

— General information —

Aims of the project

Michael Felke

Martina Mostert



Revision 2009-03-11

In collaboration with
Marc Horst, Christian Janßen



Further on the following have contributed:

Oliver Höfken, Nicholas Vollmer



Contents

1 Description of aims	4
2 List of aims / requirements specification	4
3 Minimal computing system	6
Index	7



1 Description of aims

Aim of the project is the creation of a system consisting of models, specifications, test methods and tools, which makes it possible to use a program without re-compilation and regardless of used hardware on every computer, as long as:

- it has the functional ability 5
- there are sufficient resources available
- the runtime environment¹ has the needed programming interfaces.

It shall allow in particular:

- deliver driver, that can be used on every computer
- to boot therefore developed operating systems on different hardware. 10

This is to achieved using a generalized machine code in combination with a standard representation for binary storage of persistent data. An abstract and dynamic computing model enables machine programs, developed in that way, to adapt to the particular runtime environment. The call of programming interfaces shall also be unified, so that a program can be used with different operating systems, if these put the same by the program needed interfaces at its disposal. This is completed by a uniform package format. 15

At the same time the system to be created and particularly the generalized machine code shall have as few limits as possible.

2 List of aims / requirements specification

1. The system shall allow the building of programs independent of the target hardware. 20
2. Programs shall be executed with optimum utilization of the instruction set of the actual processor.
3. All the same it should be possible to write programs, which fully utilize the available hardware.
4. The generalized machine code to be created shall: 25
 - be interpretable
 - be easily translatable for target hardware
 - be usable by a JIT engine with adequate performance
 - make it possible to translate only the just needed parts (compiling on demand)
 - be creatable with any compiler language. 30

¹in the following called execution-sphere



2 List of aims / requirements specificatAims of the project

5. Deliver compiled resp. assembled programs in a unified format for all operating systems, that allows programs
 - to be installed from the package in an adapted form using a system program
 - to be used directly from the package via a loader
 - 5 • to be fetched from a server piece by piece
 - to be delivered in different variations (e.g. variant of language German/English), from which the appropriate is used
 - to connect always with the correct version at their interfaces
 - for different operating systems and hardware to be provided on one server
 - 10 • to be used automatically in the most up to date reversion
 - to be filed with other information like source code, help files etc.
6. The developed code shall be as memory-saving as possible.
7. A program shall only use as many resources as it really needs.
8. The developed programs have to offer a possibility for verifying the resources going to be used by them statically before execution.
- 15 9. Within the scope of a security system
 - different execution spheres shall be provided on a computer
 - the machine instructions available for the program shall be restricted
 - the changing of other programs shall be averted
 - 20 • memory protection shall be considered.
10. The program code shall be as storage technology (e.g. ROM, binary file, text file, database, ISAM file) independent as possible to be store-able in nearly every forms of today and tomorrow.
11. The read-access on program code or parts of it shall be possibly piped through procedures within the program itself to perform decoding and verification.
- 25 12. Concerning data processing the system shall allow data
 - to be saved in a computer independent format (also binary)
 - to be saved block oriented and structured
 - to be read and written high- as well as low-byte-first.
- 30 13. All atomic data kinds shall be processable.
14. Support of real-time-processing.
15. Support of multitasking/-threading programming.



3 Minimal computing system

A minimal computing system is defined as least common denominator from which on the above mentioned aims and requirements are realizable.

specification:

- 8-Bit, 6502 instruction set
- 512 Byte RAM
- optional ports (serial, parallel, etc.)
- ROM 32 KiB ²

5

²Kibibyte=1024 Byte, according IEC 60027-2, IEEE 1541

Index

- binary, 5
- binary file, 5
- block oriented, 5

- compilation, 4
- compiler language, 4
- compiling on demand, 4
- computer, 5
- computer independent, 5

- database, 5
- data kind
 - atomic, 5
- data processing, 5
- decoding, 5
- driver, 4

- execution sphere, 5

- hardware, 4, 5
- help file, 5
- high-byte-first, 5

- instruction set, 4
- interface, 5
- interpretable, 4
- ISAM file, 5

- JIT engine, 4

- low-byte-first, 5

- machine code, generalized, 4
- machine instruction, 5
- memory protection, 5
- memory saving, 5
- minimal computing system, 6
- multitasking, 5
- multithreading, 5

- operating system, 4, 5

- procedure, 5

- program, 4
- program code, 5
- programming interface, 4
- program variation, 5

- read-access, 5
- real-time-processing, 5
- resource, 4, 5
- reversion, 5
- ROM, 5
- runtime environment, see execution-sphere

- security system, 5
- server, 5
- source code, 5
- storage technology, 5

- text file, 5
- translatable, 4

- verification, 5