Doctoral thesis review

Thesis title: Application of Software Components in Operating System Design

Thesis author: Mgr. Martin Děcký
Charles University in Prague
Faculty of Mathematics and Physics

Reviewer: Ing. Michal Sojka, Ph.D.
Czech Technical University v Praze
Faculty of Electrical Engineering

The submitted doctoral thesis deals with design, development and verification of a microkernel multiserver operating system (OS) called HelenOS. The main domain of the thesis is software engineering.

General comments

HelenOS was started more than ten years ago as a student project and until these days it is continuously developed by a small development team. The author of the thesis is a member of that team and plays there an important role. Although his contribution to the actual implementation of the operating system is significant, the thesis deals mainly with various software engineering aspects rather than with the implementation itself. Compared to many other software engineering projects, this thesis emphasizes the importance of software verification and certain HelenOS design decisions are heavily influenced by this fact.

The thesis is structured into several chapters. Chapter 3 states the HelenOS goals, which are: practical research and development platform, reliability and practicality. All of these goals are clearly motivated and to some extend also fulfilled. Having a practical and reliable platform with detailed knowledge of all the internals is a very valuable thing for research organizations. Although this platform was already used in several research papers by the thesis author, I think that the potential of the platform is much higher.

Chapter 5 surveys relevant operating systems and concludes with the HelenOS design principles, where some of them were inspired by the surveyed operating systems. At the top of the list of design principles is the "General-Purpose design principle", which means that the OS should not be optimized for a single purpose. It is unfortunate that real-time capabilities were explicitly excluded as being too specific to a single purpose. It has been shown in 2004 that a general-purpose OS (Linux) can be made a real-time OS at the same time (thanks to the *preempt-rt* patch). Nowadays, there is big industry demand for verified real-time operating systems and HelenOS could be more relevant if it provides real-time features.

Chapter 6 describes certain HelenOS features. While its content is sufficient for getting a basic idea about the HelenOS operating system, a lot of important details it either missing or mentioned only by referring to the literature or to other theses. I understand that the thesis has different goals than describing the implementation, but readers (or perhaps prospective HelenOS developers) would benefit from having more technical details here.

The development process of HelenOS is described in the next chapter. Although there is not much novelty – the development process is quite similar to many other successful open-source projects – I consider it quite an achievement for an academic project to maintain continuous development for about a decade and to keep code quality high even with many contributions from students. (Experience with student contributions is specifically described in Section 7.6.1; the author of this review read that section with deep understanding.) It seems that the successful execution of the development process can be credited mainly to the thesis author.

The thesis culminates in Chapter 8 devoted to verification of the HelenOS platform. The author motivates well the simultaneous use of different approaches to code verification and then describes how they are applied in the HelenOS context. An important achievement in this regard is that Coverity scan in 2006 did not detect any errors despite this tool is famous for detecting a lot of errors in other projects.

There are several interesting ideas presented in this chapter. One of them is the *abs32le* pseudo-target that is used to define behavior of platform-specific routines in a way understandable to various analyzers or verifiers. Another one is the use on non-preemtive fibrils that on one hand make programming more convenient, on the other hand simplify verification. And finally, the most distinguishing feature of HelenOS related to verification is the component-based nature of the OS and the fact that most OS components are described in formal way, which allows to verify correctness of component composition. Although all the work related to component-based verification is not yet finished, this thesis and related papers lay down good basis for future research and perhaps even industrial adoption.

Questions

1. The author mentions at several places that HelenOS uses advanced algorithms and data structures but what is missing is the evaluation of whether the benefits of the their use correspond to the expectation or at least to the results found in the literature. The question is how are these these things evaluated and how well is HelenOS doing compared to other mainstream or microkernel-based OSes. I would be interested in things like performance and scalability on real world use-cases.

2. While advanced data structures are used at many places, IPC uses "fixed-size kernel dispatch buffers". I tried to look up the corresponding code, but I only found places with memory allocation and linked lists in the IPC path. Could the author clarify the use of fixed-size buffers and what is the initial or optimal size of these buffers?

3. As I have already mentioned, I find missing real-time features a serious drawback. HelenOS kernel "does not support any kind of static priorities", but there are several run queues with different priorities in the scheduler. How are these priorities used?

Summary

To summarize the thesis, I would use the words from its beginning: “There is no silver bullet”. The author presents many techniques, principles and methods that are mostly known and by using all these in combination he achieves the stated goals – reliability and practicality. Even quick look at the source code is sufficient for concluding that its quality is exceptionally high. The thesis refers to all relevant related works and state of the art techniques known to this review author and places HelenOS into distinct position at the operating system landscape. The strong aspect of the thesis is the possibility of formal modeling the whole system (not only the kernel) and reasoning about its properties. There are of course many things that are either missing or not finished. The author does not hide them and it is not reasonable to expect them all being implemented given the resources of the development team. The results of the thesis will definitely be useful for further research and I would wish the author and the whole development also commercial success, but this might be hard.

In addition to the content, both design and the language of the thesis is very good and makes the thesis a pleasure to read. The thesis definitely proves author’s ability of independent creative work.

Prague, September 1, 2015 Ing. Michal Sojka, Ph.D.