Zpátky do školy?

http://d3s.mff.cuni.cz

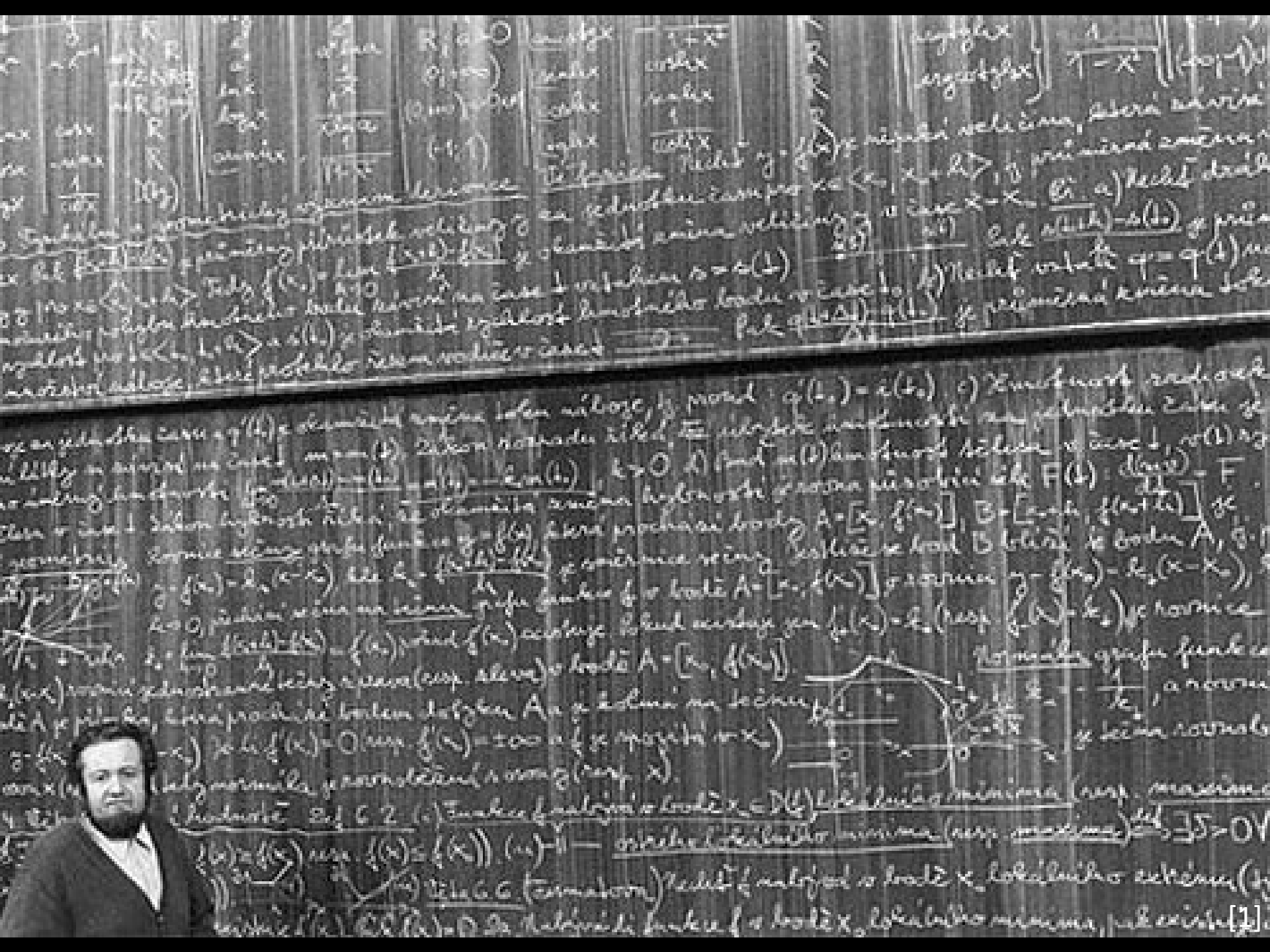
Department of Distributed and Dependable Systems

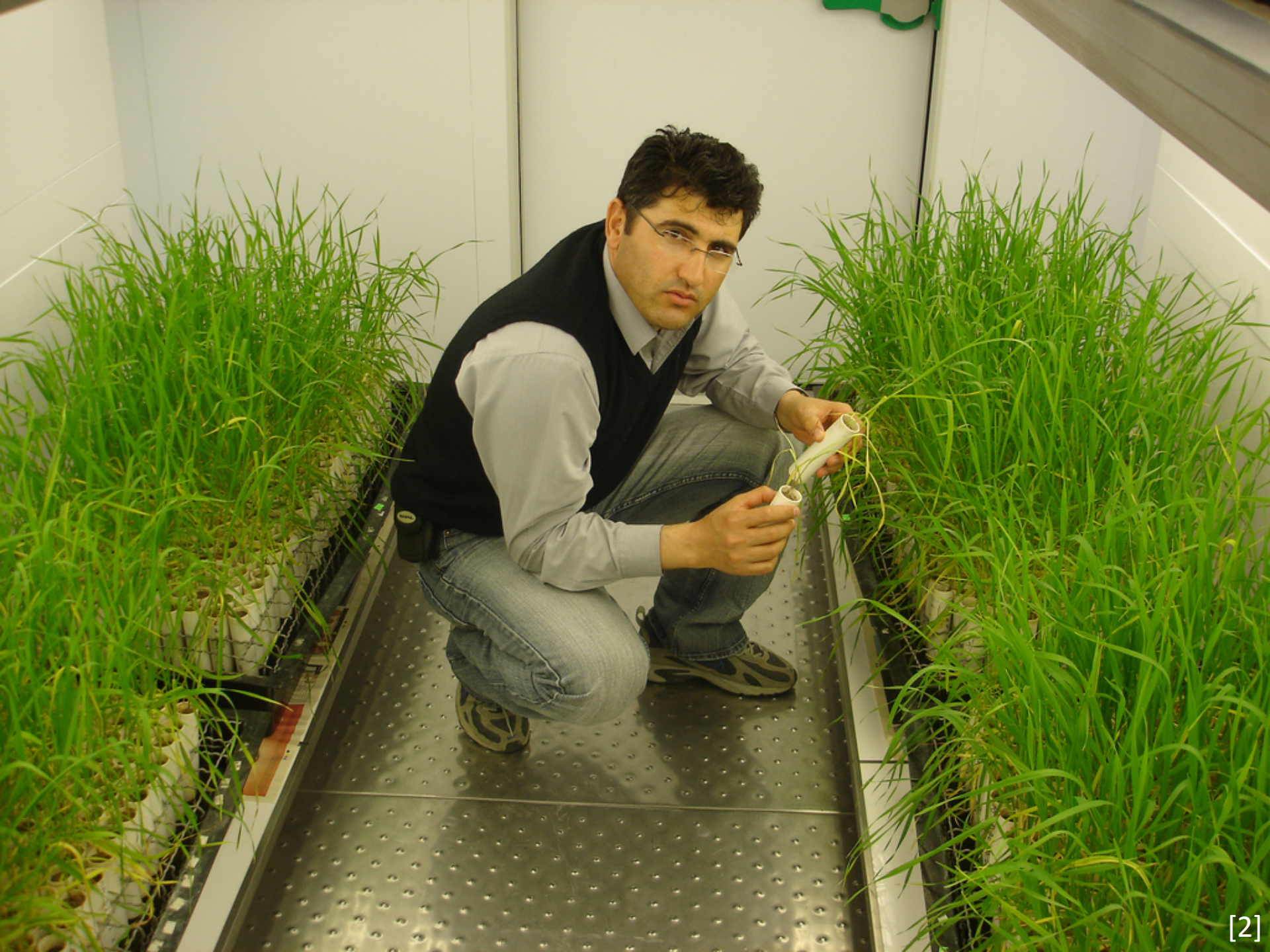
Martin Děcký

decky@d3s.mff.cuni.cz

CHARLES UNIVERSITY IN PRAGUE

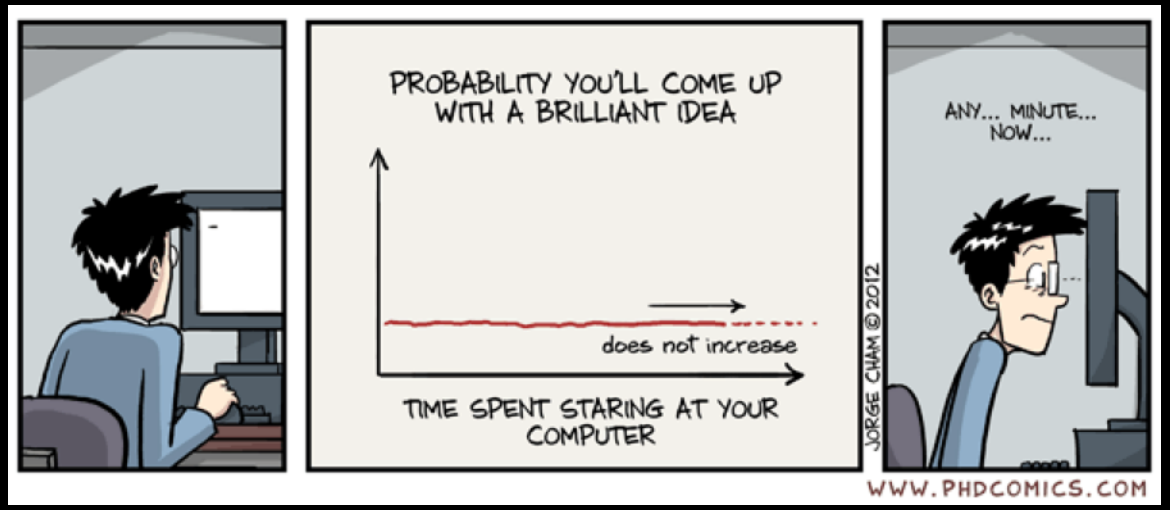
faculty of mathematics and physics





[2]

[3]


PROBABILITY YOU'LL COME UP WITH A BRILLIANT IDEA
does not increase
TIME SPENT STARING AT YOUR COMPUTER
ANY... MINUTE... NOW...
JORGE CHAM © 2012
WWW.PHDCOMICS.COM
[4]

HB 29971109F
B2
100
THE UNITED STATES OF AMERICA
HB 29971109F
100
ONE HUNDRED DOLLARS

[5]

[6]

Latest news

LMU (Project Coordinator) best German University in Excellence Initiative

AWARENESS Summer School AWASS 2012

Elsevier Theory most downloaded article 2011 of ASCENS member M. Dorigo

Blog

The Need of Rigorous Design of Ensembles
The increasing immersion and interaction of computing systems in both physical systems and...

Which is the best robotic scenario to test our ensembles?
This question is a recurrent one in many projects dealing with mobile robotics...

Engineering Distributed Adaptive Systems using Components
Traditional software engineering methodologies together with related programming paradigms have long been guiding...

Facebook

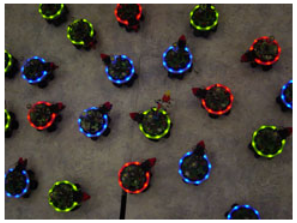
ASCENS
Like 32

SEVENTH FRAMEWORK PROGRAMME

ASCENS Profile

Project goal

Self-aware, self-adaptive and self-expressive autonomic components, running within environments which are called "ensembles", have been proposed to handle open-ended, highly parallel, massively distributed systems that can span millions of nodes with complex interactions and behaviours. However, these complex systems are currently difficult to develop, deploy, and manage.



The goal of the ASCENS project is to build ensembles in a way that combines the maturity and wide applicability of traditional software engineering approaches with the assurance about functional and non-functional properties provided by formal methods and the flexibility, low management overhead, and optimal utilization of resources promised by autonomic, adaptive, self-aware systems.

Project information

Information Society Technologies (IST) project funded by the EU as Integrated Project (IP) in the 7th Framework Programme (FP7) as part of the Future Emerging Technologies Proactive Initiative (FET Proactive).

Partners: 14
Countries: 6 EU member states and 1 associated state
Coordinator: Martin Wirsing, LMU Munich

Start: October 1st, 2010
End: September 30th, 2014

ASCENS Topics

Service-component ensemble language

Service-Component Ensemble Language (SCEL), developed as a multi-layer language for self-aware, autonomic service components (SCs) and service-component ensembles (SCEs) that integrates behavioural description with knowledge representation and reasoning about the environment.

Formal methods

Theoretical foundations and models for reliable and predictable system behaviour while exploiting the possibilities of highly dynamic, autonomic components.

Adaptation and self-expression

Adaptive systems for the integration of top-down and bottom-up approaches to adaptation and self-expression.

Swarm robotics

Generation of robot swarm with both autonomous and collective behavior.

Cloud computing

New science clouds with resource-aware cloud computing.

Energy saving e-Mobility

e-Vehicles, which are goal oriented and fully aware of its own, its neighbours' and its environment's resources.

Last Updated on Wednesday, 07 November 2012 13:03

BEEN

Benchmarking Environment

BEEN is a generic tool for automated benchmarking in a heterogeneous distributed environment.

BEEN automates all steps of a benchmark experiment from software building and deployment through measurement and load monitoring to evaluation of results. The notable features include separation of the measurement from the evaluation and an ability to adaptively scale the benchmark experiment based on the evaluation. BEEN has been designed to facilitate automated detection of performance changes during software development (regression benchmarking).

BEEN is developed as a student's software project for Distributed Systems Research Group at Faculty of Mathematics and Physics of the Charles University in Prague (Czech Republic).

HelenOS

Menu

- Home
- Features
- FAQ
- License
- Documentation
- Download
- Sources
- Roadmap
- How to contribute
- Links

Ports

- amd64
- arm32
- ia32
- ia64
- mips32
- ppc32
- sparc64

Blogs

- Martin Decky
- Vojtech Horky
- Jakub Jermar
- Vineeth Pillai
- Tim Post
- Pavel Rimsky
- Jiri Svoboda

Navigation

- Search

About HelenOS

Rather sooner than later, HelenOS will become a complete and usable modern operating system, offering room for experimenting and research. HelenOS uses its own microkernel written from scratch and supports SMP, multitasking and multithreading on both 32-bit and 64-bit, little-endian and big-endian processor architectures, among which are AMD64/EM64T (x86-64), ARM, IA-32, IA-64 (Itanium), 32-bit MIPS, 32-bit PowerPC and SPARC V9. Thanks to the relatively high number of supported architectures and suitable design, HelenOS is very portable. On top of the microkernel, HelenOS provides services such as file systems, networking, device drivers and user interface. Most of these services are composed of multiple independent server processes, which makes HelenOS one of the most modular operating systems.

As of now, HelenOS is being developed mostly by faculty members, and former and contemporary students of Faculty of Mathematics and Physics at Charles University in Prague. Nonetheless, the project is open for everyone, so we also have developers with different backgrounds from various places around the world. The source code is open and available under the BSD license. Some third party components, and components based on GPL software, are licensed under GPL.

In case you are interested in our project or have any questions about it, feel free to subscribe to our mailing list or chat with us on our IRC channel. The HelenOS operating system is, as of today, feature incomplete and the project is currently under heavy development (see roadmap). We are looking for people to join our team as co-developers or to merely try out our system and become our beta testers. If you have the skills and enthusiasm, you may consider making a contribution.

Mostly written in C
Mature, well-established codebase
Large, active development team
Very well-commented source code
Decreasing year-over-year development activity

HelenOS, updated Nov 26, 2012 more at

HelenOS

Main Language: C
Total Lines of Code: 469,877
Active Contributors: 21
Commit Activity Timeline:

5 year commit count

Updated Nov 26, 2012 more at

HelenOS 0.5.0 (Fajtl) Released!

Submitted by admin on Thu, 08/09/2012 - 15:34

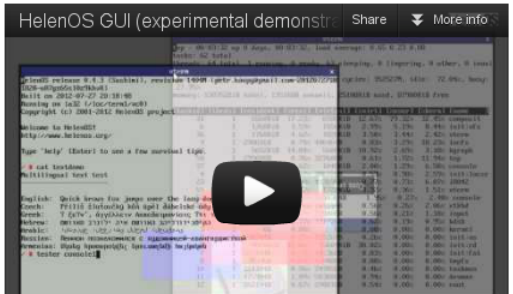
The HelenOS team is proud to announce the immediate availability of HelenOS 0.5.0, the next major release of our microkernel multiserver operating system. The 0.5.0 release is codenamed Fajtl, in honour of Lieutenant General František Fajtl, a Royal Air Force fighter pilot from the World War II.

This major release delivers many improvements that have been developed in our mainline branch since March 2011, including the deliverables from GSoC 2011. Some of the noteworthy new features are USB support, reimplemented networking stack with full TCP support and new network drivers (including Realtek RTL8139 and Intel E1000), read-only ext2 and ISO 9660 file system support, read-write MINIX FS support and several ported applications (GNU Binutils, PCC, MSIM). For a complete list of user-visible changes, see the release notes. The release files can be downloaded from the download page.

HelenOS GUI - experimental demonstration

Submitted by jermar on Mon, 08/06/2012 - 00:02

Petr Koupý apparently likes to create HelenOS screencasts during the summer break and we like watching them. Here is his latest one demonstrating the initial basic implementation of HelenOS graphics stack and GUI.



Quick links

- Browse sources
- Bug tracking
- Mailing list
- IRC Channel
- Community Wiki
- Login

Recent projects

Advanced hash table	70 %
Bithenge	70 %
Genode/SPARTAN	70 %
UDF file system	70 %
Display server	75 %
Audio support	50 %
SATA	90 %
Networking	95 %
Device drivers interface	95 %

More ...

Development

Revision: 1724
Developer: Jakub Jermar
Date: 2012-11-26
Time: 16:42:54 +0100
Make sure random() is seeded with a value that will not cause it to return negative numbers.

SEGA
MANX
Superbike
MIDWAY
SEGA

Zpátky do školy?

Zpátky do školy?

Proč ne!

Department of Distributed and Dependable Systems

References

[1] http://www.math.u-bordeaux1.fr/~pdelmora/big-blackboard-math.jpg
[2] http://www.flickr.com/photos/cimmyt/5453657777/
[3] http://d13pix9kaak6wt.cloudfront.net/background/srakyi_1315397939_73.jpg
[4] http://www.phdcomics.com/comics.php?f=1479
[5] http://www.flickr.com/photos/amagill/3366720659/
[6] http://www.flickr.com/photos/caribb/194965686/