Gesellschaft
für Informatik e.V.

Fachgruppe 3.1.4
Betriebssysteme

Prof. Dr. W. Schröder-Preikschat
Universität Potsdam
Institut für Informatik
Am Neuen Palais 10
14469 Potsdam

Tel.: (0331) 977 1032 / 1163
Fax.: (0331) 977 1720
E-Mail: wosch@cs.uni-potsdam.de

Mitteilungen

28. Januar 1997

Inhalt

Seite

1 Aus dem Leitungsgremium

Sie haben sicherlich bereits Ihre Beitragsrechnung für 1997 von der GI-Geschäftsstelle erhalten *und* der Geschäftsstelle Ihre Entscheidung zum zukünftigen Bezug der Fachgruppenmitteilung (Rundbrief) mitgeteilt. Falls Ihre Nachricht noch ausstehen sollte, teilen Sie sie bitte dem Absender der Umfrage, nämlich der GI-Geschäftsstelle, und nicht der Fachruppenleitung bzw. dem Fachgruppensprecher mit. Senden Sie den Antwortzettel auch dann zur GI-Geschäftsstelle, wenn Sie bereits vorher durch Zusendung Ihrer Email-Adresse an den Fachgruppensprecher Ihr Interesse für den elektronischen Bezug deutlich gemacht haben sollten.

Die GI-Geschäftsstelle wird einen Email-Verteiler anhand der bei ihr eingegangenen Informationen erstellen und der Fachgruppenleitung zur Verfügung stellen.

Bisher haben sich (leider nur) 175 von den 603 gemeldeten Fachgruppenmitgliedern für den elektronischen Bezug und damit für einen Fachgruppenbeitrag von 5,– DM entschieden. Beachten Sie bitte, dass der Weg über die Geschäftsstelle wichtig ist, um eine persönliche Einverständniserklärung über die von Ihnen gewünschte Beitragsstufe bei der GI vorliegen zu haben. Als "Seiteneffekt" wird dabei dann der Email-Verteiler für den elektronischen Versand erstellt.

2 WWW

Der WWW-Server der Fachgruppe ist von der TU Chemnitz zur FU Berlin umgezogen. In nächster Zeit werden GI-seitig die Referenzen auf die Einstiegsseite unserer Fachgruppe aktualisiert. Die neue URL lautet dann: `http://www.inf.fu-berlin.de/gi/fb3/fa31/fg314`.

3 ARCS '97

Die 14. ITG/GI-Fachtagung findet vom 8. bis 11. September 1997 in Rostock statt (`http://www.tec.informatik.uni-rostock.de/arcs97`). Die Fachgruppenleitung möchte Ihnen daher nochmals den **31. März 1997** als Eingangstermin für Positionspapiere zum Workshop "Betriebssystemforschung — quo vadis?" in Erinnerung rufen. Dieser Workshop wird von unserer Fachgruppe bestritten und soll als offene Podiumsdiskussion ablaufen, bei der folgende provozierende These zur Diskussion steht:

> In jüngster Zeit wird in Fachkreisen häufiger die Meinung vertreten, dass das Feld der Betriebssysteme weitgehend erforscht ist. Bedarf an grundsätzlich neuen Abstraktionen bestehe nicht. Generische Betriebssystemschnittstellen werden uns in Kürze in die Lage versetzen, unter Funktions-, Effizienz- und Zuverlässigkeitsgesichtspunkten optimierte Betriebssystemvarianten für die jeweilige Anwendung zu generieren (bzw. dynamisch aus dem Netz zu laden).

Das Podium wird aus Vertretern von Wissenschaft und Wirtschaft so besetzt, dass Befürworter und Gegner der These zu Wort kommen. Weitere Informationen dazu sind über Herrn Schröder-Preikschat (`wosch@cs.uni-potsdam.de`) bzw. `http://www.tec.informatik.uni-rostock.de/arcs97/ws3.html` erhältlich.

Gesellschaft
für Informatik e.V.

Leitung der Fachgruppe 3.1.4
Betriebssysteme

Einladung

zum Frühjahrstreffen
am 3./4. März 1997 in München

Die Schwerpunkte des Treffens sind diesmal:

"Windows NT" und "Mobile Objekte"

Zur Teilnahme füllen Sie bitte das beiliegende Anmeldeformular aus und schicken es bis spätestens

19. 2. 1997

an den lokalen Organisator

Prof. Dr. Uwe Baumgarten
Technische Universität München
Fakultät für Informatik
Arcisstraße 21
D-80290 München

Tagungsort:	Südgelände der TU München Arcisstr. 16 (Ecke Gabelsbergerstr.) Hörsaal S 1128
Beginn:	3. 3. 1997, 13.00 Uhr
Ende:	4. 3. 1997, 16.00 Uhr

Hotelreservierungen müssen selbst vorgenommen werden. Sonderkonditionen der TU München sind mit

Hotel Garni Stefanie,
Türkenstr. 35, 80799 München,
Tel 089-288-1400,
Fax 089-288-14049

vereinbart, Stichwort: "Fachgruppe Betriebssysteme". Für weitere Hotelhinweise und An- und Abreisemöglichkeiten steht Ihnen die Web-Seite des Frühjahrstreffens (`http://www13.informatik.tu-muenchen.de/fgbs/treffen97/`) zur Verfügung. Herr Baumgarten (Tel. 089-289-23259/25392, Fax 089-289-22037) oder Herr Hütter (Tel. 089-289-28247) helfen Ihnen bei Fragen gerne weiter.

Gesellschaft
für Informatik e.V.

Fachgruppe 3.1.4
Betriebssysteme

Herrn
Prof. Dr. Uwe Baumgarten
Technische Universität München
Fakultät für Informatik
Arcisstraße 21
D-80290 München

Anmeldung

zum Treffen der GI-Fachgruppe "Betriebssysteme" am 3./4. März 1997 in München.

Name: ....
Vornáme: ....
Institution: ....
Anschrift: ....

Tel.: .... Fax.: ....
E-Mail: ....

Ankunft: .... Abreise: ....

(Hiermit ist keine Hotelreservierung verbunden)

Unterschrift: .... Datum: ....

Gesellschaft
für Informatik e.V.

Leitung der Fachgruppe 3.1.4
Betriebssysteme

Programm

zum Frühjahrstreffen
am 3./4. März 1997 in München[1]

Montag, 3. März 1997	
13:00 – 13:45	**Feingranulares Monitoring von Echtzeit-Applikationen auf Windows NT**, M. Gergeleit, GMD-SET.RS
13:45 – 14:30	**Windows-NT-Cluster und -Hochverfügbarkeitslösungen**, T. Rothenwaldt, IBM Deutschland
14:30 – 15:15	**Mobile Computing mit SAP R/3 durch WWW und SMS**, R. Hüttl, iXOS Software GmbH
15:15 – 15:45	Kaffeepause
15:45 – 16:30	**Mobile Object Service — Ein Datenübertragungsdienst**, M. Hesselmann, TU Bergakademie Freiberg
16:30 – 17:15	**Laborautomation unter Nutzung von Windows NT, CORBA und OLE**, R. Kröger, B. Giegel, H. Mecke, M. Ruhl, Fachhochschule Wiesbaden
17:15 – 19:00	**Demonstration zum Betriebssystem-Praktikum**, W. Burke, Universität Karlsruhe
19:00 – ??:??	Abendessen
Dienstag, 4. März 1997	
08:30 – 09:15	**Verteilte Systeme für parallele und kooperative Problemlösungen**, P. Spies, TU München
09.15 – 10:00	**Dycos — Eine dynamisch anpaßbare Kernarchitektur zur Konstruktion verteilt-parallel kooperativer Systeme**, Chr. B. Czech, TU München
10:00 – 10:30	Kaffeepause
10:30 – 11:15	**Realisierung und Nutzung einer Mobile-IP-Umgebung**, U. Baumgarten, P. Selmayr, TU München
11:15 – 12:00	**Objektmobilität in Arts**, L. Büttner, J. Nolte, GMD-FIRST
12:00 – 14:00	Mittagspause
14:00 – 16:00	Sitzung der Fachgruppe

[1]Es stehen noch zwei Beiträge von SNI und Microsoft zum Thema Windows NT in Aussicht. Gegebenenfalls wird daraufhin das Programm noch erweitert. Informationen hierzu können vom FG-Sprecher bzw. über WWW eingeholt werden. Die möglicherweise noch vor dem Treffen eingegangenen Kurzfassungen werden auf elektronischem Wege nachgeschickt.

Gesellschaft
für Informatik e.V.

Leitung der Fachgruppe 3.1.4
Betriebssysteme

Kurzfassungen

der Beiträge zum Frühjahrstreffen
am 3./4. März 1997 in München

1. **Feingranulares Monitoring von Echtzeit-Applikationen auf Windows NT**, M. Gergeleit, GMD-SET.RS (Responsive Systeme)

   Seit seiner Markteinführung wird von Windows NT behauptet, es sei auch echtzeitfähig. Bei genauerer Betrachtung reduziert sich die Echtzeit-Unterstützung im Wesentlichen auf die Fähigkeit des Schedulers, für den Bereich der Echtzeit-Prioritäten auf jegliche Beeinflussung der Prioritäten zur Laufzeit zu verzichten. Solche "Echtzeit"-Prozesse und Threads erhalten den Prozessor dann durch ein reines preemptives, prioritätengesteuertes Dispatching. Ähnliche "Echtzeit-Prioritäten" gibt es bereits in vielen Unix-Varianten. Besonders interessant und oftmals sehr unverständlich wird der Umgang mit diesen Proritäten durch die Tatsache, dass der NT-Kern selbst (im Gegensatz zu allen kommerziellen Realzeit-Kernels und älteren Unix-Derivaten) eine ganze Reihe von Threads zur Bereitstellung der Betriebssystem-Dienste benutzt, die auf "normaler" Prioritätsstufe laufen. Das hat zur Folge, dass wann immer eine Echtzeit-Applikation Systemaufrufe tätigt, damit ein Thread-Wechsel (zu einem niederprioren Thread) verbunden ist, und die Gefahr der Verdrängung durch einen anderen Applikationsthread mit mittlerer Priorität gegeben ist. Es liegt dann also eine Art von "Priority-Inversion" vor. Darüber hinaus besteht selbstverständlich immer die Möglichkeit, dass ein Applikations-Prozeß höchster Priorität das gesamte System (inklusive jeglicher Interrupt-Service-Routinen) lahmlegt, indem er einfach den Prozessor nicht mehr freiwillig abgibt.

   Alle diese Probleme könnten nun dazu führen, dass auf die Implementierung von Echtzeit-Applikationen auf Windows NT verzichtet wird. Aus ökonomischen Gründen und auch aufgrund der vielen technischen Vorzüge des Systems in anderen Bereichen werden aber in Zukunft wohl mehr und mehr zeitkritische Anwendungen auf NT entwickelt werden. So arbeitet SET.RS gerade im Rahmen des ESPRIT-Projektes EIVIS (Embedded Interactive VIdeo Server) an der Realisierung eines leistungsfähigen Videoserver-Dienstes auf einem Cluster von NT-Maschinen, die mit einem Hochleistungs-Backplane-Bus untereinander verbunden sind. Die Server-Applikation ist durch eine ganze Reihe von Prozessen implementiert, die alle engen Zeitbedingungen genügen müssen. Es zeigte sich, dass eine sehr genaue Kenntnis der exakten Vorgänge sowohl in der Applikation als auch im Kern notwendig ist, um das Verhalten insbesondere im Hochlast-Bereich verstehen zu können. Da die bisher existierenden Monitoring-Tools die benötigten Informationen nicht liefern konnten, wurde im Rahmen des Projektes in der GMD ein Werkzeug entwickelt, dass es ermöglicht, alle Thread-Wechsel, die Interrupt-Schachteln sowie deren Relation zu Benutzer-definierten Ereignissen in einem Trace aufzuzeichnen und zu analysieren. Mit diesem Monitor wurde es möglich, den genauen Schedule des Gesamtsystems nachzuvollziehen, die Wechselwirkungen zwischen Benutzer-Prozessen, Kern-Prozessen und Interrupts zu verstehen und das exakte Zeitverhalten der kritischen Ausführungen zu bestimmen.

Da NT (im Gegensatz zu den meisten reinen Realzeit-Kernels) kein API zum Einhängen eigener Routinen in den Thread-Wechsel (seit Version 4.0 gibt es hierfür einen undokumentierten Kern-internen Mechanismus) oder die Interrupt-Behandlung vorsieht, war es nötig, für die Implementierung eines solchen Monitors in den Kern einzugreifen. Wenn man davon ausgeht, dass es nicht einfach ist, die NT-Quelldateien von Microsoft zu erhalten, bietet die Einbindung eines "Kernel Device-Treibers" die einfachste Möglichkeit, den Kern zu manipulieren. Kernel Device-Treiber laufen in einem Adreßraum mit dem gesamten Kern und es ist ihnen ohne weiteres möglich, den Kern-Code zu patchen. Genau das macht auch der Treiber des Monitoring-Systems. Er instrumentiert den Thread-Dispatcher und die Basis-Interrupt-Behandlung, so dass diese bei jedem Thread-Wechsel und bei jedem Interrupt eine Ereignisbeschreibung mit einem hochgenauen Zeitstempel in einem globalen Puffer ablegen. Da dieser Puffer auch in Benutzer-Adreßräume eingeblendet werden kann, können auch Benutzer-definierte Applikations-Ereignisse in den Strom der Kern-generierten Ereignisse eingefügt werden. Diese Information kann dann gemeinsam vom Monitoring-System (on-line oder off-line) ausgelesen und visualisiert werden.

In nächster Zukunft ist daran gedacht, unsere bestehenden Werkzeuge zur automatischen Instrumentierung von C++-Programmen mit dem feingranularen Kern-Monitoring zu verbinden. Das Resultat wird ein Meßsystem, dass ohne weitere Maßnahmen des Benutzers sehr genaue Zeitanalysen für sehr kleine Programm-Stücke im normalen Programm-Ablauf erstellen kann, ohne dabei die Effekte von Thread-Wechseln oder Interrupts vernachlässigen zu müssen. Ein solches System würde bei dem in der Realität immer noch sehr schweren Problem der "Worst Case Execution Time"-Analyse sehr viel weiterhelfen können.

2. **Windows-NT-Cluster und -Hochverfügbarkeitslösungen**, T. Rothenwaldt, IBM Deutschland Informationssysteme GmbH, Open Systems Center

   Mit dem zunehmenden Einsatz von Windows NT für Business-Anwendungen wächst der Bedarf einerseits nach besseren Skalierungsmöglichkeiten und andererseits nach Ausfallsicherheit von NT-Systemen. Beide Ziele werden seit den achtziger Jahren bei anderen Betriebssystemen durch Cluster realisiert. Dabei muss unterschieden werden zwischen reinen Failover-Clustern, die lediglich zur Maskierung von Ausfällen eingesetzt werden, und High-Performance-Clustern als Shared-Disk-Architekturen.

   Inzwischen sind von mehreren Anbietern Failover-Lösungen für NT verfügbar als erste Stufe eines Clusters: Traditionelle Spezialisten für PC-Software versuchen, ihre Technologie von Novell oder OS/2 auf NT zu übertragen; UNIX-Hersteller wenden die ihnen vertrauten Verfahren an. Obwohl die resultierenden Produkte auf unterschiedlichen Prinzipien beruhen, lassen sich gemeinsame Probleme im Zusammenspiel von Betriebssystem, Cluster-Software und Anwendung herausarbeiten. Diese Probleme sind teilweise aus der Geschichte der UNIX-Cluster bekannt, einige sind allerdings spezifisch fuer Windows NT.

   Der Beitrag stellt die verschiedenen Ansätze für NT-Cluster vor anhand typischer Implementierungen und diskutiert Vor- und Nachteile sowie Möglichkeiten zur Weiterentwicklung. Als Vergleichsmaßstab dient dabei das OpenVMS-Cluster: OpenVMS teilt mit Windows NT viele Eigenschaften des Kernels, und es hat die anerkannt ausgereifteste Cluster-Implementierung in einem Industriesystem. Daher lassen sich hier gut die grundlegenden Probleme von Cluster-Architekturen darstellen und bezüglich ihrer Lösungsverfahren mit UNIX und NT vergleichen:

   - Kommunikation der Cluster-Nodes und Membership-Kontrolle,
   - Organisation des Zugriffs auf gemeinsame Ressourcen (z.B. Disks),
   - Verhalten bei Ausfall von Komponenten,
   - Sicherung der Konsistenz von System- und Anwendungsumgebung,
   - Verwaltung von Netzadressen,

- Skalierbarkeit,
- Single System Image aus Sicht des Benutzers, Programmierers und Systemmanagers.

Abschließend wird anhand von SAP R/3 als einer sehr komplexen Anwendung dargestellt, welche Konfigurationsmöglichkeiten es in der Praxis gibt, wieder im Vergleich von NT-Clustern untereinander und mit anderen Betriebssystemen. Insbesondere die bisherigen Erfahrungen aus der Sicht eines Systemmanagers verdeutlichen, dass hier noch ein weiter Weg zurückzulegen ist. Man kann zwar konstatieren, dass die besten NT-Produkte bereits jetzt den Anschluss an die UNIX-Welt erreicht haben, aber bezüglich horizontaler Skalierbarkeit und Single System Image sind noch keine befriedigenden Lösungen in Sicht. Die Möglichkeiten, die intern der NT-Kernel bietet, können wegen der nach außen gepflegten Kompatibilität zu DOS/Windows nur schwer ausgenutzt werden.

3. **Mobile computing mit SAP R/3 durch WWW und SMS**, R. Hüttl, iXOS Software GmbH

   Durch die geschickte Kombination von WWW im Internet und SMS (Short Message Service) im GSM verbinden wir mobile Benutzer mit PDA's erstmals über einfache Standardwege kostengünstig mit ihren Applikationen in der Homebase. Der Einsatz dieser beiden Technologien ist die Basis für eine ganze Reihe neuer Anwendungen. Am Beispiel von mobilen Applikationen im SAP R/3 werden WWW und SMS zu einem funktionierenden Gesamtkonzept vereint. Der Nokia 9000 Communicator ist dabei das mobile Endgerät für den Außendienstmitarbeiter. Damit erschließt sich der Logistik, die heute oft am Werkstor endet, die Außenwelt auf einer Standardschiene.

4. **Mobile Object Service — Ein Datenübertragungsdienst**, M. Hesselmann, TU Bergakademie Freiberg

   Die *Common Object Request Architecture* (CORBA) von der *Object Management Group* (OMG) definiert eine objekt-orientierte Client/Server Architektur, die von zahlreichen Firmen und Organisationen unterstützt, implementiert und angewendet wird. Die Integration mobiler Rechner in diese objekt-orientierte Architektur ist Gegenstand dieses Beitrages. Ein *Mobile Object Service* (MOS) wird entwickelt, der die *Quality-of-Service-Parameter* (QoS-Parameter) der mobilen Kommunikationsmedien unterstützt. Die Funktionalitäten dieses Services werden im Folgenden erläutert und die Ergebnisse der prototypischen Implementierung aufgezeigt.

   Werden mobile Rechner über mobile Kommunikationsmedien in ein verteiltes System integriert, so sind die *Quality-of-Service-Parameter* (QoS-Parameter) der verschiedenen Kommunikationsmedien zu verwalten. So unterscheiden sich die mobilen Medien von denen des Festnetzes durch höhere

Kosten, kleinere Bandbreiten und/oder höhere Ausfallwahrscheinlichkeiten während einer Datenübertragung. Um die QoS-Parameter zu verwalten, wird ein Server-Objekt eingeführt, der Mobile Object Service (MOS).

Der MOS definiert ein Kommunikationsprotokoll zwischen zwei kommunizierenden Objekten. Findet während der Kommunikation ein Verbindungsabbruch statt, so kann die Kommunikation an einem definierten Protokollpunkt fortgesetzt werden. Ein Neubeginn der Kommunikation ist nicht notwendig. Der Datentransfer kann dadurch verringert werden und es wird gewährleistet, dass alle gesendeten Daten empfangen werden.

Die Kosten des Datentransfers werden vom MOS verwaltet. Nehmen diese Kosten einen Wert an, der über einem vorher definierten Schwellwert liegt, wird der Client davon in Kenntnis gesetzt. Dem Clienten wird so ermöglicht, die Kommunikation abzubrechen, um die Kosten der Datenübertragung niedrig zu halten. Die Kosten können durch Komprimierung der Daten ebenfalls verringert werden, daher bietet der MOS Datenkomprimierungsmethoden an.

Authentisierung und Authentifikation des Senders und der übertragenden Daten werden durch Verschlüsselungsmethoden des MOS unterstützt.

Die Implementierung des MOS basiert auf dem CORBA-Produkt Orbix 2.02 der Firma Iona und wird unter WindowsNT 3.51 und Visual C++ 4.0 implementiert. Als Kommunikationsmedien dienen im flächendeckenden Bereich das WIN und das D2-Mobilnetz sowie im lokalen Bereich das FunkLAN Arlan.

Die Design-Phase des MOS und eine prototypische Implementierung werden gegenwärtig parallel durchgeführt. Einzelne assoziierte Objekte des MOS, wie z. B. das Kosten-Objekt, sind realisiert und werden nun in das Kommunikationsprotokoll des MOS integriert.

Die Datenübertragung über mobile Kommunikationsmittel mit dem derzeitigen Prototyp sind erfolgreich. Die zusätzlichen Protokollschritte stellen bei großen Datenübertragungen keine Performanzverluste dar.

5. **Laborautomation unter Nutzung von Windows NT, CORBA und OLE**, R. Kröger, B. Giegel, H. Mecke, M. Ruhl, Fachhochschule Wiesbaden

   Ziele der Automatisierung chemischer (Analyse-)Labors sind Kostenreduktion durch Optimierung der Abläufe bei der Probenbearbeitung und Vollständigkeit der Nachweisführung im Sinne der "Good Laboratory Practice (GPL)". Die meisten Labore sind derzeit gekennzeichnet durch eine Vielzahl von Geräten, die aus informationstechnischer Sicht heterogen in ihren Schnittstellen sind, teilweise über eigene Bediengeräte (z.T. auch PCs und Workstations) verfügen, i.d.R. aber isoliert betrieben werden. Arbeitsabläufe sind durch Regeln verschiedenster Art vorgegeben (Normen, GPL, Firmenstandards). Laboranten erhalten Anweisungen zur Probenbearbeitung häufig durch ein Laborinformations-Management-System (LIMS, speziell ausgeprägte Datenbank), dem auch ermittelte Ergebnisse über Datenendgeräte oder Arbeitsplatzrechner übergeben werden. Zielvorstellung ist es, die Geräte des Labors als "offenes", verteiltes System zu betreiben, Abläufe durch Vorgaben des LIMS unter Nutzung von Handhabungsgeräten zu automatisieren bei gleichzeitiger automatischer Erfassung aller tatsächlichen Bearbeitungsparameter und Probenergebnisse. In einer Zusammenarbeit zwischen der FH Wiesbaden und der Bayer AG Leverkusen wird versucht, dieser Zielvorstellung durch Technologiestudien mit demonstrierbaren Prototypen näher zu kommen. In den bisherigen i.w. im Rahmen von Diplomarbeiten und Werkverträgen durchgeführten Arbeiten wurden Windows NT sowie CORBA (IONA Orbix) und Microsoft OLE (bzw. DCOM/OLE) eingesetzt.

Laborgeräte und Abläufe werden objektorientiert modelliert (Verwendung der Booch-Methode). Zur Beschreibung der Schnittstelleneigenschaften der Geräte wurden CORBA IDL sowie Microsoft MIDL eingesetzt. Neben einfachen Analysegeräten (wie Waage, ph-Meter) wurde ein Fraktionssammlerlabor modelliert, das durch ca. 15 Interfaces beschrieben wird. Es enthält mehrere Teilsysteme, darunter einen Roboter-basierten Liquid Handler. Schnittstellenbeschreibungen abstrahieren von der Implementierung der angebotenen Funktionalität. So sind z.B. für den Nutzer einer Laborgeräteschnittstelle die sehr vielfältigen unterlagerten physikalischen Schnittstellen der Laborgeräte nicht mehr von Interesse. Mittels Vererbung von Schnittstellen können Familien von Geräteklassen dargestellt werden. Spezielle Eigenschaften von Geräten bestimmter Hersteller lassen sich z.B. in abgeleiteten Klassen isolieren, während die gemeinsame Funktionalität der Geräteklasse in einer allgemeinen Schnittstelle zusammengefaßt wird. Während in der ursprünglichen, aus der DCE-Umgebung stammenden MIDL Vererbung nur durch Aggregation von Schnittstellen "nachgebildet" werden konnte, wird in der aktuellen Version die Vererbung von Schnittstellen unterstützt. Aggregation eignet sich gut für die Beschreibung von Schnittstellen, die gemeinsam sichtbar sind, wie etwa die der Teilsysteme des Fraktionssammlers. Gleichermaßen gut lassen sich komplexe, zusammengesetzte Geräte entwickeln, deren Komponenten-Schnittstellen zur Implementierung der Schnittstelle des abstrakten Gerätes verwendet werden, aber selbst nach außen nicht mehr sichtbar sind. In der Beschreibung des Fraktionssammlerlabors trifft dies z.B. auf das Teilsystem Lösungsmittelmischer zu, das selbst aus zwei Pumpen, zwei Mehrwege-Ventilen und einem Überdruckmanometer zusammengesetzt ist.

Laborgeräte werden unmittelbar oder über proprietäre "Boxen" der Gerätehersteller mittelbar i.d.R. an serielle Schnittstellen von PCs angeschlossen.

Die in den Prototypen eingesetzten PCs sind untereinander über Ethernet vernetzt und benutzen Windows NT als Betriebssystem. Ein Software-seitig verfolgter Ansatz basiert primär auf dem CORBA-Modell und verwendet den Orbix ORB zur Implementierung. In diesem Ansatz werden Geräte als CORBA-Objekte oder OLE-Objekte gekapselt und sind über den Orbix ORB bzw. unter zusätzlicher Nutzung seiner CORBA/OLE Bridge zugreifbar. Zur Implementierung der Geräte-Server wie auch zur Implementierung von Anwendungen wurden bisher sowohl Visual C++ als auch Visual Basic eingesetzt. Eine Integration eines HP ChemLMS LIMS als CORBA-Server auf einem HP-UX-System ist geplant.

Der zweite Implementierungsansatz benutzt primär das mit Windows NT 4.0 verfügbare proprietäre Microsoft DCOM/OLE, das die Netzwerk-Variante von COM/OLE darstellt und auf einer Weiterentwicklung des Microsoft DCE RPC basiert, die in der DCOM-Beschreibung als Object RPC (ORPC) bezeichnet wird. Auch wenn CORBA als das klarere Konzept erscheint und durch ein großes Konsortium getragen wird, wird dem Microsoft-Ansatz von vielen Markforschern eine erfolgreiche Zukunft vorausgesagt (insbesondere der neuerdings als ActiveX bezeichneten OLE-Variante zur Integration in Web-Dokumente). Neben prinzipiellen Nachteilen, wie der Unmöglichkeit, Objektinstanzen referieren zu können, die von anderen Prozessen erzeugt wurden, erscheint die DCOM/OLE-Umgebung derzeit noch unvollständig (z.B. fehlt in DCOM ein Namensdienst; stattdessen werden auf allen Systemen lokal zu pflegende Registry-Einträge benutzt). Auch ist die Systemkonfiguration für derartige Anwendungen derzeit noch fehleranfällig und ihre Entwicklung verlangt eine sehr spezifische Umgebung (NT 4.0, VC++ 4.2, ATL ActiveX Template Library empfehlenswert).

Der Beitrag stellt die bisherigen Ergebnisse und Erfahrungen vor.

6. **Verteilte Systeme für parallele und kooperative Problemlösungen**, P. Spies, Technische Universität München

   Der Vortrag gibt einen Bericht über laufende Arbeiten im Projekt MoDiS zur Entwicklung von V-PK-Systemen. Das sind Rechensysteme, die Anwendern ein für parallele und kooperative Problemlösungen geeignetes Instrumentarium zur Verfügung stellen und auf LAN-vernetzten Workstations realisiert werden.

   Die Ziele des Projekts werden von zwei Seiten verfolgt:

   - Anwendungsseitig werden Konzepte angeboten, mit denen auf hohem Abstraktionsniveau Lösungsverfahren frei von realisierungsbedingten Beschränkungen und Details formuliert werden können.

- Realisierungsseitig wird ein verteiltes Betriebssystem entwickelt, das sich an unterschiedliche Anwendungsanforderungen anpassen kann und das Potential der vernetzten Workstations dazu nutzt, diese Anforderungen möglichst effizient zu erfüllen.

Ein mit den anwendungsseitig zur Verfügung stehenden Konzepten konstruiertes System besteht aus einer Familie von konzeptionell strukturierten Mengen aktiver und passiver Objekte mit wählbarer Granularität, die sich mit dem Fortschreiten der Berechnungen des Systems dynamisch verändert. Es ist ein abstraktes System, dessen Eigenschaften den Anwendungsproblemlösungen entsprechen. Die Realisierung des Systems auf den vernetzten Workstations erfolgt mit dem Betriebssystem, das entwickelt wird. Es besteht aus Stellen-Mikrokernen und aus einer Familie von Managermengen, deren Eigenschaften aus dem zu realisierenden System abgeleitet sind. Die jeweiligen Managermengen sind strukturiert; die Manager kooperieren zur Erledigung ihrer Aufgaben, die letztlich mit den vernetzten Workstations zur Verfügung stehenden Ressourcen den Anforderungen entsprechend zu verwalten.

Charakteristisch für ein entsprechendes System sind seine Struktur, die Abhängigkeiten zwischen den Objekten entsprechen, seine Dynamik, die dem Fortschreiten der Berechnungen entspricht, und seine Fähigkeit, sich flexibel an Anwendungsanforderungen anzupassen. Diese Anpassungsfähigkeit ergibt sich einerseits daraus, dass wesentliche Eigenschaften des Betriebssystems aus dem zu realisierenden System abgeleitet sind. Sie ergibt sich andererseits daraus, dass Anforderungen differenziert erfaßt und entsprechende Managemententscheidungen und -maßnahmen feingranular, differenziert und reversibel getroffen werden.

Die Arbeiten im Projekt MoDiS laufen; der zu leistende Entwicklungsaufwand ist beträchtlich. Für einige, wichtige Fähigkeiten liegen (vorläufige) Ergebnisse vor, über die berichtet wird. Sie geben Einblicke in das Potential, das mit dem verfolgten Entwicklungsansatz genutzt werden soll

7. **Dycos — Eine dynamisch anpaßbare Kernarchitektur zur Konstruktion verteilt-parallel kooperativer Systeme**, Chr. B. Czech, Technische Universität München

   Moderne verteilte Betriebssysteme (BSe) basieren in vielen Fällen auf stellenlokalen Minimalkernen (Mikrokernen/Nanokernen). In der Vergangenheit boten Mikrokerne eine fest definierte Anzahl grobgranularer Kernabstraktionen und ermöglichten die Konstruktion übergeordneter BS-Dienste durch Nutzung einer starren Kernschnittstelle. Dies ist heute in vielen Bereichen nicht mehr ausreichend. Die vielseitigen Charakteristiken der Anwendungen und die daraus entstehenden stark wechselnden Anforderungen an eine BS-Architektur machen es notwendig, Mechanismen für die Erweiterbarkeit und Anwendungsanpassung eines Gesamtsystems im Kern zu verankern. Der evolutionäre Schritt von starren, unflexiblen Kernarchitekturen hin zu erweiterbaren, anwendungsanpaßbaren Minimalkernen kennzeichnet den gegenwärtigen Entwicklungsstand bei Kernen der nächsten Generation.

Der *Dynamic Context Switching (Dycos-) Kernel* ist ein Vertreter dieser neuen, erweiterbaren und anpaßbaren Generation von Mikrokernen. Er soll dazu eingesetzt werden, den Managern (Agenten) des an der TU-München gegenwärtig entwickelten verteilt-parallel kooperativen Systems (MoDiS-OS) dynamisch anpaßbare Basisfunktionalitäten bereitzustellen. Die Softwarearchitektur des Kerns basiert dabei fundamental auf dem Baukastenprinzip und beinhaltet ein Konzept zur dynamischen Komposition von Kerndiensten und -strukturen. Basismechanismen und Kernabläufe sind in Dycos anwendungsanpaßbar, d.h. sie können individuell und dynamisch auf die Charakteristik der Anwendung zugeschnitten werden. Dies wird ermöglicht durch den Einsatz eines kerninternen Instrumentariums (Toolbox), über das sich Komposition und Anordnung von Kernbausteinen steuern läßt. Durch den Einsatz von Dycos können deshalb stellenlokale Ressourcen flexibler, feingranularer und auf verschiedenen Abstraktionsniveaus verwaltet werden.

8. **Realisierung und Nutzung einer Mobile-IP-Umgebung**, U. Baumgarten, P. Selmayr, Technische Universität München

   Der Wunsch nach allgegenwärtiger Kommunikation und Informationsverarbeitung macht in zunehmendem Maße den Einsatz von mobilen Endsystemen nötig und die technologischen Entwicklungen ermöglichen diesen auch. Da viele Anwendungen und Dienste die Protokollfamilie des Internets als Basis nutzen, ist es sinnvoll, auf der Netzwerkschicht das IP-Protokoll derart zu erweitern, dass mobile Endsysteme ohne Aufwand und ohne Veränderungen der IP-Eigenschaften dessen Dienste ortsunabhängig nutzen können. Das Resultat ist das Mobile-IP.

   In diesem Beitrag wird die Funktionsweise des Mobile-IP grob erläutert. Am Beispiel von Endsystemen mit Windows Betriebssystemen wird eine Realisierung der benötigten Komponenten zur Verwaltung der Mobilität vorgestellt. Zudem wird die Treiberarchitektur insbesondere mit den von Windows NT und Windows 95 anhängigen Unterschieden für die Protokollrealisierung vorgestellt. Auf dieser Grundlage können nun nicht nur die Dienste des Internets ortsunabhängig genutzt werden, sondern es können zudem Standard-Windows-Anwendungen basierend auf dem bidirektionalen Tunnelling netzwerkweit nutzbar gemacht werden.

9. **Objektmobilität in Arts**, L. Büttner, J. Nolte, GMD-FIRST

   Arts ist ein Rahmenwerk zur Konstruktion verteilter Objektsysteme, das aus einem Laufzeitsystem für objektgebundene Fernaufrufe für die PEACE-Systemfamilie hervorgegangen ist.

   Die adaptierbaren Basismechanismen von Arts implementieren einen globalen Objektraum, der von aktiven und passiven Objekten populiert wird. Diese Objekte können über ortstransparente Kommunikationsmechanismen miteinander kommunizieren. Daher bleibt die logische Topologie das Objektraumes erhalten, auch wenn Objekte z.B. innerhalb eines Rechnernetzes bewegt werden. Mobile Objekte sind deshalb in Arts die Grundlage zur Lastbalancierung und dynamischen Rekonfiguration von verteilten Systemen.

   Der Vortrag stellt die Mechanismen des Arts Systems und ihre Sparcheinbetteung in einen verteilten C++-Dialekt vor.