

Project history

In 1999, David Zimmerli wrote an [article](#) in [EDM/2](#) in which he inaugurated the project aiming to replace OS2LDR, [OS/2](#) operation system loader. He wrote a stub program which was loaded by the boot partition [MicroFSD](#) and tested the use of its routines. The project was not finished until [osFree](#) project in 2003 has taken its sourcecode as a starting point of [osFree](#) kernel loader implementation. We left the name of the project as is because it resembled the name of [osFree](#) project, is free and [opensource](#), and used Zimmerli's program as a base. It was rewritten, so none of original code remained in it, but we decided leave the name as is. It's primary goal is to load [L4 microkernel](#), but we decided to make it as general purpose as [GRUB](#), and to fix some [GRUB](#)'s misfeatures and continue its general concepts, like [multiboot](#) compliance and OS-neutrality [history](#). We make our loader more modular and extensible. And, we developing support for kernels, other than [multiboot](#) ones. Loading of [Linux](#)-like kernels and support for [chainloading](#) unsupported kernels were moved to separate loader modules too. They were implemented as a separate [multiboot](#) kernels. The [multiboot](#) kernel for loading [OS/2](#) is being implemented now. [WinNT](#) can be loaded by [chainloading](#) the NTLDR. We decided to leave only [multiboot](#) support in base loader. Now [L4](#), [FreeDOS32](#), [ReactOS](#) and other [multiboot](#) compliant kernels can be loaded too.

The first [FreeLdr](#) programmers were Sascha Schmidt and Yuri Prokushev. They were planned to combine [MicroFSD](#) routines with [\[http://www.gnu.org/software/grub/\]](http://www.gnu.org/software/grub/)[GRUB](#)'s [Multiboot specification](#), to allow [L4](#) microkernel to be booted by [FreeLdr](#). [L4](#) uses [GNU GRUB](#) as default [bootloader](#), and uses its features extensively. This time nobody of us was skilled in assembly language, and a [FreeLdr](#) project was stalled for a while because of one problem with [MicroFSD](#) functions.

Then, the boot project was dreaming, and development continued in other fields.

After some time, in summer of 2005, a new team member, Valery Sedletski (aka valerius) joins the project. He had a little experience in assembly, so, the problem was resolved soon. In the beginning of 2006, the [FreeLdr](#) can load [multiboot](#)-compliant kernels, and [L4](#) microkernel was loaded. This time [FreeLdr](#) remains to be 16-bit. It uses native [OS/2 MicroFSD](#) routines directly. [GRUB](#)'s [multiboot](#) routines were ported to 16-bits as well.

But then it was realized, that the 16-bit internals make more difficult implementing the loader extensions and access the whole machine physical memory. The 64k limitation makes it hard to use more than 64k in loader executable. We were planned to use MSDOS style executables first. As our [bootloader](#) was written in C with small amount of assembly code, the 64k could be reached soon. Then we decided to switch to FLAT 32-bits memory model and protected mode, and take [GRUB](#) scheme as a sample.

Then, in the beginning of 2007, the new 32-bit protected mode loader was started, almost from scratch. We were still reusing [GRUB](#) routines for base [multiboot](#) support. The [GRUB](#) filesystem support code was recomposed to be loaded as separate loader plug-in modules. ([GRUB](#) has all FS support code compiled-in. We decided to move it to a separate "blackboxes"). Also, we proposed to introduce the new kinds of blackboxes to support different transparent decompression algorithms (<http://www.gnu.org/software/grub/>[\[GRUB\]](#)) had ungzipping on-the fly support, which was compiled-in too), terminal access, executable formats support etc.

history

Which means that the loader is abstracted from OS boot sequence details.

From:
<https://osfree.ru/doku/> - **osFree wiki**

Permanent link:
<https://osfree.ru/doku/doku.php?id=en:docs:boot:freeldr:history&rev=1363017094>

Last update: **2013/03/11 15:51**

