



Doctoral Thesis

Medos-2 a Modula-2 oriented operating system for the personal computer Lilith

Author(s):

Knudsen, Svend Erik

Publication Date:

1983

Permanent Link:

<https://doi.org/10.3929/ethz-a-000300091> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

23. Dez. 1983

Diss. ETH No. 7346

**Medos-2:
A Modula-2 Oriented Operating System
for the Personal Computer Lilith**

**A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH**

**for the degree of
Doctor of Technical Sciences**

**presented by
SVEND ERIK KNUDSEN
Dipl. Phys. ETH
born October 19, 1947
citizen of Thalwil (Zürich)**

**accepted to the recommendation of
Prof. Dr. N. Wirth, examiner
Prof. Dr. C.A. Zehnder, co-examiner**

N. Wirth

Zürich 1983

Abstract

Medos-2 is a single-user operating system designed and implemented for the personal computer Liliith. It is an *object-oriented* operating system conceived according to the concept of *open* systems. Its principal parts are the file system, the linking-loader, the part providing a "standard" terminal, and the command interpreter. All basic software for the Liliith computer is programmed in Modula-2. The operating system's interface to programs is therefore presented as a collection of separate Modula-2 modules and the operating system is a memory-resident Modula-2 program.

The purpose of this thesis is to show how Medos-2 provides both an easy to use and comfortable environment for developing and running Modula-2 programs by use of simple but powerful concepts for managing resources, for executing programs, and for handling files. The successful implementation, the efficiency, and the small size of the system are also due to this. The thesis also shows that it is possible to implement a realistic single-user operating system *completely* in Modula-2.

Medos-2 executes programs like "super-procedures". Any running program may activate another program. The linking-loader links a called program to its actual environment and checks the compatibility of separate modules. Both Liliith's architecture and the format of *object code files* contribute to the simplicity of the linking-loader.

Resources are managed by separate modules, typically one module for each kind of resource. A resource (e.g. an open file, main memory space) is *owned* by an activated program. Medos-2 provides several routines helping in implementing resource-managers, which may be non-resident.

The effectiveness of the file system may heavily influence the performance of a computer system. Medos-2's file system allocates disk sectors to files almost contiguously, and makes use of this fact by reading ahead on files. The buffering strategy is different for sequentially and randomly accessed files. This is enabled by simple statistics over file accesses and by the management of buffers by the generalized clock-algorithm.

Kurzfassung

Medos-2 ist ein Einbenutzer-Betriebssystem, welches für den Arbeitsplatzrechner Lilith entworfen und implementiert wurde. Es ist ein *objektorientiertes* Betriebssystem gestaltet gemäss dem Konzept von *offenen* Systemen. Die Hauptteile des Betriebssystems sind das Dateisystem, der Linking-Loader, der Teil, welcher ein "Standard" Terminal zur Verfügung stellt und der Commandinterpreter. Alle Grundprogramme für den Lilith-Rechner sind in Modula-2 programmiert. Die Schnittstellen des Betriebssystems für Programme sind deshalb eine Sammlung von separaten Modula-2 Modulen, und das Betriebssystem selbst ist ein speicherresidentes Modula-2 Programm.

Der Zweck dieser Dissertation ist zu zeigen, wie Medos-2 eine sowohl einfach zu benützende als auch komfortable Umgebung für die Entwicklung und Ausführung von Modula-2 Programmen zur Verfügung stellt. Dies wird durch einfache aber wirksame Konzepte für das Verwalten von Betriebsmitteln, für das Ausführen von Programmen und für die Behandlung von Dateien erreicht. Die erfolgreiche Implementierung, die Effizienz, und der kleine Umfang des Betriebssystems ist ebenfalls diesen Konzepten zuzuschreiben. Die Dissertation zeigt ebenfalls, dass es möglich ist, ein realistisches Einbenutzer-Betriebssystem komplett in Modula-2 zu implementieren.

Medos-2 führt Programme wie "Super-Prozeduren" aus. Jedes laufende Programm kann ein anderes Programm aktivieren. Der Linking-Loader bindet ein aufgerufenes Programm zu seiner aktuellen Umgebung und prüft die Kompatibilität der separaten Module. Sowohl Liliths Architektur als auch das Format des *object code file* trägt zur Einfachheit des Linking-Loaders bei.

Betriebsmittel werden durch separate Module verwaltet, typischerweise ein Modul für jede Betriebsmittelart. Ein Betriebsmittel (z.B. ein Datei, ein Hauptspeicherbereich) gehört einem aktivierten Programm. Medos-2 stellt mehrere Routinen zur Verfügung, die das Implementieren von eventuell nichtresidenten Betriebsmittelverwaltern erleichtern.

Die Effektivität des Dateisystems kann die Leistung eines Rechners gravierend beeinflussen. Das Dateisystem von Medos-2 alloziert, wenn möglich, benachbarte Plattensektoren zu Dateien und nützt diese Tatsache durch Vorauslesen auf Dateien aus. Verschiedene Pufferstrategie werden für sequentiell und direkt zugegriffene Dateien angewandt. Dies wird ermöglicht durch eine einfache Statistik über Dateizugriffe und die Verwaltung der Puffer mit dem generalisierten "clock"-Algorithmus.