



Doctoral Thesis

ORION

Konzept und Realisierung des Betriebssystems für ein allgemeines eng gekoppeltes Mehrprozessorsystem

Author(s):

Fischer, Rudolf H.

Publication Date:

1987

Permanent Link:

<https://doi.org/10.3929/ethz-a-000413372> →

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](#).

Diss. ETH Nr. 8206

ORION: Konzept und Realisierung des Betriebssystems für ein allgemeines eng gekoppeltes Mehrprozessorsystem

ABHANDLUNG

zur Erlangung des Titels eines
Doktors der technischen Wissenschaften
der
EIDGENÖSSISCHEN TECHNISCHEN
HOCHSCHULE ZÜRICH

vorgelegt von
RUDOLF H. FISCHER
dipl. El.-Ing. ETH
geboren am 28. Mai 1954
von Kilchberg (ZH)

Angenommen auf Antrag von
Prof. Dr. W. Guggenbühl, Referent
Prof. Dr. W. Schaufelberger, Korreferent

Zürich 1987

Zusammenfassung

ORION ist ein disk-orientiertes, allgemein verwendbares Betriebssystem, das im Rahmen des ETH-Forschungsprojektes M³ (Modulares Multi-Mikroprozessorsystem) für die M³ Ein- und Mehrprozessorsysteme des Instituts für Elektronik entwickelt wurde. Das Hauptziel dieser Arbeit liegt in der Schaffung einer Programmierumgebung für die experimentelle Untersuchung von Algorithmen auf parallelen Rechnerarchitekturen. Dabei steht die modulare Steigerung der Verarbeitungsleistung durch den Einsatz einer variablen Anzahl Prozessoren im Zentrum des Interesses.

ORION unterstützt eine weitgesteckte Klasse von Rechnerarchitekturen: Die Anzahl Prozessoren, Speicher und Peripheriegeräte ist beliebig; die Prozessor-Speicher-Verbindung ist an sich ebenfalls beliebig, vorgeschrieben ist lediglich, dass ein von allen Prozessoren effizient zugreifbarer gemeinsamer Speicher und für jeden Prozessor ein privater Speicher existiert.

Das Vorhandensein mehrerer Prozessoren in einem Computersystem erfordert eine entsprechende Ausrichtung sowohl der Anwenderprogramme als auch der Betriebssysteme. Im Betriebssystem ORION wird davon ausgegangen, dass die parallel ausführbaren Einheiten eines Programms, die "parallelen Prozesse", vom Programmierer explizit festgelegt werden und dass ihre Ausführungszeit relativ gross ist.

In einem Mehrprozessorsystem ist eine günstige Prozess-Prozessor-Zuordnung ("Konfiguration") eine wichtige Voraussetzung dafür, dass auch bei einfachen Prozessor-Speicher-Verbindungen eine grössere Anzahl Prozessoren effizient ausgenutzt werden kann. Unter dem Betriebssystem ORION erfolgt diese Zuordnung deshalb in zwei hierarchischen Stufen: Auf der höheren Stufe muss der Anwender eines Mehrprozessorsystems die vorhandenen Prozessoren zu "Ausführungseinheiten" (Pools) zusammenfassen, die Prozesse eines Programms in Gruppen (Tasks) unterteilen und jeden Task auf einen Pool statisch zuordnen. Auf der tieferen Stufe werden die ablauffähigen Prozesse der einem bestimmten Prozessor-Pool zugewiesenen Tasks durch das Betriebssystem dynamisch auf die einzelnen Prozessoren dieses Pools verteilt.

Um eine möglichst hohe Flexibilität zu schaffen, kann im Betriebssystem ORION jeder Prozessor für sich allein einen sogenannten "privaten" Pool bilden und daneben noch mit weiteren Prozessoren genau einem sogenannten "globalen" Pool angehören. Je nachdem, ob ein Pool nur einen einzigen Prozessor, mehrere oder alle Prozessoren eines Mehrprozessorsystems umfasst, lässt sich dadurch eine vollständig statische, teilweise statische/teilweise dynamische oder vollständig dynamische Prozess-Prozessor-Zuordnung realisieren.

Auf diesen Voraussetzungen aufbauend, werden die konkreten Spezifikationen für das Betriebssystem ORION erarbeitet. Anschliessend erfolgt der Grobentwurf und die detaillierte Realisierung. Die Beschreibung der detaillierten Realisierung muss sich infolge des grossen Umfanges des Betriebssystems allerdings auf einige wenige, ausgewählte Betriebssystem-Teile beschränken.

ORION ist praktisch vollständig in der höheren Programmiersprache Modula-2 geschrieben. Es umfasst neben den mehrprozessorspezifischen Funktionen zur Verwaltung der parallelen Abläufe und zur Prozess-Prozessor-Zuordnung (Pool- und Taskverwaltung) die üblichen, von Einprozessor-Betriebssystemen der mittleren Leistungsklasse her bekannten Funktionen wie Programmverwaltung, Programmlader, Ein/Ausgabe, Speicherverwaltung, Ausnahmebehandlung und Interruptbehandlung. Bei einigen dieser Komponenten, z.B. bei der Ausnahmebehandlung und der Ein/Ausgabe, bleibt das komplexere Verhalten eines Mehrprozessorsystems dem Anwender unsichtbar und äussert sich lediglich in einer komplizierteren betriebssystem-internen Realisierung. Andere Komponenten wie die Programmverwaltung und die Speicherverwaltung erfahren dagegen auch auf der Anwenderschnittstelle Erweiterungen.

ORION ist ein eigenständiges Betriebssystem und erlaubt den Einsatz der üblichen Hilfsmittel zur Programmentwicklung (Editor, Compiler, Linker) auf dem Mehrprozessorsystem selbst. Es enthält einen interaktiven Kommando-Interpreter, über welchen die auszuführenden Programme auf den gewünschten Prozessor-Pools gestartet werden können. Der Kommando-Interpreter ermöglicht im weiteren die Bildung der globalen Prozessor-Pools und die normalen, von Einprozessorsystemen her bekannten Operationen wie Arbeiten mit Files oder Definition von logischen Gerätenamen und Symbolen.

Einige Zeitmessungen an ausgewählten Betriebssystem-Funktionen und eine Leistungsbetrachtung anhand eines einfachen parallelen Sortier-Algorithmus zeigen das Leistungsverhalten des Betriebssystems ORION auf dem Mehrprozessorsystem M³. Eine grobe Abschätzung setzt zum Abschluss die mehrprozessorspezifischen Teile von ORION ins Verhältnis zu den auch in einem Einprozessorsystem notwendigen Betriebssystem-Komponenten.

Abstract

ORION is a disk-oriented, general purpose multiprocessor operating system. It has been designed and implemented in the scope of the research project M³ (Modular Multi-Microprocessor System) for the M³ single- and multiprocessor systems of the Electronics Department at ETH. The main goal of this thesis is to create a programming environment for experimental investigations of algorithms on parallel computer architectures. The purpose of using a variable number of processors is to achieve a modular increase of computing power.

ORION supports a variety of computer architectures with a variable number of processors, memories and peripheral devices. The processor-memory interconnection may be arbitrary, provided there is a common memory efficiently accessible by all processors and a private memory for each processor.

The use of several tightly-coupled processors in a computer system requires an adequate design of the operating systems as well as of the user programs. In the operating system ORION, the parallel executable entities of a program, called "parallel processes", must explicitly be fixed by the programmer; it is assumed that the execution time of a process is relatively long.

A suitable process-processor assignment ("configuration") is an essential requirement for an efficient use of more than just a few processors in a multiprocessor system, especially in a system with a relatively simple processor-memory interconnection. Under the operating system ORION, therefore, this assignment is done at two hierarchical levels: At the higher level, the programmer creates "execution units" (pools) by combining different processors, he subdivides the processes of a program into groups (tasks) and statically assigns each task of a program to one pool. At the lower level, the executable processes of all tasks of a pool are dynamically assigned to the processors of this pool by the operating system.

To provide a high flexibility, ORION treats each processor as a "private" pool; in addition, each processor may belong together with other processors to exactly one "global" pool. Depending whether a pool contains one, several or all processors of a multiprocessor system, a totally static, partly static/partly dynamic or totally dynamic process-processor assignment can be realized.

Based on these requirements, the specifications of the operating system ORION are defined. This is followed by the overall design and the detailed implementation of ORION. Due to the large size of the operating system, the description of the detailed implementation is restricted to some important parts of the operating system.

ORION is almost completely written in the high-level programming language Modula-2. In addition to the multiprocessor-specific functions for the management of the parallel processes and the process-processor assignment (pool- and task-management), it comprises most functions known from single processor operating systems: program management, loader, input/output, memory management, exception handling and interrupt handling. In some of these components (e.g. exception handling, input/output), the more complex behaviour of a multiprocessor system is not visible to the user, but it makes the implementation more complicated. Other components like program and memory management, however, become more complicated on the application layer as well.

ORION is a self-contained operating system which allows the execution of the usual program development tools (editor, compiler, linker) on the multiprocessor system itself. It contains an interactive command interpreter to load and start the executable programs on the desired pools. In addition, the command interpreter allows creation of the global pools and some operations commonly known from single processor systems, such as file management and definition of logical names and symbols.

Time measurements of some operating system functions and a speedup analysis of a simple parallel sorting algorithm show the performance of the operating system ORION on the M³ multiprocessor system. Finally, a rough estimation is given to compare the multiprocessor-specific parts of ORION with an optimized single processor operating system.