Insight ETHOS: On object-orientation in operating systems

Diss. ETH Nr. 9884

Insight ETHOS: On Object-Orientation in Operating Systems

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY (ETH) ZÜRICH

for the degree of
Doctor of Technical Sciences

presented by
CLEMENS ALDEN SZYPERSKI
Dipl.-Ing., Aachen Institute of Technology (RWTH), Germany
born on October 19, 1962
citizen of Germany and the United States of America

accepted on the recommendation of
Prof. Dr. N. Wirth, examiner
Prof. Dr. H. Mössenböck, co-examiner

1992

Abstract

Ethos: The characteristic spirit, prevalent tone of sentiment, of a people or community; the "genius" of an institution or system. (Aristoteles – Rhetorics ii. xii–xiv)

Ethos is both, an Elaborate Thesis on Operating Systems and an Ephemeral Thesis Operating System: It is the name of a project aiming at design principles of Object-Oriented Operating Systems, as well as the name of an actual implementation used as a case study to validate the found design principles. This thesis covers the Ethos project and system.

Results of the *Ethos project* are generalizations of design principles for object-oriented systems, namely the Carrier/Rider Separation, and the Directory Object concept. The former aims at a widely applicable scheme for separating data management from data access tasks. The latter generalizes the notion of prototype objects to manage integration of extensions into the running system. Another result of the Ethos project is the introduction of a flyweight pre-emption concept called *engines*, with the key attribute that engines may share a single stack.

The *Ethos system* is an evolutionary successor of the Oberon system, and is implemented using the language Oberon-2. The Ethos system is based on a strongly typed hierarchy of abstractions. Each abstraction has a default implementation, but multiple implementations of the same abstraction may be used simultaneously. As the extensibility of Ethos goes down to the very machine level, Ethos is well suited to prototype new operating system services.

Ethos contains a finalizing garbage collector with integrated support for identity directories, a facility for dynamic integration of modules into the running system, and a framework for developing extensions. The document editor Write has evolved as part of the Ethos system and is used as the primary medium for user activities. (Write has also been used to prepare this thesis.) A refined version of the Oberon user interface is provided in the standard setting.

Kurzfassung

Ethos ist zugleich der Name eines Projekts und der Name eines Systems. Das Projekt zielt auf die Isolierung und Bestimmung geeigneter Designprinzipien objektorientierter Betriebssysteme ab, während das System diesen Prinzipien folgend entworfen wurde. Als Fallstudie eines konkreten Betriebssystems untermauert das System die grundlegenden Gedanken des Projekts.

Wesentliche Resultate des Projekts sind Generalisierungen bekannter Designprinzipien objektorientierter Systeme. So werden "Model-View-Controller"-artige Konzepte zum *Carrier-Rider* Trennungsprinzip verallgemeinert, das eine breite Anwendung in der systematischen Trennung von Datenzugriff und Datenverwaltung findet. Weiter werden "Prototypes" zum *Directory Object* Konzept verallgemeinert, um eine feine Kontrolle über die Anbindung einer Implementationsvariante an eine gegebene Abstraktion zu ermöglichen. Schliesslich wird ein besonders leichtgewichtiges Konzept, genannt *Engine*, zur Realisierung von preemption-basierten Verfahren eingeführt. Das spezielle Merkmal von Engines ist die gemeinsame Benutzung eines Stacks.

Das Ethos-System, das vollständig in der Sprache Oberon-2 implementiert wurde, kann als evolutionärer Nachfolger des Oberon-Systems verstanden werden. Das Rückgrat von Ethos bildet eine streng typisierte Hierarchie von Abstraktionen. Zu jeder Abstraktion gehört mindestens eine Standardimplementierung – die gleichzeitige Verwendung mehrerer, alternativer Implementierungen der selben Abstraktion ist jedoch stets möglich. Da die Erweiterbarkeit und Austauschbarkeit von Abstraktionen im Ethos-System bis auf die Ebene der eigentlichen Maschine hinab reicht, kann Ethos auch zum Entwickeln von Prototyp-Implementierungen neuer Betriebssystemfunktionen verwendet werden.

Ethos ist mit einem finalisierenden Garbage Collector ausgestattet, der Identitätsverzeichnisse direkt unterstützt. Das dynamische Laden und Integrieren neuer Module in das laufende System ist jederzeit möglich. Die Konstruktion neuer Erweiterungen wird durch vorgezeichnete Rahmenkonstruktionen ("Frameworks") erleichtert. Als primäres Medium für alle Benutzeraktivitäten wird der als Bestandteil des Ethos-Systems entwickelte Dokumenteneditor Write verwendet. (Mit Write wurde auch die vorliegende Arbeit erstellt.) Schliesslich gehört eine Weiterentwicklung der Oberon-Benutzungsoberfläche zur Grundausstattung.