

# Inhaltsverzeichnis

1	Verteilte Betriebssysteme - Ein aktueller Ueberblick .....	1
1.1	Einleitung .....	1
1.1.1	Motivationen für die Verteilung eines Betriebssystems über mehrere Maschinen .....	1
1.1.2	Der Grad der Verteiltheit eines Systems .....	3
1.2	Exemplarische Betrachtung ausgewählter Systeme .....	6
1.2.1	Einführung .....	6
1.2.2	Apollo .....	8
1.2.3	Mach .....	14
1.2.4	Quicksilver .....	23
1.2.5	Argus .....	27
1.2.6	Eingliederung des ISO/OSI-Modells in die Betriebssystem-Umgebung .....	33
1.3	Entwicklungstendenzen im Gebiet verteilte Betriebssysteme .....	47
1.3.1	Eine vergleichende Bewertung basierend auf den betrachteten Systemen .....	47
1.3.2	Forschungsschwerpunkte in diesem Gebiet .....	51
2	Anwendung von objektorientierten Konzepten in der Betriebssystem-Umgebung .....	56
2.1	Einleitung .....	56
2.2	Betriebssystem-Aspekte von Smalltalk .....	57
2.2.1	Die virtuelle Maschine .....	58
2.2.2	Smalltalk-Klassen mit Betriebssystem-Funktionen .....	61
2.3	Was ist ein objektorientiertes Betriebssystem? .....	63
2.3.1	Was ist objektorientierte Programmierung? .....	63
2.3.2	Objektorientierte Konzepte auf der Betriebssystem-Ebene .....	64
2.3.3	Anforderungen an das objektorientierte Betriebssystem .....	73
2.4	Problematik der Verteilung eines objektorientierten Betriebssystems .....	76
2.5	Verteilte Datenbanken versus verteilte Betriebssysteme .....	81
2.5.1	Weshalb brauchen wir verteilte Datenbanken .....	81

2.5.2	Anforderungen durch verteilte Datenbanken ans Betriebssystem ..	83
2.5.3	Anwendung von Resultaten aus der Datenbank-Theorie .....	87
2.5.4	Objektorientierte Datenbanken.....	88
3	Ein integrierter Ansatz für eine Betriebssystem-Architektur, basierend auf Prolog....	95
3.1	Prolog - eine homogene Lösung von der Systemprogrammierung bis zur Anwendungsprogrammierung .....	95
3.2	Prolog für den Anwender - micro-Prolog .....	97
3.3	Prolog für Datenbank-Anwendungen - offene Probleme .....	97
3.4	Parallelität in Prolog.....	99
3.5	Die Verteilung von Prolog.....	100
4	Existierende Konzepte zur Ermöglichung von ereignisgesteuerter Programmierung.	102
4.1	Einleitung.....	102
4.2	Constraints .....	105
4.2.1	Implementation von Constraints.....	106
4.2.2	Auflösung statischer Constraints.....	107
4.2.3	Triggering von dynamischen Constraints .....	108
4.3	Hybrid - eine objektorientierte Sprache für die Programmierung mit aktiven Objekten .....	108
4.3.1	Überblick.....	108
4.3.2	Triggering von aktiven Objekten .....	110
4.3.3	Implementation .....	111
4.4	Active Values in Loops und Kee.....	112
4.4.1	Loops.....	112
4.4.2	Kee.....	113
4.5	Das SunNeWS-Postscript-Taskingsystem .....	114
4.5.1	Aktivation eines Prozesses.....	114
4.5.2	Implementation des Event-Interest-Systems .....	114
4.6	Ein Event/Trigger-Mechanismus zur Unterstützung semantischer Regeln (ETM).....	117
4.7	Folgerungen aus dem Vergleich schon existierender Mechanismen.....	118
4.7.1	Constraints.....	119
4.7.2	Active Values .....	119
4.7.3	Event/Trigger-Mechanismen.....	119
5	Eine Betriebssystem-Erweiterung zur Programmierung mit aktiven Objekten: .....	122
5.1	Einleitung.....	122
5.2	Beschreibung des Event-Distribution-Mechanismus (EDM) .....	123
5.2.1	Terminologie und Voraussetzungen .....	123
5.2.2	Aufbau und Funktionsweise des EDM .....	124
5.2.3	Die Klientenschnittstelle.....	128
5.3	Implementation des EDM mit NCS unter Sun's NFS.....	134
5.3.1	Implementation unter Verwendung von Apollo's NCS .....	134

5.3.2	Die Architektur des EDM.....	135
5.3.3	Die Event-Handler-Funktion.....	137
5.3.4	Die Implementation von blockierenden Events .....	139
5.4	Exemplarische Anwendungen des EDM.....	144
5.4.1	Ressourcen-Locking .....	144
5.4.2	Atomare Transaktionen.....	149
5.4.3	Konsistenzerhaltung replizierter Kopien eines Files .....	154
6	Dynamisch synchronisiertes Locking - Ein Locking-Protokoll für Ressourcen-Locking in einer statuslosen Umgebung .....	158
6.1	Einleitung und Zielsetzungen.....	158
6.2	Das statuslose versus das statusbehaftete Client-Server Modell .....	159
6.2.1	Allgemeines.....	159
6.2.2	Nachteile des statusbehafteten Servers in einer Netzwerk-Umgebung .....	161
6.2.3	Die Locking-Problematik für statuslose Server.....	162
6.3	Existierende Lösungen für das statuslose Locking-Problem .....	163
6.3.1	Sun's Lock-Manager .....	163
6.3.2	Weitere theoretische Ansätze.....	164
6.4	Dynamisch Synchronisiertes Locking - Das DSL-Protokoll .....	165
6.4.1	Die grundlegende Idee.....	165
6.4.2	Berücksichtigung von Client-Crashes .....	167
6.4.3	Berücksichtigung der Netzwerk-Verzögerung - Kollision von Lock-Requests.....	170
6.4.4	Unzuverlässiger Broadcast.....	172
6.4.5	Weitere Schwierigkeiten: Partitionierung des Netzwerks und mehrfache Client-Crashes.....	174
6.4.6	Zusammenfassung des DSL-Protokolls.....	175
6.4.7	Informeller Korrektheitsbeweis des DSL-Protokolls .....	177
6.4.8	Spezifikation in Pseudocode.....	179
6.5	Implementation von DSL unter Sun's NFS .....	181
6.5.1	Architektur des DSL-Services .....	181
6.5.2	Einige technische Details .....	182
6.6	Vergleich zwischen bereits existierenden Lösungen und DSL .....	183
6.6.1	Vergleich zwischen DSL und dem Sun-Lock-Manager .....	183
6.6.2	Vergleich zwischen dem DSL-Protokoll und anderen Algorithmen .....	185
6.7	Mögliche Modifikationen und Erweiterungen von DSL .....	186
7	Schlussfolgerungen und Ausblick.....	189
Literatur	.....	193
Nach Spezialgebieten .....		193
Betriebssysteme.....		193

Netzwerke und Distributed Computing .....	194
Verteilte Datenbanken .....	195
Objektorientiertes Design .....	196
Constraints und Triggering .....	197
Logische Programmierung und Prolog .....	198
Spezielle verteilte Betriebssysteme .....	199
EDM und DSL .....	201
Vollständiges alphabetisches Verzeichnis .....	202
Glossar .....	214
Stichwortverzeichnis Englisch - Deutsch .....	221
Index .....	224