kernel (space)
- User-space
- Some discussion

OSv

History & Facts

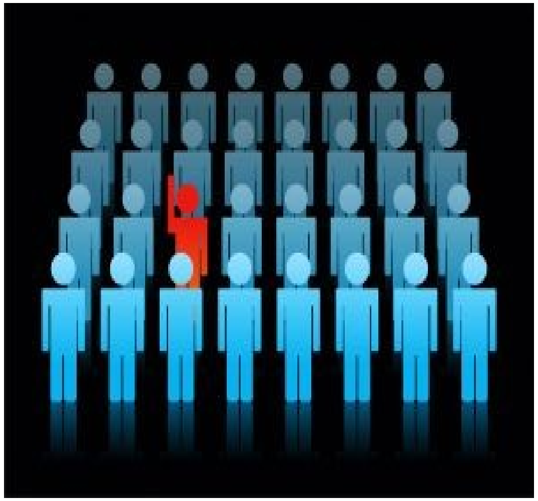
- Initial release: September 2013
- License: BSD-3
- Kernel: Monolithic
- Language: C++
- Platforms:
  - x86_64: virtual only
  - ARM64: (Still) to come

Kernel

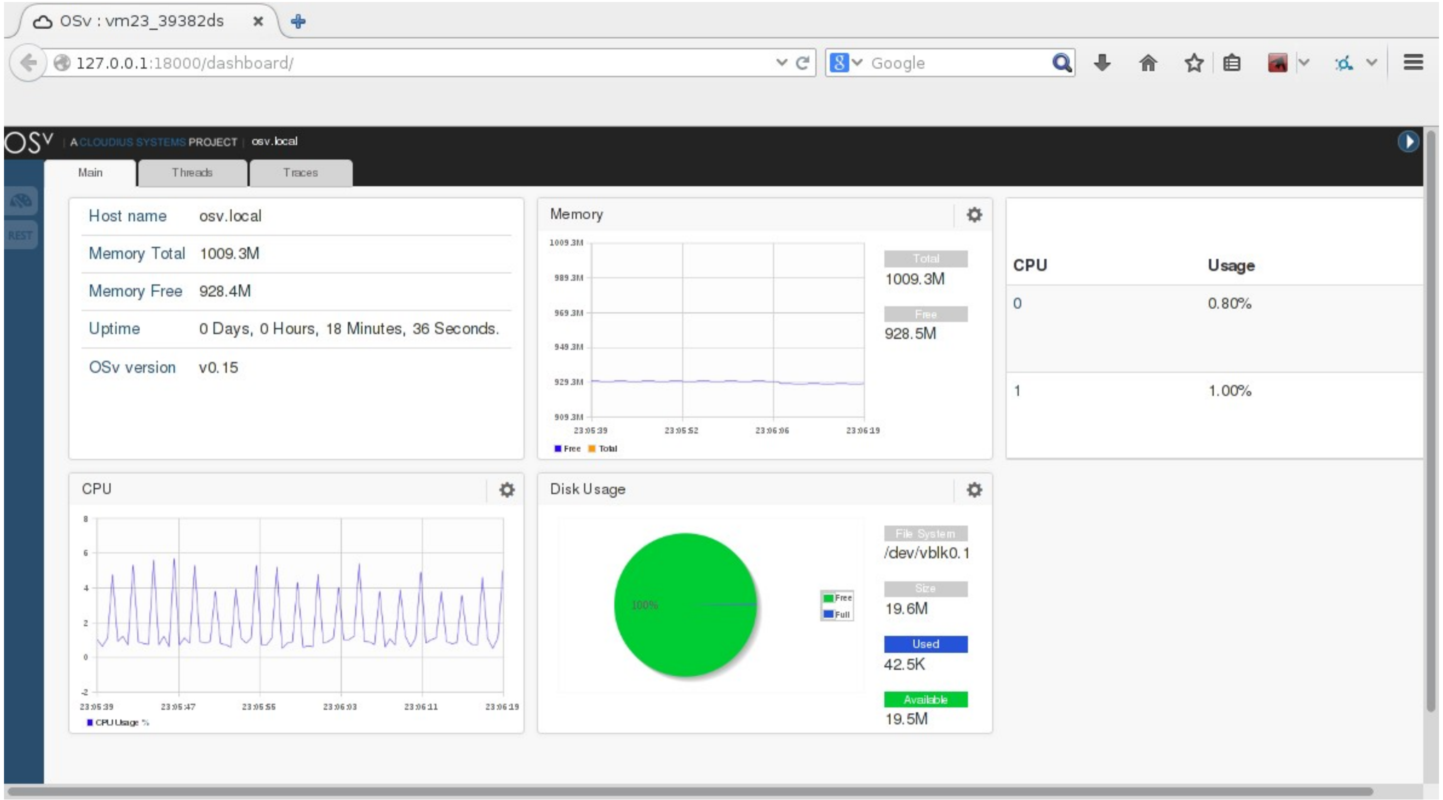
- Newly developed
- No physical hardware support
- Smallest possible layer
  - Run on hypervisor
  - Host application
  - E. g. no spinlocks
- Linux-ABIs in place

User space

- Not existing!
- No user concept
- Single process only
  - Threads
  - Relocatable object code application
  - Missing: *fork(), vfork(), clone()*



REST API/Dashboard



Host name	osv.local
Memory Total	1009.3M
Memory Free	928.4M
Uptime	0 Days, 0 Hours, 18 Minutes, 36 Seconds.
OSv version	v0.15

Memory

Total
1009.3M

Free
928.5M

CPU	Usage
0	0.80%
1	1.00%

Disk Usage

File System
/dev/vblk0.1

Size
19.6M

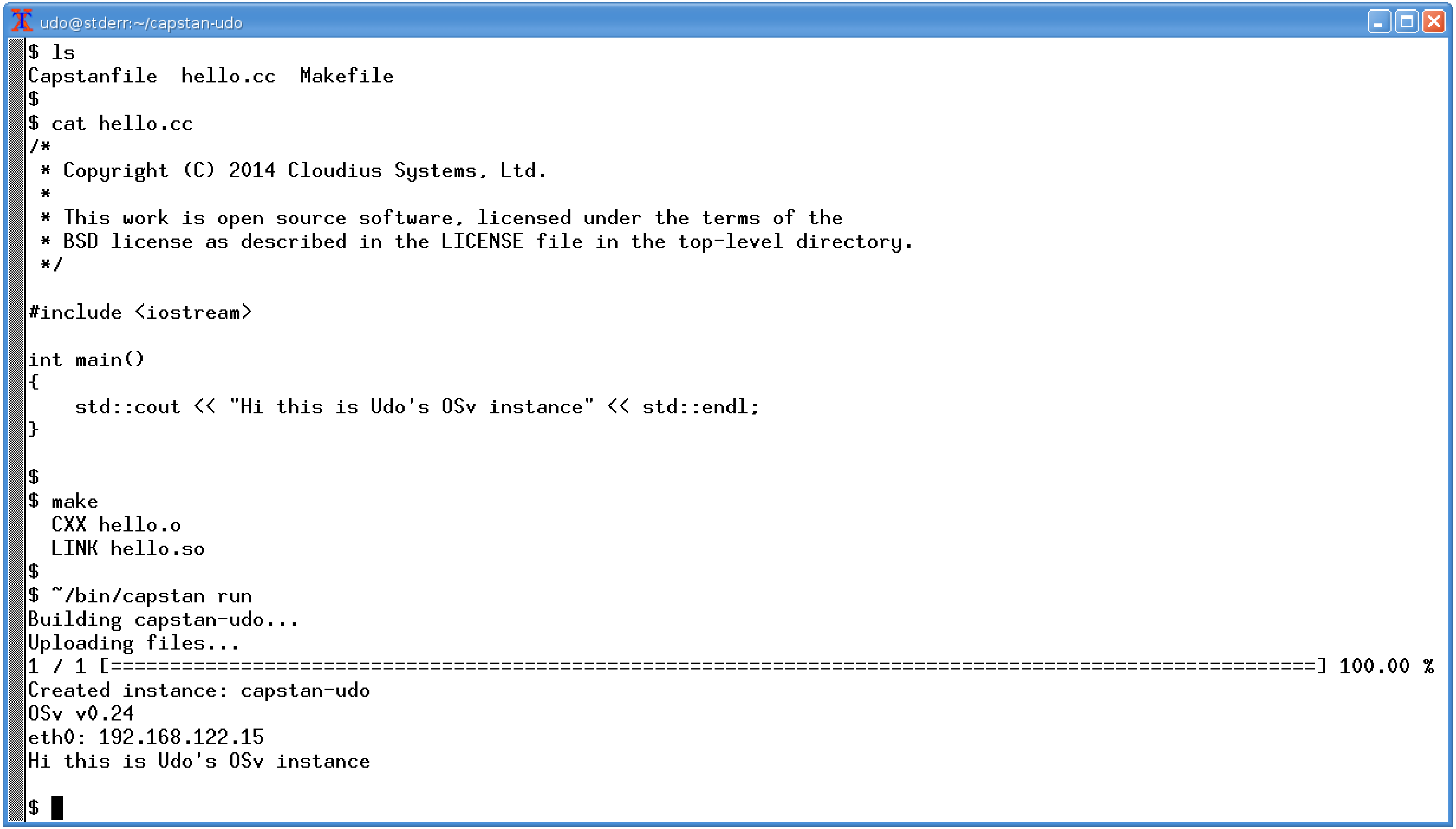
Used
42.5K

Available
19.5M

Application Build

- Source code
- Build environment
- OSv integration
  - Locally
    - *capstan* → *Capstanfile*
    - Download, build, run
  - Cloud-init

'Hello world'



```
$ ls
Capstanfile  hello.cc  Makefile
$
$ cat hello.cc
/*
 * Copyright (C) 2014 Cloudius Systems, Ltd.
 *
 * This work is open source software, licensed under the terms of the
 * BSD license as described in the LICENSE file in the top-level directory.
 */

#include <iostream>

int main()
{
    std::cout << "Hi this is Udo's OSv instance" << std::endl;
}

$
$ make
  CXX hello.o
  LINK hello.so
$
$ ~/bin/capstan run
Building capstan-udo...
Uploading files...
1 / 1 [======================================================================================================] 100.00 %
Created instance: capstan-udo
OSv v0.24
eth0: 192.168.122.15
Hi this is Udo's OSv instance

$ █
```

Images Market Place

- Java (openJDK)
- Tomcat
- Memcached
- Reddis
- Cassandra
- Kafka
- Netperf
- … DIY

Who is using OSv?

- Cloudius Systems
  - CloudRouter
  - OpenDaylight
- Research projects
- ???

MirageOS

History & Facts

- Initial release: December 2013
- License: ISC
- Kernel: N/A (libOS)
- Language: OCamel/OPAM
- Platforms:
  - x86_64: virtual (Xen)
  - ARMv7+: virtual (Xen)

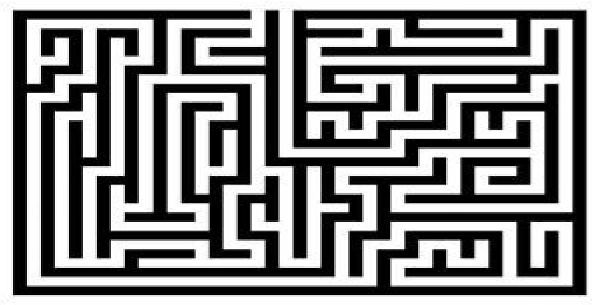
Kernel

- N/A!
  - Library Operating System
  - Different targets
  - DIY for almost everything
- Written in OCamel
- Functions via libraries

User space

- Not existing
- Actually …

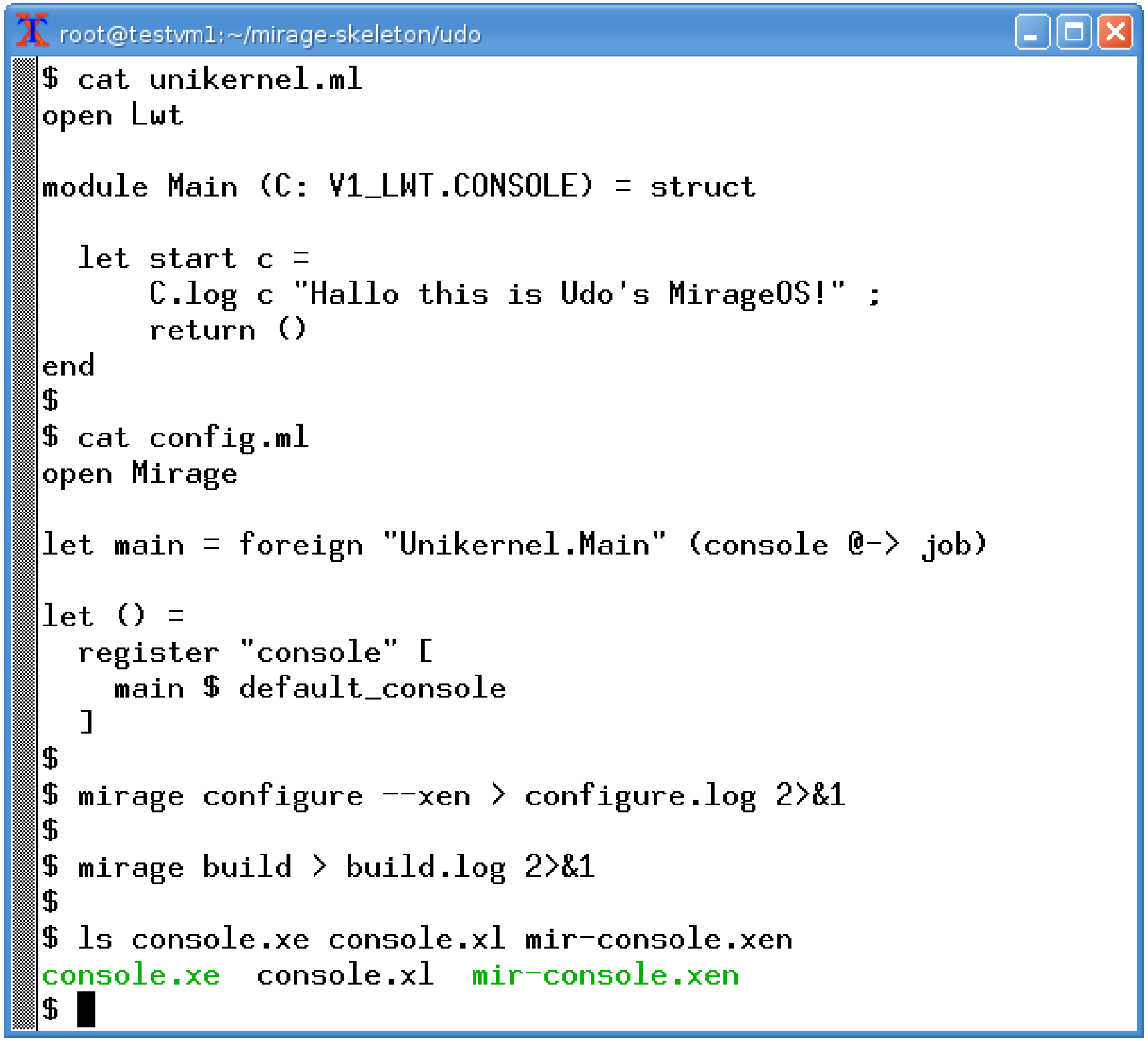
  … just the O/S Kernel



Application build

- Lot of (pre-)thinking
  - Application
  - Kernel
- OCamel and OPAM
  - *unikernel.ml*
- Mirage Integration
  - *config.ml*
  - *mirage ….*

'Hello world'



```
$ cat unikernel.ml
open Lwt

module Main (C: V1_LWT.CONSOLE) = struct

  let start c =
      C.log c "Hallo this is Udo's MirageOS!" ;
      return ()
end
$
$ cat config.ml
open Mirage

let main = foreign "Unikernel.Main" (console @-> job)

let () =
  register "console" [
    main $ default_console
  ]
$
$ mirage configure --xen > configure.log 2>&1
$
$ mirage build > build.log 2>&1
$
$ ls console.xe console.xl mir-console.xen
console.xe  console.xl  mir-console.xen
$
```

'Kernel Libraries'

- Core
- Storage, e.g.
  - Block device
  - File system
- Network, e.g.
  - TCP/IP
  - HTTP
- Formats, e.g
  - JSON

Known/tested Use Case

- Webserver
- DNS
- Openflow Controller

Who is using MirageOS?

- Research projects
- ???

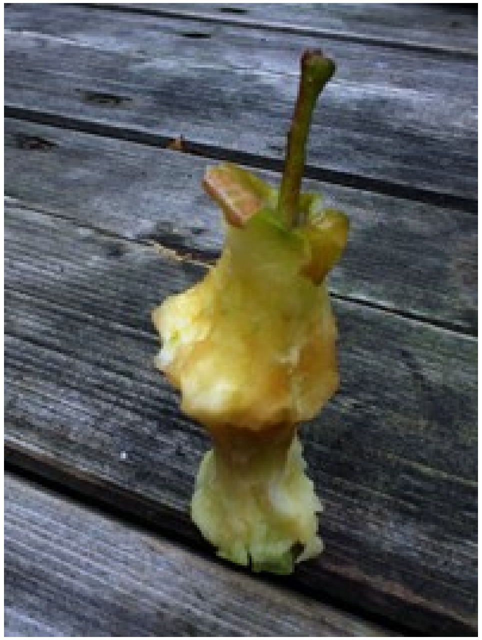
Rumprun

History & Facts

- Initial release: 2012/2013
- License: BSD-2
- Kernel: Monolithic .. but ...
- Language: C
- Platforms:
  - x86_64: physical and virtual
  - ARM: dito

Kernel

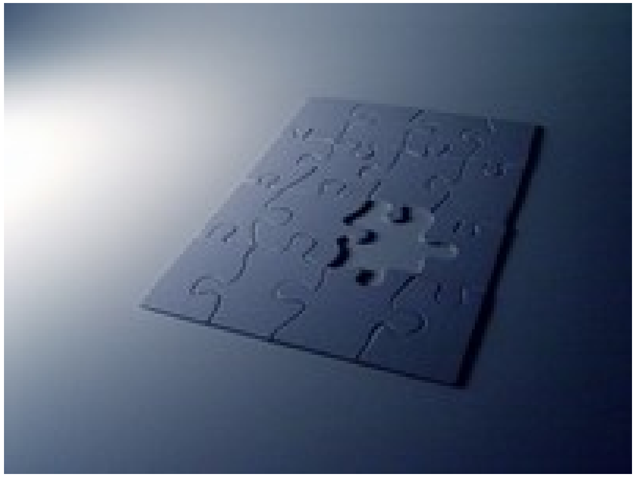
- Derived from Rump kernels (NetBSD)
- Comparable to OSv
- Hardware drivers
  - Needed
  - kernel vs. user-space



User space

- Not existing
- Actually …

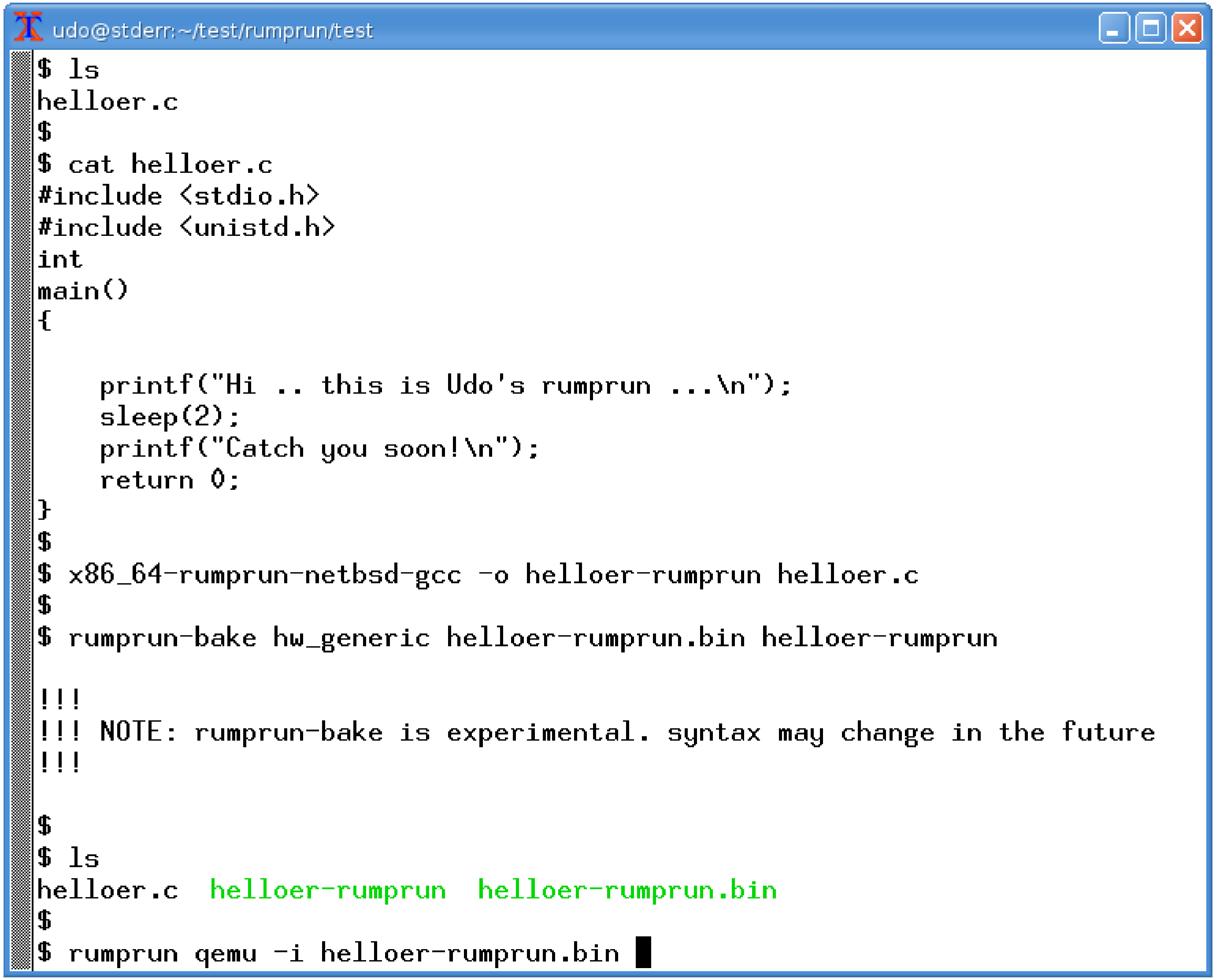
  … just the O/S Kernel
- See OSv and MirageOS



Application build

- Keep using: development framework
- Forget:*fork()/vfork()/clone()*
- Rumprun integration
  - Ready to go
  - Quite easy
- See also *rumpctrl*

'Hello world'



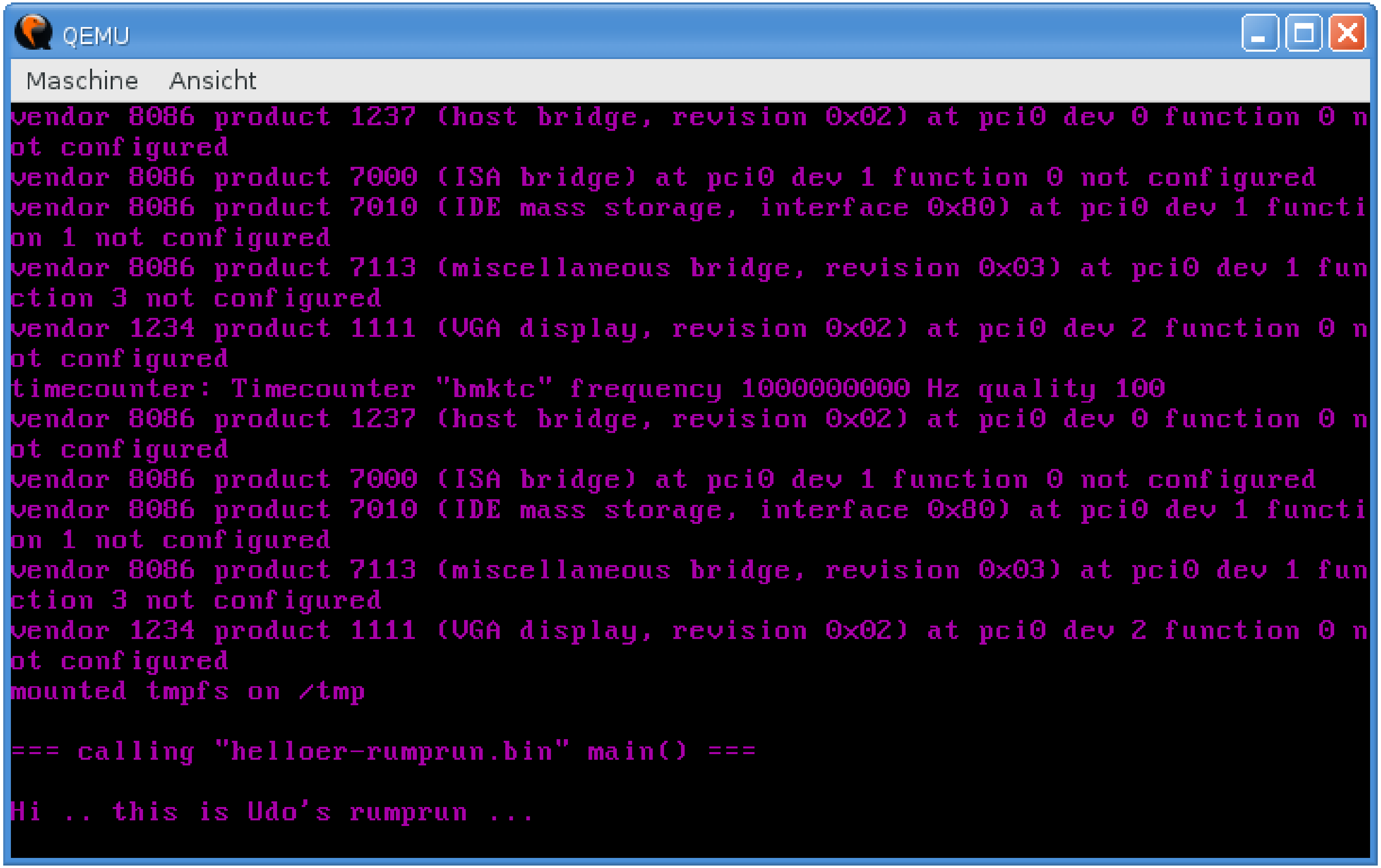
```
$ ls
helloer.c
$
$ cat helloer.c
#include <stdio.h>
#include <unistd.h>
int
main()
{

    printf("Hi .. this is Udo's rumprun ...\n");
    sleep(2);
    printf("Catch you soon!\n");
    return 0;
}
$
$ x86_64-rumprun-netbsd-gcc -o helloer-rumprun helloer.c
$
$ rumprun-bake hw_generic helloer-rumprun.bin helloer-rumprun

!!!
!!! NOTE: rumprun-bake is experimental. syntax may change in the future
!!!

$
$ ls
helloer.c  helloer-rumprun  helloer-rumprun.bin
$
$ rumprun qemu -i helloer-rumprun.bin
```

'Hello world'



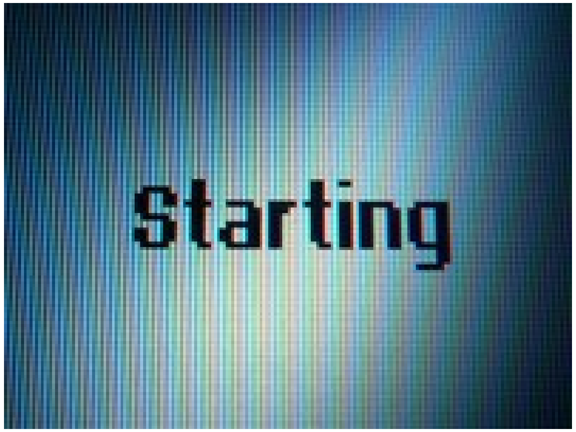
```
vendor 8086 product 1237 (host bridge, revision 0x02) at pci0 dev 0 function 0 n
ot configured
vendor 8086 product 7000 (ISA bridge) at pci0 dev 1 function 0 not configured
vendor 8086 product 7010 (IDE mass storage, interface 0x80) at pci0 dev 1 functi
on 1 not configured
vendor 8086 product 7113 (miscellaneous bridge, revision 0x03) at pci0 dev 1 fun
ction 3 not configured
vendor 1234 product 1111 (VGA display, revision 0x02) at pci0 dev 2 function 0 n
ot configured
timecounter: Timecounter "bmktc" frequency 1000000000 Hz quality 100
vendor 8086 product 1237 (host bridge, revision 0x02) at pci0 dev 0 function 0 n
ot configured
vendor 8086 product 7000 (ISA bridge) at pci0 dev 1 function 0 not configured
vendor 8086 product 7010 (IDE mass storage, interface 0x80) at pci0 dev 1 functi
on 1 not configured
vendor 8086 product 7113 (miscellaneous bridge, revision 0x03) at pci0 dev 1 fun
ction 3 not configured
vendor 1234 product 1111 (VGA display, revision 0x02) at pci0 dev 2 function 0 n
ot configured
mounted tmpfs on /tmp

=== calling "helloer-rumprun.bin" main() ===

Hi .. this is Udo's rumprun ...
```

Known/tested Use Case

- See OSv
- ???



Who is using Rumprun?

- Research projects
- EMC (UniK)

Time to think

First Thoughts on Osv/Rumprun

- More adaptation needed
- ~~'Traditional' approach~~
- Potential for code re-use
- More options/freedom/legacy on Rumprun

Embrace OSv/Rumprun

- Single application!
- ELF shared object (OSv)
- RUN in Kernel space
- Short/medium runtime
- Roll-out/-back on system instance level
- Data management

First Thoughts on MirageOS

- Bigger changes
- ~~'Traditional' approach~~
- ~~Code re-use~~
- Almost nothing known left

Embrace MirageOS

- Review Kernel needs
- Review coding language
- Test with UNIX target
- Application = Kernel
- Data Management

Summary

First 'Last' Thoughts

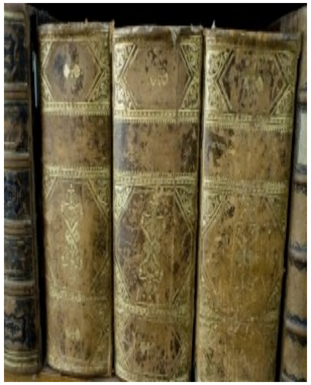
- Ufff!! ...Crystal ball?!?
- Potentially lot of work
  - Technically
  - Mindset/paradigm change
- Customer and business

Take Aways

- Focus shift
- Further paradigm shifts to come
- Today: multiple options
- Open your mind :-)
- Take it easy :-D

References

- http://osv.io
- http://openmirage.org
- http://rumpkernels.org



Thank you!

Quo vadis Linux?!?

The rise of new (cloud) operating systems

Dr. Udo Seidel

TUEBIX 2016