

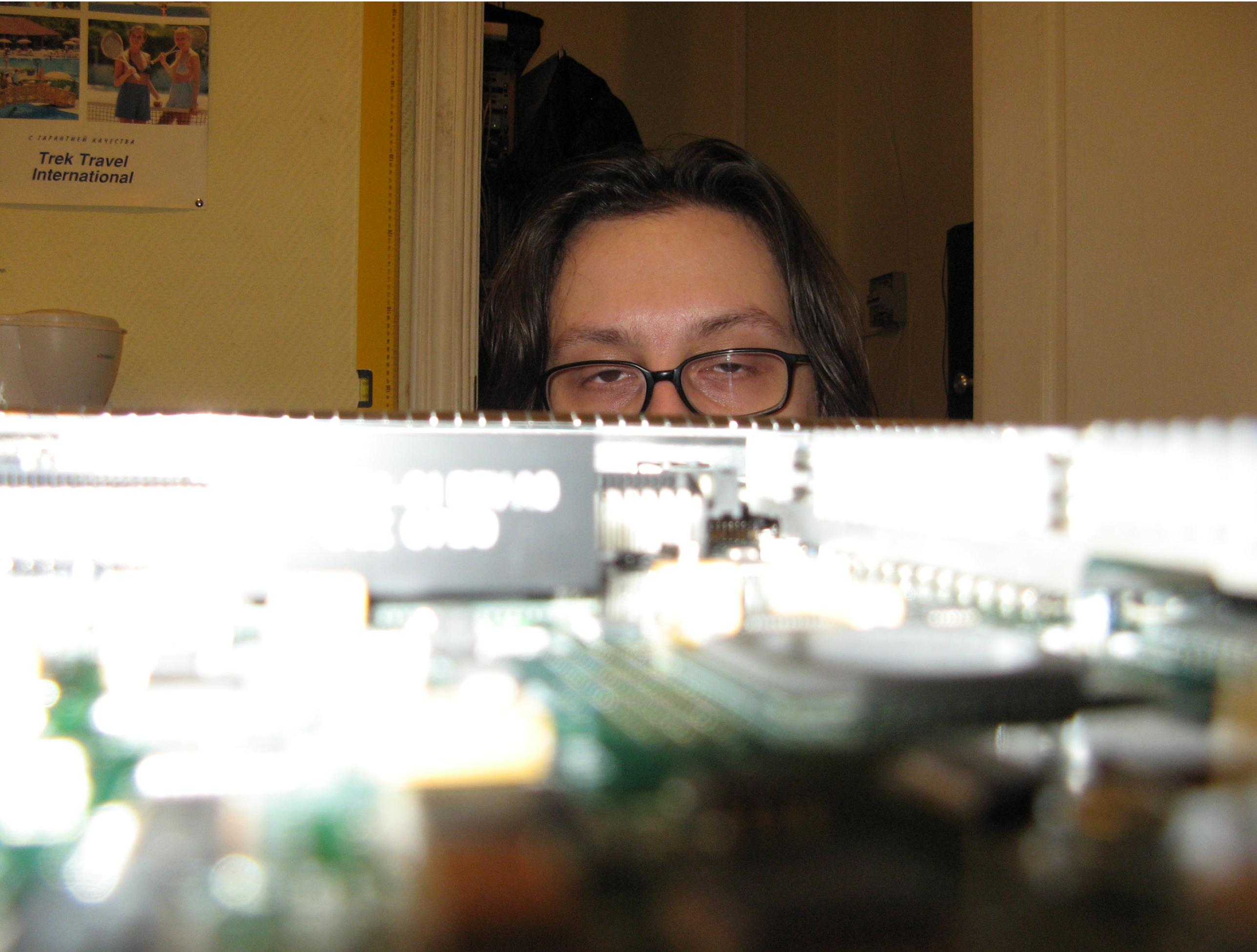
/evrone.

R'N'D СВОИМИ РУКАМИ

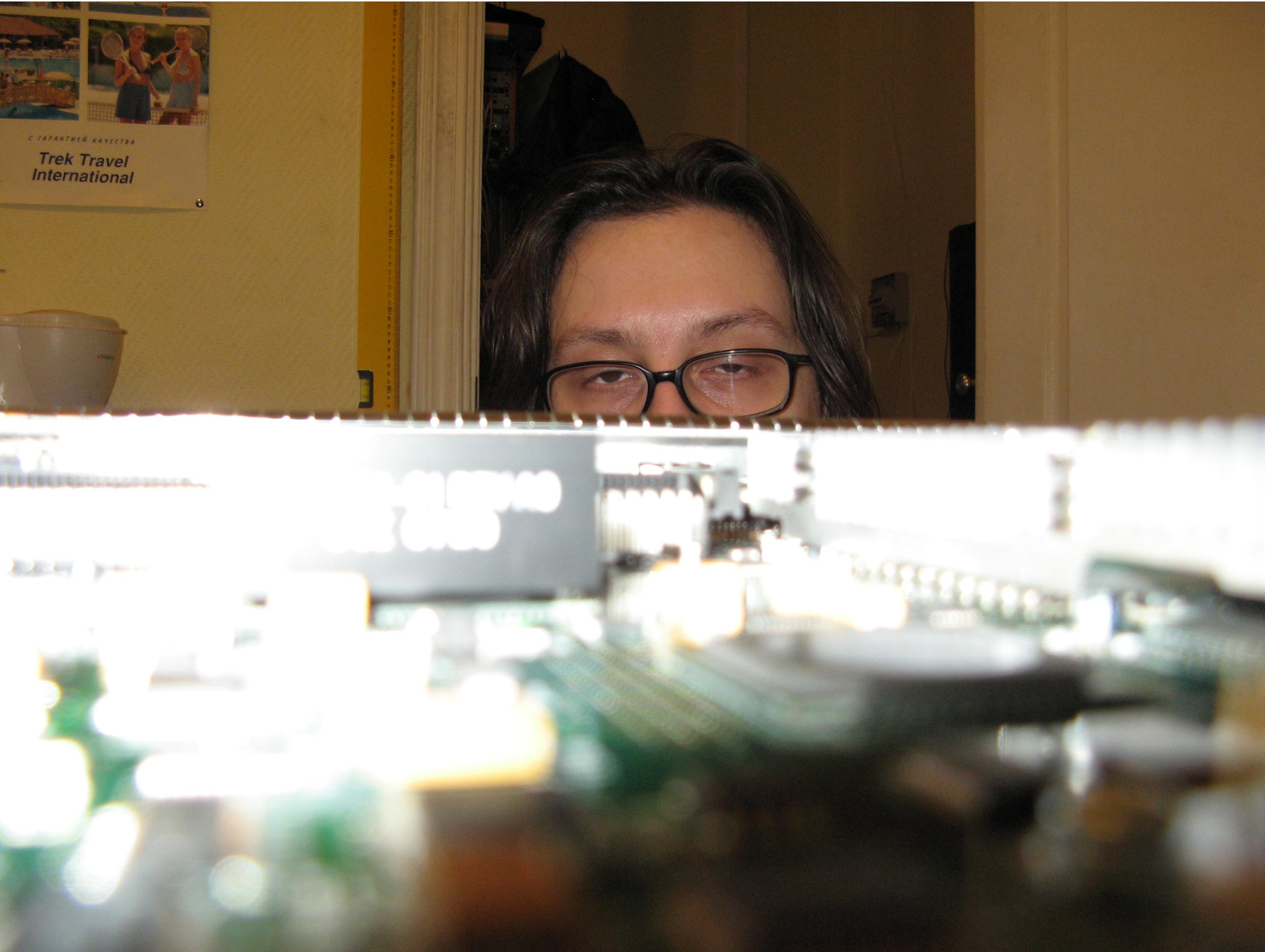
---

**Учимся играть в unikernel**

**CV:**



**CV:**

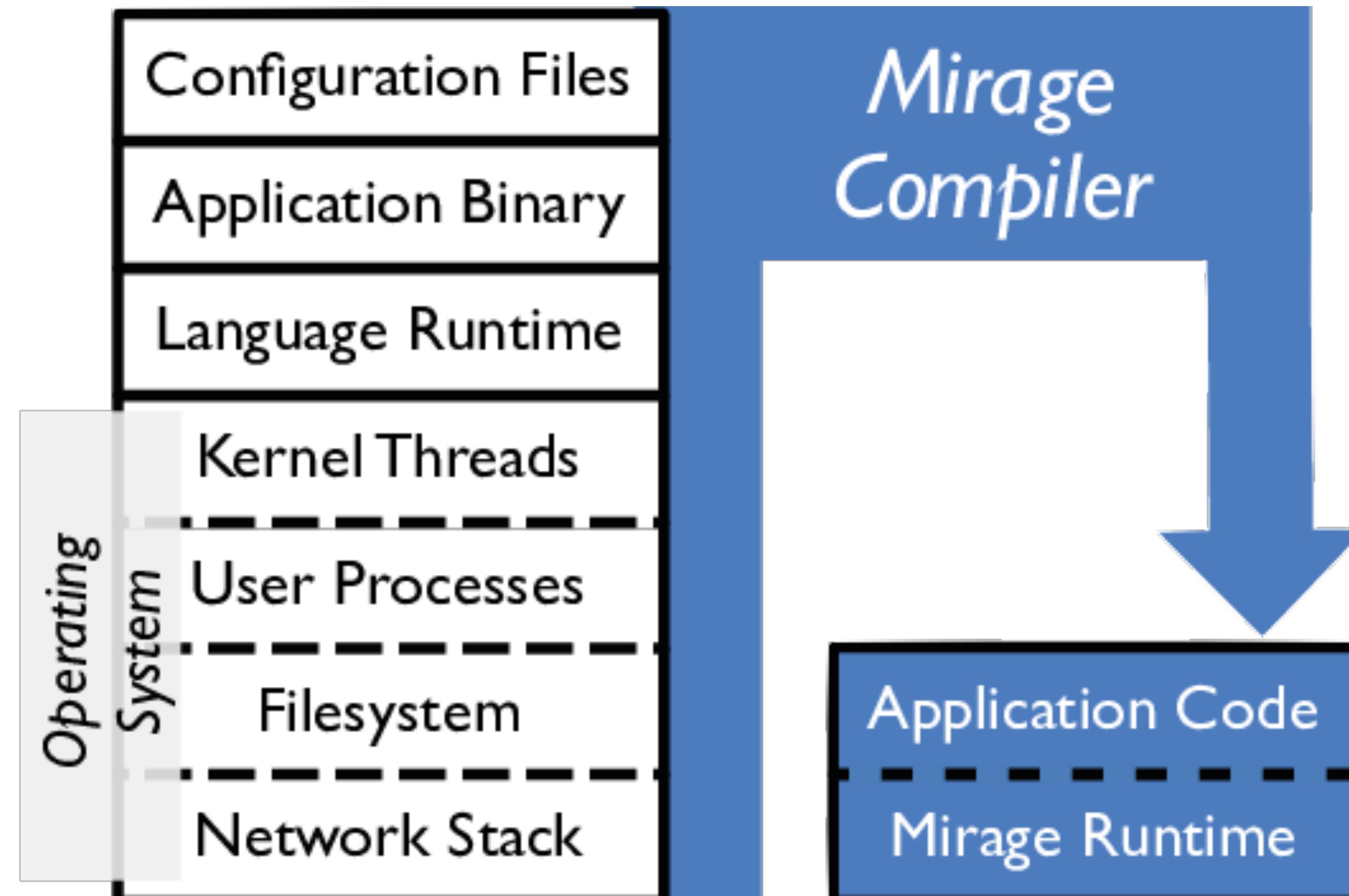


## UNIKERNEL

- ▶ Фреймворк
- ▶ Компактность
- ▶ Безопасность
- ▶ Оптимизация

# UNIKERNEL

- ▶ Фреймворк
- ▶ Компактность
- ▶ Безопасность
- ▶ Оптимизация



## UNIKERNEL

- ▶ Писать программу самому!
- ▶ Знать язык!
- ▶ Возиться и отлаживать!

## UNIKERNEL

- ▶ Писать программу самому!
- ▶ Знать язык!
- ▶ Возиться и отлаживать!

Работать трудно...



# MIRAGE OS

- ▶ OCaml
- ▶ Libraries (Lwt, cohttp, TCP/IP, etc.)
- ▶ XEN
- ▶ UNIX (статический бинарник)
- ▶ Virtio/KVM/VBox (Solo5 — загрузчик)

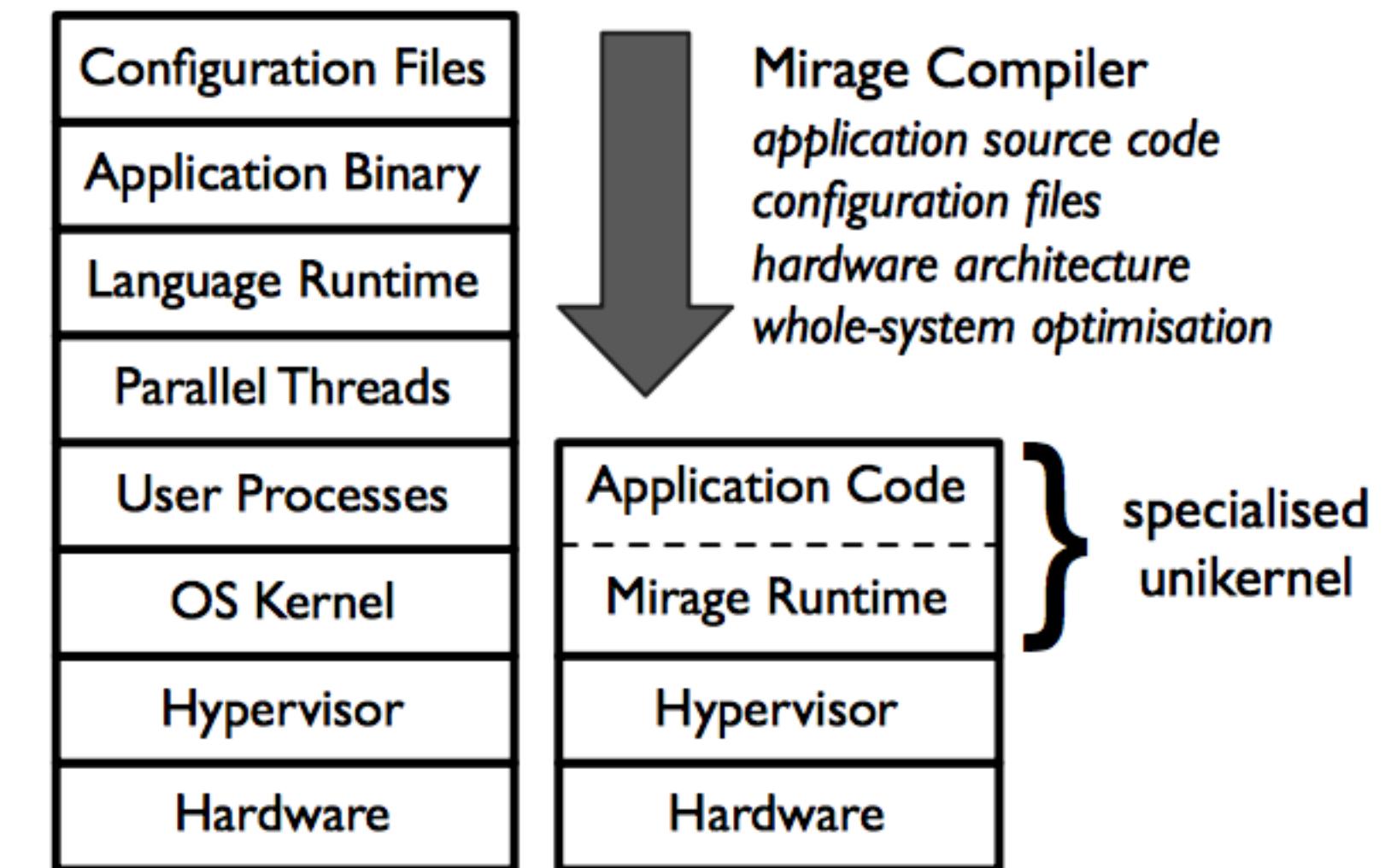


Figure 1: Contrasting software layers in existing VM appliances *vs.* unikernel's standalone kernel compilation approach.

# MIRAGE OS

- ▶ Web-сервер ([mirage.io](http://mirage.io))
- ▶ Blockchain ([tezos.com](http://tezos.com))
- ▶ Datakit ([Docker et al.](#))
- ▶ Cuekeeper (TODO app)
- ▶ ... ([it's OCaml](#))



The MirageOS Blog on building functional operating systems

Apr 15 2017 By Hannes Mehnert, Gemma Gordon, Anil Madhavapeddy

## Cambio, OCaml and Karaoke: MirageOS Hack Retreat, Marrakech 2017

This March, 34 people from around the world gathered in Marrakech for a spring Mirage hackathon. This is fast becoming a [MirageOS tradition](#), and we're a little sad that it's over already! We've collected some trip reports from those who attended the 2017 Hack Retreat, and we'd like to thank our amazing hosts, organisers and everyone who took the time to write up their experiences. Props go especially to Hannes Mehnert who initiated the event and took care of many of the logistics, and to Gemma Gordon for designing and printing [limited edition t-shirts](#) especially for the occasion!

In addition to the reports below, you can find other information online:

- the daily [tweets about the event](#), including sophisticated "paper slides" by [Hannes Mehnert](#) and [Reynir Björnsson](#) which you can [note up](#) on their personal sites.

**Hannes Mehnert**  
At the retreat, 34 people from all around the world (mainly Western Europe) interested in MirageOS gathered for a week in Marrakech.

**Recent Posts**

- MirageOS March 2017 Hackathon roundup
- Easy distributed analytics with Irmin 1.0
- Adding the Qubes target to Mirage
- Size matters: how Mirage got smaller and less magical
- Announcing MirageOS 3.0.0
- Irmin 0.12 with portable filesystem watching released
- MirageOS Summer 2016 Hackathon roundup
- MirageOS Summer 2016 Hackathon announcement, and talk roundup
- MirageOS Spring 2016 Hackathon!
- MirageOS security advisory 00: mirage-net-xen
- Introducing Functoria
- Run Mirage Unikernels on KVM/QEMU with Solo5
- MirageOS hackathon
- Introducing Charrua — a DHCP implementation
- Unikernel.org
- Videos from around the world!
- Getting Started Screencasts
- Organized chaos: managing

[dev] CueKeeper - Mozilla Firefox

Schedule Review Add or search Export Show history Close all

An action in Start using CueKeeper (show)  
Context: Reading (show)  
Contact: (no contact)  
Repeats: (never)  
[http://en.wikipedia.org/wiki/Getting\\_Things\\_Done](http://en.wikipedia.org/wiki/Getting_Things_Done)  
(add log entry) (edit)  
Created 2015-04-18 13:58 (Sat) (delete)

An action in Subscribe to Mirage list (show)  
Context: Email (show)  
Contact: MirageOS-devel (show)  
Repeats: (never)  
<http://lists.xenproject.org/cgi-bin/mailman/listinfo/mirageos-devel>  
(add log entry) (edit)  
Created 2015-04-20 19:51 (Mon) (delete)

# OCAML

- ▶ FP
- ▶ Strict
- ▶ ML
- ▶ Изумительная документация
- ▶ Быстрый старт
- ▶ ГОДЫ промышленной эксплуатации
- ▶ Нативная компиляция «подо всё, что движется» и т.д.

```
1  module Hello (Time : Mirage_time_lwt.S) = struct
2    let start _time =
3      let rec loop = function
4        | 0 -> Lwt.return_unit
5        | n ->
6          Logs.info (fun f -> f "hello");
7          Time.sleep_ns (Duration.of_sec 5) >> loop (n - 1)
8      in
9      loop 1000
10 end
```

## УСТАНОВКА/НАСТРОЙКА ИНСТРУМЕНТОВ

- ▶ \$ brew install ocaml opam
- ▶ \$ opam init \*
- ▶ \$ opam switch 4.04.0+flambda
- ▶ \$ opam install -y ocamlfind utop core merlin \*
- ▶ \$ opam install mirage

/evrone.

---

**HELLO**

<https://github.com/argent-smith/mirage-hello>

```
1. emacs /Users/paul/Development/own/mirage/hello (Emacs)
1 |   /Users/paul/Development/own/mirage/hello:
2 |   total used in directory 48 available 4290874134
3 |   drwxr-xr-x  9 paul  staff  306  8 май 19:32 .
4 |   drwxr-xr-x  3 paul  staff  102  8 мар 00:58 ..
5 |   drwxr-xr-x 15 paul  staff  510  8 май 19:32 .git
6 |   -rw-r--r--  1 paul  staff  146  8 мар 01:01 .gitignore
7 |   -rw-r--r--  1 paul  staff  109 12 мар 16:31 .merlin
8 |   -rw-r--r--  1 paul  staff   85 12 мар 11:24 README.md
9 |   -rw-r--r--  1 paul  staff   42 13 мар 00:46 _tags
10 |  -rw-r--r--  1 paul  staff  157 13 мар 00:26 config.ml
11 |  -rw-r--r--  1 paul  staff  444 14 мар 01:38 unikernel.ml
```

585 12: 0 UUUR-~/Development/own/mirage/hello All Dired by name yas company Helm WS +

The screenshot shows an Emacs window with a dark theme, displaying OCaml code. The buffer title is "1. emacs /Users/paul/Development/own/mirage/hello (Emacs)". The code defines a module "Mirage" with a function "main" that is foreign to OCaml, calling a "Unikernel.Hello" function. It also registers a "hello" service with a default time duration.

```
1 | open Mirage
2 |
3 | let main =
4 |   foreign
5 |     ~packages:[package "duration"]
6 |     "Unikernel.Hello" (time @-> job)
7 |
8 | let () =
9 |   register "hello" [main $ default_time]
```

At the bottom of the screen, the status bar shows the file name "config.ml" and the mode "All -multiple-loops Tuareg utop +".

157 10: 0 UU- --[hello]config.ml  
End of buffer

All -multiple-loops Tuareg utop +

The screenshot shows an Emacs window with a dark theme, displaying OCaml code. The code defines a module `Hello` that structurally matches `Mirage_time_lwt.S`. It initializes `durations` as [1; 2; 3; 4; 5] and `iterations` as 1000. A recursive function `loop` is defined to iterate through the durations, logging each iteration and sleeping for one second before continuing. Finally, it maps over the `durations` list and joins the results using `Lwt.join`.

```
1 | module Hello (Time : Mirage_time_lwt.S) = struct
2 |   let durations = [1; 2; 3; 4; 5]
3 |   and iterations = 1000
4 |
5 |   let start _time =
6 |     let rec loop count n =
7 |       match count with
8 |       | 0 -> Lwt.return_unit
9 |       | c ->
10 |         Logs.info (fun f -> f "hello from instance %i iteration %i" n (iterations - c));
11 |         Time.sleep_ns (Duration.of_sec n) >> loop (c - 1) n
12 |     in
13 |     ListLabels.map ~f:(loop iterations) durations |> Lwt.join
14 | end
```

1. emacs /Users/paul/Development/own/mirage/hello (Emacs)

444 15: 0 UU- -[hello]unikernel.ml All -multiple-loops Tuareg utop +  
End of buffer

The screenshot shows an Emacs window with a dark theme, displaying OCaml code. The code defines a module `Hello` that takes a `Time : Mirage_time_lwt.S` argument. It contains constants for durations (1 to 5 seconds) and iterations (1000). The main logic is a recursive function `loop` that logs a message and sleeps for a duration before recursing. Finally, it maps over a list of durations to perform the loop in parallel using `Lwt.join`. The file path is `1. emacs /Users/paul/Development/own/mirage/hello (Emacs)`.

```
1 | module Hello (Time : Mirage_time_lwt.S) = struct
2 |   let durations = [1; 2; 3; 4; 5]
3 |   and iterations = 1000
4 |
5 |   let start _time =
6 |     let rec loop count n =
7 |       match count with
8 |       | 0 -> Lwt.return_unit
9 |       | c ->
10 |         Logs.info (fun f -> f "hello from instance %i iteration %i" n (iterations - c));
11 |         Time.sleep_ns (Duration.of_sec n) >> loop (c - 1) n
12 |     in
13 |     ListLabels.map ~f:(loop iterations) durations |> Lwt.join
14 | end
```

444 15: 0 UU- -[hello]unikernel.ml All -multiple-loops Tuareg utop +  
End of buffer

```
1. emacs /Users/paul/Development/own/mirage/hello (Emacs)
```

```
1 | module Hello (Time : Mirage_time_lwt.S) = struct
2 |   let durations = [1; 2; 3; 4; 5]
3 |   and iterations = 1000
4 |
5 |   let start _time =
6 |     let rec loop count n =
7 |       match count with
8 |       | 0 -> Lwt.return_unit
9 |       | c ->
10 |         Logs.info (fun f -> f "hello from instance %i iteration %i" n (iterations - c));
11 |         Time.sleep_ns (Duration.of_sec n) >> loop (c - 1) n
12 |       in
13 |       ListLabels.map ~f:(loop iterations) durations |> Lwt.join
14 | end
```

444 15: 0 UU- -[hello]unikernel.ml

All -multiple-loops Tuareg utop +

End of buffer

```
1. emacs /Users/paul/Development/own/mirage/hello (Emacs)
```

```
1 | module Hello (Time : Mirage_time_lwt.S) = struct
2 |   let durations = [1; 2; 3; 4; 5]
3 |   and iterations = 1000
4 |
5 |   let start _time =
6 |     let rec loop count n =
7 |       match count with
8 |       | 0 -> Lwt.return_unit
9 |       | c ->
10 |         Logs.info (fun f -> f "hello from instance %i iteration %i" n (iterations - c));
11 |         Time.sleep_ns (Duration.of_sec n) >> loop (c - 1) n
12 |       in
13 |       ListLabels.map ~f:(loop iterations) durations |> Lwt.join
14 | end
```

<https://mirage.io/wiki/tutorial-lwt>

```
444 15: 0 UU- --[hello]unikernel.ml          All -multiple-loops Tuareg utop +
End of buffer
```

# СБОРКА И ЗАПУСК: UNIX

- ▶ \$ mirage configure
- ▶ \$ make depend
- ▶ \$ make
- ▶ \$ ./hello --help
- ▶ \$ ./hello

```
~/Development/own/mirage/hello · (multiple-loops)
> ./hello
2017-05-08 19:17:04 +03:00: INF [application] hello from instance 1 iteration 0
2017-05-08 19:17:04 +03:00: INF [application] hello from instance 2 iteration 0
2017-05-08 19:17:04 +03:00: INF [application] hello from instance 3 iteration 0
2017-05-08 19:17:04 +03:00: INF [application] hello from instance 4 iteration 0
2017-05-08 19:17:04 +03:00: INF [application] hello from instance 5 iteration 0
2017-05-08 19:17:05 +03:00: INF [application] hello from instance 1 iteration 1
2017-05-08 19:17:06 +03:00: INF [application] hello from instance 2 iteration 1
2017-05-08 19:17:06 +03:00: INF [application] hello from instance 1 iteration 2
2017-05-08 19:17:07 +03:00: INF [application] hello from instance 3 iteration 1
2017-05-08 19:17:07 +03:00: INF [application] hello from instance 1 iteration 3
2017-05-08 19:17:08 +03:00: INF [application] hello from instance 4 iteration 1
2017-05-08 19:17:08 +03:00: INF [application] hello from instance 2 iteration 2
2017-05-08 19:17:08 +03:00: INF [application] hello from instance 1 iteration 4
2017-05-08 19:17:09 +03:00: INF [application] hello from instance 5 iteration 1
2017-05-08 19:17:09 +03:00: INF [application] hello from instance 1 iteration 5
2017-05-08 19:17:10 +03:00: INF [application] hello from instance 3 iteration 2
2017-05-08 19:17:10 +03:00: INF [application] hello from instance 2 iteration 3
2017-05-08 19:17:10 +03:00: INF [application] hello from instance 1 iteration 6
2017-05-08 19:17:11 +03:00: INF [application] hello from instance 1 iteration 7
2017-05-08 19:17:12 +03:00: INF [application] hello from instance 4 iteration 2
2017-05-08 19:17:12 +03:00: INF [application] hello from instance 2 iteration 4
2017-05-08 19:17:12 +03:00: INF [application] hello from instance 1 iteration 8
2017-05-08 19:17:13 +03:00: INF [application] hello from instance 3 iteration 3
2017-05-08 19:17:13 +03:00: INF [application] hello from instance 1 iteration 9
```

## СПАСИБО ЗА ВНИМАНИЕ

- ▶ @argent\_smith
- ▶ [evrone.ru](http://evrone.ru)
- ▶ [mirage.io](http://mirage.io)
- ▶ [unikernel.org](http://unikernel.org)
- ▶ [ocaml.org](http://ocaml.org)
- ▶ [ocsigen.org/lwt](http://ocsigen.org/lwt)
- ▶ GH: [argent-smith/mirage-hello](https://github.com/argent-smith/mirage-hello)