open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch
What is HelenOS?

open source general-purpose
multiplatform POSIX-similar microkernel
multiserver operating system designed
from scratch
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch

Not a Linux/*BSD/etc. distribution
SPARTAN microkernel
Custom user space
http://www.helenos.org
open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch

3-clause BSD license
bzr://bzr.helenos.org/mainline
http://trac.helenos.org
http://www.ohloh.net/p/helenos
What is HelenOS?

open source general-purpose
multiplatform POSIX-similar microkernel
multiserver operating system designed
from scratch
What is HelenOS?

open source general-purpose
multiplatform POSIX-similar microkernel
multiserver operating system designed
from scratch.

Rather breadth-first than depth-first
Desktop
Server
Embedded
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel multi-server operating system designed from scratch

IA-32 and AMD64 (PC)
IA-64 (Itanium)
ARM (Neo FreeRunner, BeagleBoard, BeagleBone, Raspberry Pi)
MIPS (Malta)
PowerPC (iMac G4)
UltraSPARC (Ultra 60, Enterprise T1000)
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch

- Mostly familiar “common sense” API
- Unicode (no legacy character sets)
- No skeletons in the closet (strcpy, signals, fork & exec)
- libposix as an optional emulation layer
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch

Memory management
Scheduling
Asynchronous IPC
Not a trivial kernel
What is HelenOS?

open source general-purpose
multiplatform POSIX-similar microkernel
multiserver operating system designed
from scratch
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel

**multiserver** operating system designed from scratch

Component-based design
Separate and isolated user space tasks (servers)

Naming service, VFS, file system drivers, Location service, device drivers, network layers, graphics stack layers, etc.
What is HelenOS?

open source general-purpose multiplatform POSIX-similar microkernel multiserver operating system designed from scratch
Designed from scratch

- **SPARTAN kernel since 2001**
- **HelenOS since 2005**
- **Latest release 0.5.0 (Fajtl) in August 9th 2012**
- **Some 330K lines of code in the mainline branch**
  - About 93 % in C, 3 % in assembler
  - About 30 % of comments
    - “Very well-commented source code” [ohloh.net]
- **Some 45 contributors**
  - About 20 active during the last year
    (only 10 of them at least 10 commits)
async_exch_t *async_exchange_begin(async_sess_t *sess) {
    if (sess == NULL) {
        return NULL;
    }
    async_exch_t *exch;
    fibril_mutex_lock(&async_sess_mutex);
    if (!list_empty(&sess->exch_list)) {
        /*
         * There are inactive exchanges in the session.
         */
        exch = (async_exch_t *)
            list_get_instance(list_first(&sess->exch_list),
                              async_exch_t, sess_link);
        list_remove(&exch->sess_link);
        list_remove(&exch->global_link);
    } else {
        /*
         * There are no available exchanges in the session.
         */
        if (!((sess->mgmt == EXCHANGE_ATOMIC) ||
               (sess->mgmt == EXCHANGE_SERIALIZE))) {
            exch = (async_exch_t *) malloc(sizeof(async_exch_t));
            if (exch != NULL) {
                link_initialize(&exch->sess_link);
                link_initialize(&exch->global_link);
                exch->sess = sess;
                exch->phone = sess->phone;
            }
        }
    }
}
Commits
Lines of code
Design principles

- Microkernel design principle
- General-purpose design principle
- Non-fundamentalistic design principle
- Full-fledged design principle
- Multiserver design principle
- Split of mechanism and policy design principle
- Encapsulation design principle
- Portability design principle
Why?
HelenOS vs. Linux
HelenOS vs. Linux

HelenOS and Linux
A fatal exception OE has occurred at 0028:C562F1B7 in VXD ctpci9x(05) + 00001853. The current application will be terminated.

* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue
Why?

Reliability
Robustness
Dependability
Natural design
Dependability

- **IEEE definition**
  - “Dependability is a measurable and provable degree of system's availability, reliability and its maintenance support”

- **In other words**
  - Formal verification of correctness and quality of service with respect to predefined specification/criteria
Dependability (2)

- Semantic information in source code
- Architecture and behavior specification
- Architecture models
- Extra-functional properties
Dependability (2)

- Compiler checks
- Static analyzers
- Abstract interpretation

- Semantic information in source code

- Architecture and behavior specification
- Architecture models

- Compatibility checks
- Compliance checks
- Code generation

- Extra-functional properties

- Model checking
- Use case analysis

- Performance modeling
- Schedulability analysis
Functional properties
Frama-C

Pre/post-conditions checking
Invariant checking

Architecture Description
ADL/CDL with extensions

Behavior Description
{T|E|-}BP with preprocessing

HelenOS sources
C99 with GNU extensions

Stubs, skeletons, connectors generation
Run-type interface type checks

Interface properties checking
(Correct sequencing, live/deadlock freedom)

Use cases
domain-limited English

Behavior compatibility
Substituability

Extra-functional properties
timing properties

Extraction
Extraction

Component
Implementation properties
Frama-C

Pre/post-conditions checking
Invariant checking

Functional properties
Extra-functional properties

Timing properties

Behavior Description
Behavior Description

Use cases

Microkernel OS community

- **FOSDEM 2012**
  - February 4th – 5th 2012, Brussels, Belgium
  - Université libre de Bruxelles
  - Jakub Jermář chaired the *Microkernel OS Devroom*
    - Participation of HelenOS, Genode Labs, NOVA, MINIX, Hurd

- **FOSDEM 2013**
  - February 2nd – 3rd 2013, Brussels, Belgium
  - Genode Labs chaired the *Microkernel and Component-based OS Devroom*
# Microkernel OS Devroom

Room: K.3.201

## Sunday 2012-02-05

<table>
<thead>
<tr>
<th>Event</th>
<th>Speaker</th>
<th>Room</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome! or Why do we meet here today</td>
<td>Jakub Jermář</td>
<td>K.3.201</td>
<td>09:00-09:25</td>
</tr>
<tr>
<td>Introduction to the NOVA kernel API</td>
<td>Julian Steklina</td>
<td>K.3.201</td>
<td>09:30-10:00</td>
</tr>
<tr>
<td>Introduction to HelenOS</td>
<td>Jakub Jermář</td>
<td>K.3.201</td>
<td>11:10-11:55</td>
</tr>
<tr>
<td>The microkernel overhead</td>
<td>Martin Děcký</td>
<td>K.3.201</td>
<td>13:00-14:00</td>
</tr>
<tr>
<td>The agony of choice - the diversity of microkernels in Genode</td>
<td>Stefan Kalkowski</td>
<td>K.3.201</td>
<td>14:10-14:55</td>
</tr>
<tr>
<td>Dive into HelenOS Device Drivers</td>
<td>Jiří Svoboda</td>
<td>K.3.201</td>
<td>15:05-15:50</td>
</tr>
<tr>
<td>Panel discussion</td>
<td>Julian Steklina, Jakub Jermář, Ben Gras, Christian Helmuth</td>
<td>K.3.201</td>
<td>16:00-17:00</td>
</tr>
</tbody>
</table>
Learning by doing

- Keep the Moore's law in mind early during the design phase
- Do not put the kernel in charge of purely user space namespaces
- Too much synchronization spoils the kernel
Features

- **File systems**
  - ext4, FAT, exFAT, ISO 9660, UDF, MFS

- **Networking**
  - IPv4 & IPv6, NE2000, E1000, RTL8139

- **GUI**
  - Composing desktop

- **USB 1.1 (UHCI, OHCI)**
  - HID, mass storage

- **PATA, SATA (AHCI)**

- **GNU binutils, Portable C Compiler (PCC), MIPS simulator**
What next?
What next?

- Towards self-hosting
  - GCC, Clang
- Go
- VFS2
- FUSE
- BIRD, KnotDNS
What next? (2)

- User space driven system-wide scheduler
- User space driven SMP management
- New RCU algorithms
  - AP-RCU (highly portable, decently scalable PaR)
  - AH-RCU (highly scalable, microkernel-friendly)
- Implicitly shared resources management
  - De-duplicated caching, future usage prediction (read-ahead), resource pressure evaluation (out-of-memory conditions)
Join us!

www.helenos.org