



# PowerKey 2/94

## Inhalt

### *Preface/Vorstand*

Editorial . . . . . 4

### *NiCE Internals*

Veranstaltungskalender . . . . . 5

### *Aktuell*

Quo vadis NeXT? . . . . . 6

### *Root*

Das Internet – was ist das? . . . . . 8

NEXTSTEP – Grundlage des OpenStep Standards 11

### *TeX*

TeX-Kurs, Teil 5 . . . . . 22

### *Leserbriefe*

Monitorprobleme, die Zweite . . . . . 28

«Sicherer» Kode jetzt doch entschlüsselt . . . 29



## Liebe Leserin, lieber Leser

Schon wieder? Das dürfte wohl die Reaktion manches aufmerksamen Lesers bzw. mancher aufmerksamen Leserin sein. Da stand doch genau an dieser Stelle in der letzten Nummer, dass (vorerst) nur noch vier Ausgaben des **PowerKey** geplant seien – und nun ist kaum ein Monat vergangen, und schon liegt die nächste Ausgabe vor!

Richtig sagt da der aufmerksame Redaktor und liefert auch gleich eine Erklärung nach. Bei der Drucklegung der letzten Ausgabe war noch nicht sicher, ob es überhaupt ein nächstes **PowerKey** geben würde. Neben Adriano Gabaglio (Exredaktor) hatte sich nämlich auch Matthias Heubi – der frühere Verleger – zum Rücktritt entschlossen. Keine besonders gute Ausgangslage für den Weiterbestand unseres Magazins.

Heute sieht es um das **PowerKey** wieder besser aus. Seit der letzten Ausgabe haben wir in Peter Burgdorfer einen neuen Verleger gefunden, der nun zusammen mit mir dafür sorgen wird, dass das **PowerKey** rechtzeitig in Eure Stuben kommt. Ich möchte Ihn ganz herzlich für sein kurzfristiges Einspringen danken. Es ist sicher auch in seinem Interesse, wenn ich kurz noch einmal einen Aufruf in Sachen Mitgliederartikel mache. Es ist unbedingt notwendig, dass Ihr, die Leser, aktiv an der Gestaltung des **PowerKey** teilnehmt! Und das könnt Ihr, indem Ihr einen eigenen Artikel verfasst und ihn der Redaktion zur Veröffentlichung zukommen lässt. Eine andere Möglichkeit besteht darin, den «Machern» von **PowerKey** konkrete Vorschläge zu gewünschten Beiträgen zu unterbreiten. Ganz allgemein würde ich mir einen stärkeren Feedback auf unser Tun erwünschen.

Nun zu einem anderen leidigen Thema. In der letzten Zeit kommt es leider häufiger vor, dass ein NiCE Meeting kurzfristig in eine *talk & copy* Veranstaltung umgewandelt werden muss! Als Redaktor habe ich mich bereits schon dabei ertappt, gewohnheitsmässig im Veranstaltungskalender «*Themen noch nicht bekannt*» zu schreiben. Das soll nichts anderes heissen, als dass es keine Themen gibt bzw. dass es niemanden gibt, der

einen Vortrag oder eine Vorstellung organisieren und durchführen möchte. In der Rubrik NiCE Internals steht jeweils der Satz «*Beiträge von Mitgliedern sind jederzeit willkommen*». Und das ist durchaus ernst gemeint!

Ich möchte einmal an die vielen durchgeführten Anlässe erinnern. So zum Beispiel an den Abend, an dem wir einen ersten Blick auf «NEXTSTEP for Intel» werfen konnten. Es besteht ganz klar ein Bedürfnis nach derartigen Veranstaltungen, was die Teilnehmerzahlen immer wieder gezeigt haben. Damit es weiterhin solche interessanten Abende gibt, ist (einmal mehr) Eure Mitarbeit gefragt. Der NiCE-Vorstand sucht schon lange jemanden, der die Organisation und Durchführung der Meetings übernehmen möchte. Zudem sind Vorträge zu NeXT-spezifischen Themen (im weitesten Sinn) und Vorstellungen z.B. von Software gesucht. Interessenten sollen sich bitte beim Vorstand melden.



Nun aber zum Inhalt: In der aktuellen Ausgabe von **PowerKey** findet Ihr einen Artikel unseres Präsidenten Neil Franklin zum Thema Internet und wie man sich darin zurechtfindet. Peter Lipps, Leiter Systems Engineering von NeXT Deutschland, hat uns einen Artikel über NEXTSTEP/OpenStep zur Verfügung gestellt, den wir in dieser Ausgabe veröffentlichen. Daneben erscheint bereits Teil 5 des TeX-Kurses zum Thema Tabellensatz und ein neuer Leserbrief über das Sorgenkind NeXT Monitor.

Ich wünsche allen viel Vergnügen mit dieser Ausgabe von **PowerKey**.

Dominik Moser, Redaktor



## Veranstaltungen

### NiCE-Meetings

**Wann:** jeden zweiten Dienstag im Monat  
Beginn 19.00 Uhr (bis ca. 21.30)

**Wo:** ETH Zürich, Hauptgebäude  
Raum HG F5  
Adresse: Rämistr. 101  
Tram 6/9/10, Haltestelle ETH/Uni'spital

**Themen:** Immer aktuell! Werden nach Möglichkeit im **PowerKey** bekanntgegeben.  
Beiträge von Mitgliedern sind jederzeit willkommen.

### talk & copy

**Wann:** jeden vierten Dienstag im Monat  
Beginn 19.00 Uhr

**Wo:** Informatik-Gebäude (IFW) der ETH  
Raum IFW A44  
Adresse: Haldeneggsteig 4/Weinbergstr.  
Tram 6/7/10/15, Haltestelle Haldenegg

**Themen:** Hier haben Mitglieder die Möglichkeit, Fragen zu stellen, Erfahrungen auszutauschen sowie die neuste Software aus unserem grossen Archiv zu kopieren. Bring doch einfach Deine Disketten oder besser Deine Festplatte mit! Die NiCE besitzt (fast) alle dazu erforderlichen SCSI-Kabel und -Terminatoren.

## FTP Server

Auf dem NiCE ftp-Server befinden sich die neusten Programme und Daten rund um den NeXT. Zugriff für Mitglieder entweder anonym über Internet oder persönlich während jeder talk & copy Veranstaltung.

## NiCE-Agenda 1994

12. Juli: Meeting  
Themen noch nicht bekannt

26. Juli: talk & copy

9. August: Meeting  
Themen noch nicht bekannt

23. August: talk & copy

13. September: Meeting  
Themen noch nicht bekannt

27. September: talk & copy

11. Oktober: Meeting  
Themen noch nicht bekannt

25. Oktober: talk & copy

## NiCE-Markt

Ich verkaufe ab sofort:

- NeXT Color Monitor Fr. 1000.-
- NEXTSTEP 3.2 Developer Fr. 2000.-

**Michael Forrer**  
Bardella  
7304 Maienfeld

Tel.: 081/330 10 01  
Fax: 081/330 10 02



## Quo vadis NeXT?

**Am 14. April trafen sich im Technopark Zürich all diejenigen NeXT- oder NEXTSTEP-Fans, die wissen wollten, was die nähere Zukunft für sie offenhält. Der Einladung der neuerdings im Technopark angesiedelten Uptime Object Factory Inc folgten – neben den offiziellen Vertretern von NeXT Europa – zwei helvetische VAD's und rund 100 Interessierte.**

### Pressekonferenz

Unter der Leitung von Max Winiger (Medienverantwortlicher von NeXT für die Schweiz) fand im Vorfeld der öffentlichen Veranstaltung ein Presse-Meeting statt. Anwesend waren neben Rainer Bernert (Marketingdirektor für Europa) und Peter Lipps (Leiter Systems Engineering), beide von NeXT Deutschland, auch die Herren René Fischer (Verkaufsleiter Credal AG) und Julio Cesar Salgado (Direktor der Andes Informatiques SA) sowie George Fankhauser (neu zuständig für Marketing & Public Relations) von Uptime.

Zuerst ergriff Rainer Bernert das Wort und stellte in wenigen Worten die Firma NeXT vor. 1985 von Steven P. Jobs gegründet, beschäftigt man heute nach der Einstellung der Hardwareproduktion weltweit gerade noch 230 Personen. Der Hauptabsatzmarkt liegt in den USA, danach kommen Europa und Asien. Innerhalb Europas wird zwischen dem britischen, französischen und dem deutschsprachigen Markt unterschieden, letzterer umfasst neben Deutschland und Österreich natürlich auch die Schweiz.

Die Ziele von NeXT für das laufende Jahr sind einfach zu umschreiben. Sie führen deutlich vor Augen wo das grösste Problem liegt, das der Firmenleitung immer noch zu schaffen macht: Trotz Lob und Anerkennung von allen Seiten war der Markterfolg von NEXTSTEP bisher eher bescheiden. NeXT möchte auch gar nicht die breite Masse der Computeranwender ansprechen. Vielmehr sollen im Sinne einer Nischenstrategie «innovative Entwickler» für NEXTSTEP gewonnen werden.

Damit würde NEXTSTEP auch in den Entscheidungsfindungsprozess einbezogen werden, wenn es im Rahmen eines Projektes z.B. darum geht neue Lösungen zu finden. Zu diesem Zweck will man bei NeXT Zwecken mit grossen etablierten Firmen eingehen. Auf diese Weise erschliessen sich NeXT auf einen Schlag bestehende Märkte mit einer sehr grossen Zahl von potentiellen Kunden. Zusammen mit Canon visiert man vor allem den asiatischen Markt an und hofft durch den japanischen Partner eine höhere Akzeptanz zu erreichen. Die Zusammenarbeit mit HP und Sun – den bedeutendsten Anbietern von Workstations – bringt NeXT zunächst einmal Zugang zu einem Grossteil aller existierenden Workstations. Zusätzlich dazu arbeiten NeXT und SunSoft an der OpenStep-Spezifikation. Weitere Ehen etwa mit Data General oder Digital sollen in der nächsten Zeit folgen.

Die Allianz mit der Softwaretochter von Sun Microsystems war gemäss Bernert ein enorm wichtiger Schritt für das Unternehmen. Mit der Kooperation melden die beiden Firmen ihren Führungsanspruch in Sachen objektorientierter Betriebssysteme und Entwicklungsumgebungen an. Die Spezifikation von OpenStep soll Ende Juni offengelegt und an Standardisierungsbehörden, wie die OMG (Object Management Group), übermittelt werden.

Nach Rainer Bernert hatten die schweizerischen Distributoren Gelegenheit sich kurz vorzustellen. Die Zürcher Unix-Spezialistin Credal vertreibt neu als Value Added Distributor (VAD) NEXTSTEP. Daneben bleiben die workstation AG für die Deutschschweiz und Andes Informatique für die Romandie die offiziellen VAD's. Die Verteiler arbeiten eng mit der Uptime Object Factory Inc zusammen, die als Kompetenz-Zentrum die VAD's bei Schulung, Consulting, Support und Entwicklung unterstützt. Anfang Juni dieses Jahres ist die Uptime zum autorisierten NeXT Trainingscenter ernannt worden. Mit fünf zertifizierten Trainern bietet sie ein komplettes NEXTSTEP/OpenStep Grundlagenprogramm an. Auf Kundenwunsch werden Workshops und Kurse zu Anwenderprogrammen und spezifischen Entwicklungstechniken angeboten. Im Bereich OpenStep zeigt die Uptime Inc schon heute Migrations- und Entwicklungsmöglichkeiten auf.



## Präsentation

Da die Pressekonferenz etwas länger als geplant ausfiel, verpasste ich leider den ersten Teil der anschliessenden öffentlichen Präsentation. Wie beim Pressemeeting auch wurde die Vorstellung von Julio Cesar Salgado ein Opfer seines unverständlichen Englisch und viel schlimmer noch, ein Opfer des damokleshaften Zeitdrucks. Danach zeigte Matthias Heubi wie schnell (und einfach) man mit Hilfe des Interface Builders eine Applikation unter NEXTSTEP erstellen kann. Ohne eine Zeile zu programmieren (abgesehen von den vorbereiteten Objekten :-)) schuf er in wenigen Minuten eine Anwendung aus der Musikbranche. Aus einer Datenbank wurden Bestellungen von CD's gelesen und angezeigt – dass dabei auch sämtliche Titel mit Spieldauer und das Plattencover angezeigt wurden, versteht sich wohl von selbst. Als Krönung wurde dem (zum Teil) staunenden Publikum die auf der jeweiligen CD enthaltene Musik vorgespielt.

Zum Schluss der Präsentation war dann Peter Lipps an der Reihe, der auch gleich klarstellte, dass er nicht den langen Weg von München gekommen sei, um sich von der fortgeschrittenen Zeit unter Druck setzen zu lassen. Der Medienprofi nahm die Leute sofort in seinen Bann und hatte die Lacher auf seiner Seite, als er wissen wollte, wer unter den Anwesenden sich öffentlich zu Windows bekennen würde.

Als Ausblick in die nähere Zukunft stellte er die PDO's (Portable Distributed Objects) vor. Besonders deutlich strich er heraus, dass der Entwickler von morgen seine Gedanken voll auf die Software und nicht auf die Hardware richten soll. Es muss ihm so ziemlich egal sein auf welcher konkreten Maschine in welcher Programmiersprache er seine Objekte zusammenstellt – solange er sich an die Schnittstelle der PDO's hält, läuft sein Programm plattformübergreifend sowohl lokal als auch innerhalb eines heterogenen Netzwerkes. Daher sei, so Lipps, die Frage nach einer konkreten Prozessorunterstützung ziemlich müssig und soll in Zukunft nicht im Mittelpunkt der Entwicklungen von NeXT stehen.

## Apero

Beim anschliessenden Aperero hatte ich Gelegenheit in lockerer Atmosphäre mit den Vertretern von NeXT Deutschland zu sprechen. Nach Aussage von Rainer Bernert hat die Portierung von NEXTSTEP auf diverse Systeme oberste Priorität. Basierend auf der Version 3.2 soll bis Mitte Jahr die Portierung auf HP PA-Risc abgeschlossen sein. Bis Ende Jahr will man auch die Version für die Sparc-Rechner fertig haben. Gleichzeitig wird natürlich die Arbeit an OpenStep und den neuen PDO's vorangetrieben – die Version für HP-UX ist sogar schon erhältlich.

## Fazit

Alles in allem ein gelungener und interessanter Nachmittag. Einzig schade fand ich den schon erwähnten Zeitdruck. Die Atmosphäre war sehr familiär, was dadurch verstärkt wurde, dass sich die offiziellen Vertreter von NeXT nicht abschotteten, sondern für jeden ein offenes Ohr hatten.

**Dominik Moser, Redaktor**

## Kontaktadressen:

- *Uptime Object Factory Inc.*, Technopark Zürich, Telefon 01 445 16 99, Fax 01 445 16 98
- *Credal AG*, Silberstrasse 12, 8953 Dietikon, Telefon 01 741 11 15, Fax 01 741 17 40
- *ANDES Informatique SA*, Côtes de Montbenon 8, 1003 Lausanne, Telefon 021 311 57 57, Fax 021 312 93 57



## Das Internet – was ist das?

In der letzten Zeit wird in den Medien immer öfter von der Datenschnellstrasse (*Information Data Highway*) gesprochen. Was dies bedeutet ist einfach: nach all den Jahren haben auch die Politiker, Manager und Journalisten endlich das Internet entdeckt. Aber was ist nun dieses Internet? – Wenn Sie dieses nicht wissen ist dieser Artikel genau das Richtige für Sie.

### Was ist das Internet

Vor 18 Monaten habe ich meinen Rechner wie in früheren **PowerKey** Artikeln beschrieben mit Mail auf UUCP Basis und NiCEmail ausgestattet. In der Folge habe ich die Vorteile eines vernetzten Computer schätzen gelernt. Aber Mail ist nicht alles, was man mit einem Netz anstellen kann, es gibt noch viel mehr. Um dieses anzustellen reicht UUCP aber nicht mehr aus. Dafür wurde das Internet erfunden.

Als allererstes ist das Internet ein standardisiertes Verfahren (TCP/IP), direkte Verbindungen zwischen zwei Rechnern aufzubauen und darüber beliebige Datenströme in beiden Richtungen auszutauschen. Im Gegensatz dazu steht z.B. das *store-and-forward* Verfahren von UUCP, das nur den Versand von vor dem Versand bereits vollständigen Datenblöcken (z.B. Mails) erlaubt. Das Internet erlaubt gleichzeitig eine beliebige Anzahl von interaktiven Verbindungen von einem Rechner zu einem oder mehreren anderen Rechnern offen zu halten. Über jede dieser Verbindungen können in beliebiger Reihenfolge beliebige Daten verschickt werden. In zweiter Linie ist das Internet eine Sammlung von Anwendungen, die Daten über das Netz in standardisierten Formaten unter einander austauschen. Für Beispiele solcher Anwendungen siehe Abschnitt «*Was kann das Internet*».

Wichtig ist das Internet, weil es dank seiner Leistungsfähigkeit, Flexibilität und Günstigkeit viele Personen

und ihre Dienste angelockt hat. Diese sind der Hauptgrund, warum es sich lohnt, Mitglied beim Internet zu sein, genauso wie die vielen erreichbaren Personen der Grund sind, ein Telefon zu haben.

Ich bin davon überzeugt, dass ein Internet Anschluss in 10 Jahren genauso selbstverständlich sein wird, wie ein persönlicher Computer heute ist (war es vor nur 10 Jahren noch nicht).

### Wie gross ist das Internet

Je nach dem, wem man glaubt gibt es zwischen 10 und 20 Millionen Teilnehmer, zwischen 1 und 2 Millionen Maschinen und zwischen 10 und 20 Tausend Teilnetze. Die genauen Zahlen ist ohnehin nicht wichtig, da sie sich dauernd ändern. Genauer gesagt, sie verdoppeln sich ca. alle 8 Monate! Wichtig ist, dass diese alle durch ein einziges Standardverfahren direkt und transparent zusammenhängen, was zur Folge hat, dass man jeden Teilnehmer erreichen kann, ohne sich mit dem dazwischen liegenden Netz auseinandersetzen zu müssen, genau so wie es auch beim Telefon der Fall ist.

Kurz: Das Internet ist *das* Computernetz schlechthin.

### Was kann das Internet

#### Internet Mail

Im Gegensatz zum UUCP basierten Mail mit der NiCEmail Sortierzentrale mit ca. 80 Personen stehen einem auf dem Internet 10-20 Millionen Personen als Adressaten zur Verfügung, darunter fast alle Universitäten, alle wichtigen Unix Softwareschreiber, die Redaktionen der meisten Computerzeitschriften etc. Natürlich wohl wird niemand Mails an alle diese Personen schicken wollen, aber die grosse Anzahl macht es wahrscheinlicher, dass die Person, die man erreichen will, dabei ist (man telefoniert ja auch nicht mit allen Telefonbesitzern).

Weiterhin sind fast alle anderen Mail Systeme der Welt (z.B. CompuServe) mit dem Internet verbunden. Damit hat man auf einem Schlag mehr als 100 Millionen Adressaten zur Verfügung (allerdings mit zum Teil fürchterlichen Adressverrenkungen).



## NetNews

Im vorletzten **PowerKey** habe ich von Mailing Listen gesprochen. Noch besser als das ist NetNews (auch als Usenet bekannt). NetNews ist ein sog. *Bulletin Board System* (wie CompuServe) aber viel grösser und über Zehntausende von Rechnern verteilt (es wäre viel zu gross für einen einzelnen Zentralrechner).

NetNews hat mehrere Millionen Teilnehmer, die täglich ca. 150 MB an Meldungen auf ca. 5000 Newsgruppen (Foren, Anschlagbretter und elektronischen Zeitschriften) verschicken. Neben den von CompuServe bekannten «Frage-und-Antwort» (Q&A) Foren sind auch die vielen über NetNews verteilten elektronischen Zeitschriften zu erwähnen. Viele Wissenschaftler und Softwareentwickler benützen heute NetNews als ihr Hauptmedium um Forschungsergebnisse auszutauschen und zu veröffentlichen. Wer will kann bei den meisten «mithorchen» (das ist völlig legal und erwünscht). Dies ist viel aktueller als auf die papierne Fachpresse zu warten (die macht oft auch nichts anderes als hier mitzuhorchen und es Monate später zu drucken).

Es wird im Gegensatz zu dem Mailing Listen keine Anmeldung benötigt. Jeder holt sich den Inhalt der Newsgruppen, die ihn interessieren. Dazu einfach auf dem lokalen Rechner einstellen, was einem interessiert. Damit man nicht 5000 Gruppen angeben muss sind sie hierarchisch geordnet (wie Dateien in Verzeichnissen); man kann neben einzelnen Gruppen auch ganze «Verzeichnisse» bestellen oder nicht.

## FTP

Wer länger als einige Monate mit Computern gearbeitet hat weiss, dass man dazu Software braucht. Diese kann man sich entweder teurer bei kommerziellen Firmen kaufen oder man holt sie sich als Public Domain Software. Neben dem Vorteil, nichts zu kosten, ist meist auch der Sourcecode mit enthalten, was eigene Modifikationen erlaubt.

Auf dem Internet sind ca. 1000 grosse Softwarebibliotheken mit im Schnitt ca. 3000 MB an Dateien aufzufinden. Davon ist allerdings nur ein Teil Software, der grosse Teil besteht aus Dateien mit ungefähr jedem möglichen Inhalt, für jeden Geschmack und Bedarf etwas, auch z.B. Bilder im TIFF oder GIF Format.

Unsere Vereinsmaschine *nice* ist auch eine FTP Maschine. Wer Dateien von der *nice* will und nicht ans *talk&copy* kann, kann sich diese einfach über's Internet holen, solange sie nicht zu gross sind (>1 MB, nur die Telefonkosten begrenzen dies)<sup>1</sup>.

## Telnet

Immer der selbige Rechner zuhause – wie wär's mit einem Anderen? Nein allererstens, es gibt auf dem Internet Tausende von Rechnern mit ungewöhnlichen Programmen und Fähigkeiten (die Dienste von 5 Supercomputern anderen Universitäten zur Verfügung zu stellen war die allererste Aufgabe des Internets). Auf einigen von ihnen kann man sich einloggen um ihre Programme zu verwenden. Sehr interessant sind zum Beispiel Bibliothekskataloge, wie das *ETHICS* der ETH Bibliothek. Wer ein Buch sucht kann einfach in über einer Million Titeln, Autoren und Themen suchen, bis er das richtige findet. Wer bei der ETH eingetragen ist kann sogar von zuhause aus sich das Buch bestellen/reservieren und per Post zuschicken lassen.

## Archie

Dies ist wohl das interessanteste der via Telnet erreichbaren Programme. Wer vor den vielen FTP Servern steht und den Wald vor lauter Bäumen nicht mehr sieht, findet mit diesem Programm die berühmte Nadel im Heuhaufen, will heissen die gesuchte Datei.

Einfach in einen der Archie Rechnern einloggen (kein Passwort) und nach Dateinamen oder Thema fragen. Das ergibt einem oft eine lange Liste von FTP Maschinen inklusive Verzeichnis und Dateinamen.

## WWW (World Wide Web)

Dies ist das Neueste vom Neuen. Ein wie FTP und NetNews über beliebig viele Maschinen verteiltes interaktives multimediafähiges Hypertextsystem (bei den Adjektiven noch mitgekommen?). Ist eine supereinfache Methode (einfach anklicken), um Informationen aller Arten (Text, Sound, Bilder, Video) zu finden und zu sich nach hause zu holen.

<sup>1</sup> *Anm. der Redaktion:* In einer der nächsten Ausgaben werden wir einen Artikel über den ftp-Zugang an den NiCE-Server veröffentlichen.



WWW wird von vielen Personen als die Killer-App (ein Programm, das ein System zum Muss macht) des Internet bezeichnet. Es setzt aber eine während des Gebrauchs dauernd offene Leitung voraus, was mit einem Modemanschluss ans Internet ziemlich teuer werden kann, wenn man davon angefressen wird (PTT lässt grüssen).

## Chat

Mehrkanaliges «Tipp-CB-Funk-System» mit anderen ans Internet angeschlossenen Personen. Macht extrem schnell süchtig (Aussage unseres Vizepräsidenten). Noch teurer, wenn das passiert.

## Verhältnis SLIP zu UUCP und NiCEmail

Wer bereits ans NiCEmail angeschlossen ist wird sich jetzt wohl Gedanken über die Verträglichkeit der beiden Systeme machen, insbesondere über die Teilnahme an zwei verschiedenen Mailsystemen (2 Modem-Verbindungen).

Glücklicherweise ist dies gar kein Problem. Das NiCEmail ist schon seit jeher mit dem Internet Mail voll kompatibel und verbunden. Nur etwa 20 der 80 NiCEmail Adressen sind UUCP Anschlüsse, der Rest sind Internet Mail Anschlüsse (die meisten in Firmen oder an der ETH oder Uni). Einige wenige darunter sind aber bereits «Heim-» Internetadressen, wie z.B. meine.

Das NiCEmail wird einfach so eingestellt, das die Mails ins Internet Mail weitergeleitet werden. Sie werden dann mit den anderen Internet Mails heruntergeladen. UUCP ist nicht mehr nötig. Es wird ebenfalls das bereits bekannte NeXTmail Programm benutzt. Selbst die Adressen bleiben gleich. Es muss lediglich der Eintrag in die NiCEmail Steuerdatei vom Mailadmin abgeändert werden.<sup>2</sup>

## Wie kommt man von zuhause ans Internet

Das Internet wird von den meisten Personen hardwaremässig mit Ethernet und Standleitungen in Verbindung gebracht. Dies liegt daran, das es an Unis und in Grossfirmen zumeist so installiert ist. Tatsächlich aber läuft TCP/IP über jede erdenkliche Leitungsform, die beliebige Blöcke von Bytes übermitteln kann. Da man selten eine Ethernet-Leitung nach Hause legen lassen kann und Standleitungen teuer sind, greift man als Privatperson zum (hoffentlich vom UUCP her bereits vorhandenen) RS232 Modem mit mindestens 9600 Baud und einer Telefonleitung. Diese Art von Internetanschluss ist als *Serial Line Internet Protocol* (SLIP) bekannt.

Die NiCE hat mit der EUnet (einem Anschlusslieferanten) eine Sonderaktion für solche SLIP Anschlüsse zu einem von Privatpersonen bezahlbaren Preis ausgehandelt. Jedes Mitglied kann einen bekommen. Diese Anschlüsse dürfen aber nur privat benutzt werden, wer seine Firma anschliessen will muss einen SLIP Anschluss zum Normaltarif beziehen.

Der Preis beträgt Fr.30.-/Monat sofern man max. drei Std/Monat die Verbindung nutzt. Jede Zusatzstunde kostet Fr.15.- (keine Voranmeldung, wird einfach mit verrechnet). Reger Mailverkehr (ich bin immerhin Präsident) verbraucht lediglich etwa 1 Std/Monat davon. Mit 2 Std/Monat für NetNews und FTP kommt man mit dem Auswerten der Daten gerade noch nach.

## Alle Klarheiten beseitigt

Wer noch Fragen zum Internet hat kann mir diese selbstverständlich über NiCEmail zukommen lassen.

**Neil Franklin**  
Neil.Franklin@nice.ch

<sup>2</sup> *Anm. der Redaktion:* In der nächsten Ausgabe erscheint zu diesem Thema der Artikel «Verbindung mit dem Internet via SLIP und EUnet» von Neil Franklin.



## NEXTSTEP – Grundlage des OpenStep Standards

Die Kooperation von NeXT und SunSoft, mit dem Ziel der Festlegung eines offenen Standards für objektorientierte Entwicklungs- und Betriebssysteme auf der Basis von NEXTSTEP, wird die Verbreitung objektorientierter Technologie deutlich beschleunigen. Dieser Artikel vermittelt einen Überblick über die Kooperationsvereinbarungen und beschreibt die Architektur von NEXTSTEP – der Grundlage der OpenStep Spezifikation.

### 1 OpenStep – Das offene NEXTSTEP am Beispiel OpenStep/Solaris von SunSoft

Seit der erstmaligen öffentlichen Vorstellung im Jahr 1989 ist NEXTSTEP das einzige kommerziell verfügbare durchgängig objektorientierte Entwicklungs- und Betriebssystem. Es liegt 8 Jahre nach Entwicklungsbeginn inzwischen im dritten Release vor. Weltweit wurden bisher mehr als 75.000 Binärlizenzen vergeben. Das aktuelle Release NEXTSTEP 3.2 gilt als ausgesprochen stabil und wird in verschiedensten Industriebereichen, von der Fertigungsindustrie bis hin zum Finanzdienstleistungssektor, insbesondere als Entwicklungsplattform für massgeschneiderte Individualsoftware, sog. «*Mission Critical Custom Applications*», eingesetzt. Darüber hinaus erfreut sich NEXTSTEP in Forschung und Lehre zunehmender Beliebtheit.

NEXTSTEP wird von anerkannten Analysten ein Entwicklungsvorsprung von mindestens zwei Jahren gegenüber vergleichbaren Ansätzen anderer Anbieter bescheinigt, die bisher weitgehend nur auf dem Papier existieren.

Die führende Rolle von NeXT auf dem Gebiet der objektorientierten Systeme und der deutliche Technologievorsprung von NEXTSTEP haben SunSoft als den

weltweit führenden UNIX-Anbieter nun bewegen, die NEXTSTEP Technologie in zukünftige Versionen des Betriebssystems Solaris zu integrieren.

Im einzelnen sehen die Abkommen mit SunSoft und anderen Tochtergesellschaften von Sun Microsystems folgendes vor:

- NeXT wird die Programmierschnittstelle seines objektorientierten Entwicklungs- und Betriebssystems NEXTSTEP offenlegen und diese in der OpenStep Spezifikation bis zum 30. Juni 1994 veröffentlichen. OpenStep und das OpenStep Warenzeichen sind frei lizenzierbar für alle unabhängigen Implementierungen, die die Kompatibilitätsprüfung durch die OpenStep Testsuite bestehen.

Dieses Vorgehen ist vergleichbar mit der Offenlegung des PostScript Standards für Seitenbeschreibungssprachen durch Adobe und der Veröffentlichung der Spezifikation des *Network File System* (NFS) durch Sun.

- NeXT und SunSoft werden gemeinsam mit anderen Mitgliedern der *Object Management Group* (OMG) die OpenStep Spezifikation den unabhängigen Standardorganisationen OMG und dem X/Open-Konsortium zur Prüfung und zur Übernahme als Standard für objektorientierte Systeme vorlegen.
- SunSoft lizenziert die NEXTSTEP Anwendungs- und Entwicklungsumgebung und wird eine OpenStep kompatible Implementierung in Solaris integrieren und die Programmiersprache Objective-C unterstützen.
- NeXT wird in Zusammenarbeit mit dem SPARC Technology Business Geschäftsbereich von Sun Microsystems die Portierung von NEXTSTEP auf SPARC-Prozessoren durchführen.
- NeXT erwirbt verschiedene Lizenzen von SunSoft, u.a. CORBA (*Common Object Request Broker Architecture*), die OMG IDL (*Interface Definition Language*) und das NFS (*Network File System*).



Hewlett-Packard (HP), der weltweit umsatzstärkste Hersteller von UNIX-Systemen und Gründungsmitglied der Standardisierungsorganisationen *Open Software Foundation* (OSF) und *OMG*, der mit NeXT seit geraumer Zeit in der *Object-Enterprise* Kooperation eng zusammenarbeitet, hat NeXT bereits seine volle Unterstützung bei der Standardisierung der *OpenStep* Spezifikation durch die *OMG* und *X/Open* zugesagt.

Das nachfolgende Kapitel beschreibt die verschiedenen Ebenen von *NEXTSTEP* und dient somit als Beschreibungsgrundlage für *OpenStep*.

## 2 *NEXTSTEP* 3.2 – Die Basis von *OpenStep*

Neben der durchgängigen Objektorientierung, die sowohl dem einfachen Anwender als auch dem erfahrenen Softwareentwickler zu Gute kommt, bietet *NEXTSTEP* die Einfachheit der Bedienung eines *Macintosh* auf der Basis des leistungsfähigen *UNIX* Betriebssystems. Durch die Verwendung von *Display PostScript Level II* als grundlegendes Beschreibungsmodell des Fenstersystems ist wahres *WYSIWYG* («*what-you-see-is-what-you-get*») garantiert, d.h. die Übereinstimmung der Darstellung auf dem Bildschirm mit dem Ergebnis auf einem Drucker oder einem anderen Ausgabemedium.

Die Diskussion der Vorteile der objektorientierten Technologie wird häufig anhand der nachweisbaren deutlichen Entwicklungsvorteile geführt, zu denen kürzere Entwicklungszeiten, höhere Qualität, bessere Wartbarkeit und Wiederverwendbarkeit zählen.

In diesem Artikel sollen die Anwendervorteile im Mittelpunkt stehen, die sich aus dem Zusammenwirken verschiedener Anwendungen («*Inter-Application Communication*») ergeben, die nur auf der Basis einer vollständig objektorientierten Anwendungsumgebung möglich ist.

### 2.1 Architektur

#### Hardware

*NEXTSTEP* kann plattformübergreifend eingesetzt werden: Seit Mai 1993 ist es auf dem *PC* Industrie-

standard *Intel 486* und *Pentium* ebenso verfügbar wie auf der *Motorola 68040* basierten *NeXT* Hardware, auf der *NEXTSTEP* ursprünglich entwickelt wurde. Ab dem Frühjahr 1994 wird *NEXTSTEP* auf *PA-RISC* Rechnern von *HP* im Rahmen der *Object•Enterprise* Kooperation zwischen *NeXT* und *Hewlett-Packard* zur Verfügung stehen. Die Verfügbarkeit der *SPARC*-Portierung ist für die zweite Jahreshälfte 1994 angekündigt.

Im Gegensatz zu anderen Betriebssystemen spielt hierbei die Byteanordnung («*endianess*») der verschiedenen Prozessoren keine Rolle. Evtl. notwendige *Bytetauschoperationen*, z.B. bei der Verarbeitung von auf *Motorola* erstellten *Binärdaten* auf einem *Intel*-Rechner, werden automatisch und für den Anwender transparent durchgeführt. Der auftretende *Performanceverlust* ist absolut zu vernachlässigen.

Unter *NEXTSTEP* entwickelte Programme sind dabei völlig plattformunabhängig und vollständig *Quellcode* kompatibel, d.h. lediglich eine *Neuübersetzung* eines auf *Motorola* entwickelten Programms genügt, um es auf einem *Intel*-Rechner ablaufen zu lassen. *Quellcodeveränderungen* sind dazu nicht nötig.

Die *NEXTSTEP* Entwicklungsumgebung umfasst zudem standardmässig *Crosscompiler* für alle unterstützten Plattformen, so dass in einem Arbeitsgang sog. «*Multi-Architecture-Binaries*» (*MABs* oder auch «*fat binaries*») erzeugt werden können, die den ausführbaren Code für mehrere Plattformen erhalten. Für den Anwender genügt dann ein *Doppelklick*, um die jeweilige Anwendung zu starten, ganz egal, welcher Prozessortyp im Rechner des Anwenders steckt.

#### *MACH* und *UNIX 4.3 BSD*

Die Auswahl des der *NEXTSTEP* Anwendungs- und Entwicklungsumgebung zugrundeliegenden Betriebssystems wurde durch die folgenden Anforderungen bestimmt: *Multiuser* und *Multitasking* Fähigkeit, *Netzwerk*kompatibilität, performante *Interprozesskommunikation*, sowie *Robustheit* und *Erweiterbarkeit*.



NeXT hat sich daher bereits 1988 für das MACH Betriebssystem, das an der Carnegie Mellon Universität (CMU) entwickelt wird, entschieden. MACH basiert auf einem kleinen, auf effiziente Interprozesskommunikation ausgerichteten Kern, der bereits im Design auf die Unterstützung verteilter und paralleler Programmierung angelegt ist.

Die von NeXT verwendete Version setzt auf einer Portierung der CMU Version 2.0 auf, erweitert um einige Features der Version 2.5 sowie NeXT spezifischen Erweiterungen, wie z.B. dynamisch ladbaren Treibern, sog. «*Bootstrap Servers*» und verbessertem Schedulingalgorithmus.

Durch die vollständige Kompatibilität von MACH zu UNIX 4.3 BSD, sowohl auf der Ebene der Systemaufrufe als auch der UNIX Kommandos, ist die problemlose Integration in bestehende Netzwerke sichergestellt. Das von NEXTSTEP verwendete UNIX-Dateisystem ist ebenfalls 4.3 BSD kompatibel. Weitere Dateisysteme für DOS, das Macintosh Hierarchical File System (HFS), die CD-ROM-Formate Rockridge und ISO 9660 («*High Sierra*») sind transparent integriert.

Das NEXTSTEP Betriebssystem erfüllt die Anforderungen des IEEE POSIX 1003.1 Standards und ist auf den C2 Sicherheitsstandard gemäss «*Orange Book*» vorbereitet.

Über die aus 4.3 BSD bekannten Leistungsmerkmale, wie u.a. Netzwerkdienste hinaus, bietet MACH die folgenden Features: mehrere Prozesse (sog. «*tasks*») in jeweils einem eigenen virtuellen Adressraum, mehrere leichtgewichtige Prozesse (sog. «*threads*») innerhalb einer Task, flexibles Scheduling, gemeinsamer Speicher («*shared memory*») für verschiedene Tasks, effiziente Interprozesskommunikation über Nachrichtenaustausch, «*memory-mapped files*», Unterstützung für flexible Sicherheitskonzepte und Multiprozessorarchitekturen.

MACH wird häufig bereits als objektorientiertes Betriebssystem bezeichnet, da die meisten Systemdienste von Programmen durch den Versand von MACH Nachrichten aktiviert werden. Es ist allerdings zu beachten,

dass es sich bei MACH Objekten/Nachrichten und Objective-C Objekten/Nachrichten, wie sie bei der Implementierung der verschiedenen NEXTSTEP Toolkits verwendet werden, um verschiedene Konzepte handelt, die allerdings aufeinander aufbauen. D.h. aus einem Toolkit können direkt Nachrichten an MACH Objekte versendet bzw. direkt UNIX Systemaufrufe abgesetzt werden.

## Netzwerkdienste und NetInfo

Zusätzlich zu dem im UNIX-Bereich vorherrschenden Internetprotokoll TCP/IP und UDP sowie den darauf aufsetzenden Diensten und Protokollen, wie FTP, Telnet, den Berkeley Remote-Diensten (rsh, rlogin, rcp, usw.), NFS, u.a. wie SNMP, werden die Novell-Netware Protokolle IPX und SPX unterstützt, so dass jeder NEXTSTEP-Rechner Zugriff auf Novell-Datei- und Druckerserver hat.

Neben den aus anderen UNIX Systemen bekannten Netzwerkdiensten beinhaltet NEXTSTEP das NetInfo Netzwerkadministrations- und Konfigurationssystem, das einerseits vollständig kompatibel zu DNS (*Internet Domain Name Service*) und NIS (*Network Information System*, früher «*yellow pages*» genannt) ist, andererseits aber weit leistungsfähiger ist und die netzwerkweite Verwaltung von Ressourcen, wie Benutzerkonten, Rechnernamen, Drucker und Faxmodems, Dateisystemen usw. in beliebig tiefen Domain-Hierarchien drastisch vereinfacht. Netzwerkweites Drucken sowie netzwerkweiter Versand und Empfang von Fax sind dadurch ein Kinderspiel.

NetInfo kann im Quellcode lizenziert werden und ist als Binärlizenz für UNIX System V.4, Solaris, HP/UX, AIX, OSF/1 und andere Systeme verfügbar.

In Zusammenarbeit mit Hewlett-Packard ist eine Integration von NetInfo in das *Distributed Management Environment* (DME) der OSF geplant.

## Window Server

Durch die Verwendung des multitaskingfähigen MACH Betriebssystemkerns können mehrere Anwendungen quasiparallel ablaufen. Zu jeder Anwendung wiederum



gehören ein oder mehrere Fenster, deren Inhalt angezeigt wird und für die Ereignisse, wie z.B. ein Mausklick, an die Anwendung gemeldet werden muss.

Diese beiden Aufgaben und die Koordination der verschiedenen Fenster unterschiedlicher Anwendungen übernimmt der Window Server als eigenständiger Prozess. Jede NEXTSTEP-Anwendung ist ein Klient des Window Servers und interagiert mit diesem über verschiedene Objekte des Application Kit (siehe «*Entwicklungsumgebung*»). Klient und Server können dabei durchaus auf verschiedenen Rechnern in einem Netzwerk ablaufen.

Der Window Server akzeptiert Zeichenbefehle im Display PostScript Format (im wesentlichen für 2D Graphik) und im RenderMan Interface Bytestream Code (für 3D Graphik). Die NEXTSTEP-Lizenz beinhaltet daher sowohl eine Lizenz für Display PostScript Level II von Adobe als auch für Interactive RenderMan von Pixar.

### Workspace Manager

Über den Workspace Manager – vergleichbar mit einem sog. *Desktop Manager* im X/Motif-Umfeld – greift der Anwender auf das Dateisystem und die darin enthaltenen Dateien, Ordner und Programme zu.

Detaillierte Informationen zu den verschiedenen Dateien, z.B. Dateigröße und -zugriffsrechte, die verschiedenen Werkzeuge, die eine Datei (genauer das Dateiformat dieser Datei) bearbeiten können, sowie eine Inhaltsanzeige werden in einem Inspektor dargestellt. Für die gängigen Dateiformate (siehe auch den Abschnitt «*Inter-Application Communication*»), wie Microsoft RTF und TIFF sowie Adobe PostScript, gibt es vordefinierte Inspektoren. Inspektoren für andere Dateiformate können – für den Anwender völlig transparent – hinzugefügt werden.

Programme werden durch Doppelklick auf das zugehörige Programmsymbol bzw. durch Doppelklick auf das Dateisymbol eines assoziierten Dateiformats gestartet.

### Entwicklungsumgebung

Die Komponenten der Entwicklungsumgebung (Toolkits und Werkzeuge) werden im Kapitel «*NEXTSTEP Developer*» erläutert.

### Anwendungsprogramme

Der aktuelle Softwarekatalog [1] listet weit mehr als 500 kommerziell verfügbare NEXTSTEP Anwendungen vom Tabellenkalkulationsprogramm über Textverarbeitung und DTP-Programmen bis hin zum Datenbankadministrationswerkzeug auf. Darüber hinaus sind eine Vielzahl von Programmen frei im Internet verfügbar.

Mit Hilfe von Insignia Solutions' SoftPC sind nahezu alle MS-DOS und MS-Windows-Programme unter NEXTSTEP ablauffähig, wobei im Falle von NEXTSTEP/Intel keine Emulation der CPU notwendig, also der volle Zugriff auf den Intel 80486 Prozessor gegeben ist. Zudem gibt es die gängigen Terminalemulationen, von VT220 und VT320 über X Window System (X11 R5) inkl. Motif bis hin zur Mainframewelt mit 3270 und 5250.

Neben den im SW-Katalog aufgeführten sog. «*shrink-wrapped*» Programmen gibt es mehr als 100 Klassenbibliotheken, sog. Toolkits, deren Einsatzgebiet von der einfachen Texteingabeprüfung bis zur Implementierung komplexer Finanzmodelle reicht. NeXT war der erste Softwarehersteller, der einen eigenen Katalog [2] für solche Softwarebausteine, sog. *ObjectWare*, herausgebracht hat.

Zunehmend setzt sich dabei der Trend durch, dass Teilfunktionalitäten eines «*shrink-wrapped*» Programms in Form einer Bausteinsammlung für Entwickler zugänglich gemacht werden. Ein Entwickler kann dadurch z.B. diese Bausteine für den Anwender völlig transparent in seine eigene Anwendung integrieren oder eine Anwendung von für den Anwender in seinem Anwendungskontext unnutzer Funktionalität befreien (welcher Anwender nutzt schon die Gesamtfunktionalität der heute verfügbaren Textverarbeitungsprogramme).



Allen Anwendungen und Softwarebausteinen ist gemeinsam, dass sie nahezu beliebig miteinander kombiniert werden können. Die verschiedenen Kombinationsmechanismen sind im nachfolgenden Kapitel unter der Überschrift «*Inter-Application Communication*» zusammengefasst.

## 2.2 OpenStep Architektur

Im Prinzip handelt es sich bei OpenStep um die Funktionalität von NEXTSTEP ohne den spezifischen Anteil des Kernbetriebssystems MACH/UNIX 4.3 BSD. Anstelle dieses Kernbetriebssystems kann nahezu jedes hinreichend mächtige Betriebssystem als Trägersystem für OpenStep Verwendung finden. SunSoft wird seine OpenStep Implementierung auf Solaris aufsetzen, damit OpenStep auf der Basis eines COSE kompatiblen UNIX realisieren und gleichzeitig die Integration von OpenStep Anwendungen mit Anwendungen des Common Desktop Environment gewährleisten.

Ein Programm, das nur die OpenStep Schnittstelle nutzt, ist somit zwischen verschiedenen OpenStep und NEXTSTEP Implementierungen ohne Quellcodeänderung portierbar.

## 3 Inter-Application Communication

### 3.1 Das Konzept

Aus der Sicht des Anwenders besteht die wesentliche Stärke des durchgängig objektorientierten NEXTSTEP Systems darin, dass Anwendungen der verschiedensten Hersteller miteinander nahezu beliebig kombiniert werden können. Dabei kommunizieren i.d.R. kleine, überschaubare Programme miteinander, deren Funktionalität für den Anwender nachvollziehbar ist.

An die Stelle undurchschaubarer, monolithischer Gesamtpakete (oder «*eierlegender Wollmilchsäue*»), die vorgeben, Alleskönner zu sein, treten hier Geflechte von Anwendungen, die jede für sich eine Teilfunktionalität sehr gut implementieren und zusammen die gewünschte Gesamtfunktionalität erbringen. Hierbei gilt sehr oft: «*Das Ganze ist mehr als die Summe der Teile*».

Man beachte, dass ein objektorientiertes System an dieser Stelle sehr viel weitergehende Unterstützung bietet (die Objektdefinition ist bereits Protokolldefinition), als ein reines Fenstersystem. (Als Negativbeispiel sei hier X/Motif angeführt, das über den primitiven und obendrein unzulänglichen Datenaustausch über Zwischenpuffer per Definition keine weiteren Mechanismen anbietet und somit in seinem Leistungsangebot selbst hinter MS-Windows weit zurückfällt).

Die Mechanismen, die der Kommunikation der verschiedenen Programme zugrundeliegen, werden nachfolgend beschrieben.

### 3.2 Die Mechanismen

#### Durchgängig unterstützte Datenformate

Für den nahtlosen Datenaustausch zwischen unterschiedlichen Anwendungen ist die durchgängige Unterstützung hinreichend mächtiger Datenformate durch das zugrundeliegende Betriebssystem die wichtigste Grundvoraussetzung.

NEXTSTEP unterstützt daher u.a. Adobe PostScript Level II und *Encapsulated PostScript* (EPS), das *Rich Text Format* (RTF) von Microsoft und das *Tagged Image File Format* (TIFF) für Bitmapdaten von Altsys, *RenderMan Interface Bytestream Code* (RIB) von Pixar und Faxdaten gemäss Gruppe 3 und 4 CCITT Definition.

Weitere Formate, wie etwa die Bitmapformate PICT, GIF oder PCX bis hin zu beliebigen applikationsspezifischen Datenformaten, können völlig transparent integriert werden (siehe «*Filter*»).

Die im folgenden beschriebenen Datenaustauschmechanismen sind im Application Kit implementiert.

#### Cut & Paste Zwischenpuffer

Zwischen NEXTSTEP-Anwendungen können beliebig strukturierte Daten formaterhaltend über getypte Zwischenpuffer ausgetauscht werden.



Zu diesem Zweck haben alle Anwendungsprogramme Zugang zu einem gemeinsamen zentralen Pasteboard-Server. Anwendungen nutzen dabei verschiedene Zwischenpuffer, abhängig von den zu übertragenden Datentypen. So gibt es allgemein vordefinierte Puffer für Text (ASCII und RTF), Zeichensätze, Bilder (TIFF und EPS) und Absatzformate inkl. Tabulatoren, sowie unbeschränkt viele weitere Puffer für anwendungsspezifische Datentypen.

Jeder Zwischenpuffer wird durch ein Pasteboard-Objekt repräsentiert, wobei ein solches Objekt mehrere verschiedene Repräsentationen des gleichen Datums speichern kann, z.B. sowohl eine Bitmap- als auch eine PostScript-Darstellung eines Bildes oder eine RTF- und eine ASCII-Repräsentation des gleichen Textes.

Legt eine Anwendung verschiedene Repräsentationen im Puffer ab, kann die Anwendung, die den Pufferinhalt anfordert, zwischen diesen Repräsentationen wählen. In der Regel wird die «beste» Darstellung angefordert, also z.B. RTF- statt reinem ASCII-Text.

Übrigens werden nicht alle möglichen Darstellungen gleichzeitig im Puffer abgespeichert, sondern im Zuge einer sog. «*lazy buffer evaluation*»-Strategie wird die geeignete Repräsentation erst dann berechnet, wenn die anfordernde Anwendung eine konkrete Darstellung benötigt. D.h., die Kommunikation zwischen den am Pufferaustausch beteiligten Anwendungen ist i.a. vollständig bidirektional.

## Object Linking

Object Links bilden die Grundlage des aktiven Dokumentaustauschs («*inter-document communication*») in NEXTSTEP.

Object Links erweitern den o.a. Pasteboard-Mechanismus, so dass Anwendungen sich dynamisch verändernde Daten gemeinsam teilen können, was u.a. zur Erstellung sog. «lebender» Dokumente verwendet werden kann, z.B. ein Textdokument, das eine Tabelle enthält, deren Werte aus einer Datenbank abgefragt werden, deren Inhalt sich im Verlauf der Zeit verändert.

Jedem Dokument wird dabei ein LinkManager-Objekt assoziiert. Wenn nun der Anwender ein Datum aus diesem Dokument ausschneidet bzw. kopiert, legt der LinkManager nicht das Datum selbst, sondern einen Verweis auf das Ausgangsdokument und die zugehörige Anwendung im Zwischenpuffer ab. Fügt der Anwender anschliessend den Zwischenpufferinhalt in ein anderes Dokument ein und erstellt einen Verweis auf das Ausgangsdokument, wird nicht nur das Datum selbst eingefügt, sondern auch die ursprüngliche Selektion und das Ausgangsdokument vermerkt. Zudem kann ein Object Link im Dateisystem abgelegt werden und per Drag & Drop in andere Dokumente übernommen werden.

Über einen Verweisinspektor kann der Anwender bestimmen, ob und wann – manuell, ständig, oder beim Speichern des Ausgangsdokuments – das Zieldokument aktualisiert werden soll. Zudem kann er über diesen Inspektor das Ausgangsdokument öffnen oder den Verweis aufheben.

Andere Fenstersysteme wie Microsoft Windows und Apple Macintosh System 7 – nicht aber das in der UNIX-Welt verbreitete X Window System – unterstützen mit OLE 2.0 bzw. Publish/Subscribe einen aus Anwendersicht sehr ähnlichen Mechanismus. Während aber ein Entwickler in diesen Systemen erheblichen Programmieraufwand zur Unterstützung dieses Mechanismus einplanen sollte, genügt für eine NEXTSTEP-Anwendung die Definition und Registrierung des LinkManager-Objekts. Im Falle des im Quellcode mitgelieferten Zeichenprogramms Draw bedeutet dies z.B. im wesentlichen nur acht neue Programmzeilen gegenüber der Version, die mit herkömmlichen Cut & Paste arbeitet.

## Drag & Drop

Will der Anwender den Vorgang des Datenaustauschs beschleunigen, so kann er i.a. die explizite Verwendung von Zwischenpuffern umgehen und das Datum in Form einer symbolischen Darstellung direkt von einer Anwendung in eine andere ziehen.

Kann die empfangende Anwendung mit dem übertragenen Datenformat nicht umgehen, z.B. wenn eine GIF-Datei in Edit gezogen wird, so können spezielle



Filterdienste automatisch – für den Anwender unmerklich – eine Wandlung des Datenformats vornehmen, im vorliegenden Beispiel von GIF nach TIFF.

## Dienste

Als eines der herausragenden Konzepte der UNIX-Shell wird gerne der Pipelinemechanismus zitiert, der es ermöglicht, verschiedene UNIX-Kommandos hintereinanderzuschalten, die nacheinander einen Datenstrom verarbeiten.

Grob gesprochen, sind Dienste ein Pipelinemechanismus auf der Ebene der NEXTSTEP-Benutzerschnittstelle: Bei der jeweils aktiven Anwendung können sich andere Anwendungen anmelden und sog. *Dienste* anbieten. Der Anwender kann diese Dienste über das kontext-sensitive Dienstemenü der aktiven Anwendung auswählen.

Dabei muss die aktive Anwendung die möglichen Dienstanbieter vorab nicht kennen. Die Auswahl der Dienstanbieter wird dynamisch getroffen: die aktive Anwendung identifiziert den selektierten Datentyp und alle Anwendungen, die mit diesem Datentyp umgehen können, melden sich dann als potentielle Dienstanbieter an.

Der Datenaustausch selbst erfolgt über getypte Zwischenpuffer, wie sie auch für Cut & Paste verwendet werden.

Im Gegensatz zum unidirektionalen UNIX-Pipelinemechanismus ist der NEXTSTEP-Dienst-Mechanismus bidirektional, d.h. die aktive Anwendung kann nicht nur Daten an den Dienstanbieter übergeben, sondern auch Daten erhalten. Dienste werden mit Hilfe der Speaker- und Listener-Klassen des Application Kit realisiert.

## Filter

Ein Filter ist ein spezieller Dienst zur automatischen Datenkonversion. Filter werden u.a. im Verlauf von Drag & Drop Operationen implizit und von diversen Anwendungen explizit, z.B. von Digital Librarian (genauer dem Indexing Kit) bei der Anzeige von UNIX-Handbuchseiten und der damit verbundenen Konver-

tionierung von Dateien im TROFF- in das RTF-Format, aktiviert.

## Distributed Objects (DO)

Distributed Objects stellen einen Mechanismus für transparente, netzwerkweite Interprozesskommunikation dar. Anwendungen kommunizieren miteinander, indem sie ihre Objective-C Objekte bekannt machen und diesen Nachrichten senden. Dabei macht es keinen Unterschied, ob diese Objekte im Adressraum der gleichen Anwendung oder einer anderen Anwendung existieren. Die andere Anwendung kann sogar auf einem völlig anderen Rechner laufen, der im Netzwerk erreichbar ist.

Dieser Mechanismus ist wesentlich allgemeiner als herkömmlich verwendete *Remote Procedure Calls* – auch allgemeiner als die Verwendung von *Speaker/Listener* zur Implementierung von Diensten – und vereinfacht die Entwicklung von Client/Server-Anwendungen und kooperativen Anwendungen drastisch.

## Portable Distributed Objects (PDO)

Die Portable Distributed Objects (PDO) erweitern die NEXTSTEP Laufzeit- und Softwareentwicklungsumgebung um «fremde» Betriebssysteme, wie z.B. HP-UX 9.0, Data General UNIX V.4 und Solaris 2.3, indem sie den Zugriff und den Ablauf von Distributed Objects auf diesen Systemen erlauben. Sie bilden somit die Grundlage für die Wiederverwendung von Softwarebausteinen in einer heterogenen Netzwerklandschaft. Unternehmensweites verteiltes Computing vom Desktop-Rechner unter NEXTSTEP bis hin zum Hochleistungsrechner unter UNIX werden damit Wirklichkeit.

PDO ist voll kompatibel zu den Distributed Objects und umfasst die folgenden Komponenten: Objective-C++ Compiler mit NEXTSTEP Laufzeitsystem, Debugger, Basisobjektklassen (z.B. Object, List, HashTable, ...), die Distributed Object Klassen, Mach IPC Emulation Library, Portable nmserver, Portable BuildServer.

Die Programmentwicklung kann vollständig auf dem Frontend-Rechner unter NEXTSTEP erfolgen, wobei das NEXTSTEP-Entwicklungswerkzeug ProjectBuilder über den DO Mechanismus mit dem Objective-C++



Compiler auf dem Backend-Rechner kommuniziert.

Die jetzige Implementierung der PDO basiert auf einer Emulation der MACH Interprozesskommunikation über TCP/IP. Zukünftige Implementierungen werden auf dem Distributed Computing Environment (DCE) der OSF aufsetzen.

## **4 Entwicklungsumgebung NEXTSTEP Developer**

### **4.1 Überblick**

Eine objektorientierte Entwicklungsumgebung umfasst die folgenden Komponenten: Eine erweiterbare, objektorientierte Programmiersprache, effiziente Interprozesskommunikation, eine hinreichend grosse Menge von Basisobjekten, sowie Werkzeuge zur Verwaltung und Verknüpfung aller Arten von Objekten, sowohl von Basisobjekten, wie auch von applikationsspezifischen, eigengefertigten Objekten.

Der vorliegende Artikel kann die einzelnen Bestandteile der NEXTSTEP Entwicklungsumgebung nur überblicksartig darstellen. Auf eine Darstellung der objektorientierten Programmierung muss an dieser Stelle verzichtet werden, da sie den Umfang dieses Artikels bei weitem sprengen würde. Ein sehr gute Darstellung der grundlegenden Konzepte und der mit der Objektorientierung verbundenen Entwicklungsvorteile findet sich in [3]. Für eine detaillierte Beschreibung der gesamten Entwicklungsumgebung, ebenso wie für eine Einführung in die objektorientierte Programmierung mit Objective-C sei auf [4] und insbesondere auf die NEXTSTEP Developers Library [5] verwiesen.

Der NEXTSTEP Vorteil [3] lässt sich wie folgt zusammenfassen:

- Durchgehend objektorientierte Architektur – vom Betriebssystem über die Entwicklerwerkzeuge bis hin zur graphischen Benutzerschnittstelle.
- Kleine Menge von Basisobjekten, die die Grundlage jeder Anwendung bilden und die Interaktion unter-

schiedlichster Anwendungen sicherstellen.

- Reichhaltige Menge von Supportobjekten, für z.B. Ausdruck, Texteditierung, Rechtschreibprüfung, On-Line Hilfe, Dateiverwaltung, Farb- und Zeichensatzauswahl u.v.a.m., die die konsistente Bedienbarkeit von verschiedenen Anwendungen garantieren.
- Spezielle Toolkits (siehe unten) für z.B. Datenbankzugriff, Textindizierung, 3D-Graphik, Ton und Musik u.v.a.m.
- Objektorientierte, graphische Entwicklungsumgebung, die die Entwicklung graphischer Benutzerschnittstellen und anderer Programmkomponenten gegenüber herkömmlichen UNIX basierten Werkzeugen deutlich – bis zum Faktor 10 [6] – beschleunigt.
- Durch kundenspezifische Objekte und Toolkits von Drittanbietern transparent erweiterbare Entwicklungsumgebung.

### **4.2 Die Komponenten**

#### **Programmiersprachen**

Die Programmiersprache ANSI-C und deren objektorientierte Erweiterungen Objective-C und C++ werden gleichberechtigt unterstützt. Der Objective-C++ Compiler, der ein Compiler-Frontend beinhaltet, das alle drei C-Dialekte implementiert, basiert auf der GNU Compiler-Technologie der Free Software Foundation (FSF).

Neben dem Compiler wird u.a. der GNU Assembler und symbolische Debugger verwendet. Alle GNU Werkzeuge und die von NeXT unterstützte GNU C++ Klassenbibliothek libg++ sind im Quellcode verfügbar.

Zudem ist die Programmierung in Display PostScript (siehe Window Server) direkt und über die Einbettung in C-Funktionen, sog. PostScript Wrapper, möglich.

Wesentlich wichtiger als eine bestimmte Programmiersprache ist aber sicher der Vorrat an vorgefertigten Softwarebausteinen, wobei zu bedenken bleibt, ob eine Sprache die Erstellung von Bausteinbibliotheken ggf. besser oder schlechter als eine andere Sprache unterstützt. (Das überschaubar kleine Angebot an Klassenbibliotheken für C++ gibt hier zu denken).



## Toolkits

Mit NEXTSTEP erwirbt der Entwickler in Form der nachstehend kurz beschriebenen Toolkits eine reichhaltige Auswahl solcher Bausteine. Zudem sind eine Vielzahl weiterer Bausteinbibliotheken als ObjectWare von Drittanbietern erhältlich. Alle von NeXT angebotenen Toolkits sind in Objective-C geschrieben.

### •Application Kit

Das Application Kit bildet als *«Integrated Object-Oriented Application Framework»* die Grundlage jeder graphischen NEXTSTEP-Anwendung. Es beinhaltet eine komplette Bibliothek von Benutzerschnittstellen-Objekten (z.B. Window, View, Control, Responder, u.v.a.m.), die nicht nur das Aussehen einer Anwendung (inkl. ggf. Beschreibung der zweidimensionalen Darstellung mit PostScript), sondern auch ihr Verhalten definieren, z.B. die Reaktion auf Benutzeraktionen wie Mausklick oder Tastatureingabe.

Durch vorgefertigte Klassen zur Texthandhabung, Farb- und Schriftauswahl, usw. (siehe *«Der NEXTSTEP Vorteil»*: *«einmal gelernt – immer gelernt»*), ist die gleichartige Funktionalität von Anwendungen sichergestellt.

Kurz: Das Application Kit implementiert den einheitlichen NEXTSTEP *«Look & Feel»*.

### •Database Kit

Zielsetzung bei der Entwicklung des Database Kits war die Erstellung eines datenbankunabhängigen Toolkit, das vollständig in die NEXTSTEP-Entwicklungsumgebung integriert ist, auf einer offenen objektorientierten Programmierschnittstelle basiert und somit einfach und flexibel erweiterbar ist.

Das Database Kit ist ein Werkzeugkasten für Entwickler zur Erzeugung graphisch-interaktiver Datenbankanwendungen, die unabhängig vom Datenbankserver sind. Derzeit werden u.a. die unten genannten Datenbankmanagementsysteme unterstützt, weitere werden in Zukunft folgen:

Sybase, Oracle (beide im Lieferumfang enthalten), sowie DB/2, Gupta, Indexing Kit, Informix, Ingres, Interbase, Teradata (von Drittanbietern).

Das Database Kit ist neben dem Mach Kit das prominente Beispiel für eine Klassenbibliothek, die die Entwicklung von Client-/Server-Architekturen extrem vereinfacht.

### •Indexing Kit

Das Indexing Kit implementiert ein Framework für effizientes Management grosser Datenmengen inklusive schnellem Datenspeicher und Retrieval.

Zu den Hauptleistungsmerkmalen zählen: Datenintegrität, Transaktionskonzept, exzellente Performance, Unterstützung von MACH-Threads, tiefgehende Integration in die NEXTSTEP Entwicklungsumgebung und die Eigenschaft, effizient Objective-C Objekte sowie unstrukturierte Daten, wie Text, Ton und Bilder, abzuspeichern und wiederzufinden.

Das Indexing Kit bildet die Grundlage vieler NEXTSTEP-Anwendungen, z.B. des Digital Librarian und des integrierten Hilfesystems.

### •Mach Kit - Distributed Objects

siehe Abschnitt *«Distributed Objects»* im Kapitel *«Inter-Application Communication»*.

### •3D Graphics Kit

Das 3D Graphics Kit stellt die Programmierschnittstelle zur Nutzung von Interactive RenderMan dar. Ähnlich dem Application Kit umfasst es Benutzerschnittstellen-Objekte, die allerdings RenderMan Interface Byte-stream (RIB) Code statt PostScript Code an den Window Server senden.

### •Sound Kit und Music Kit

Das Sound Kit umfasst Klassen zur Aufnahme, Anzeige und Wiedergabe von Tondaten. Das Music Kit zur computergestützten Komposition wird zwischenzeitlich am *Center for Computer Research in Music and Acoustics* der Universität Stanford weiterentwickelt.

### •NetInfo Kit

Das NetInfo Kit ist eine Sammlung von Klassen zum Zugriff auf sog. NetInfo Domains (siehe Abschnitt *«Netzwerkdienste und NetInfo»*).



## •Driver Kit

Driver Kit ist ein in Objective-C realisiertes objekt-orientiertes Toolkit, das die Erstellung von Gerätetreibern extrem vereinfacht. Damit wird die objektorientierte Vorgehensweise zum ersten Mal durchgängig auf die Treiberentwicklung angewendet.

Das Driver Kit enthält u.a. allgemeine Klassen für bestimmte Bustypen, wie ISA und EISA, sowie spezielle Klassen für bestimmte Kartentypen, wie Netzwerk-, Graphik-, SCSI Controller Karten u.v.a.m.

Das Driver Kit ist Thread-basiert, Interrupts können verzögert behandelt werden, wobei der MACH Kern über MACH Nachrichten mit dem jeweiligen Treiber kommuniziert, wenn ein spez. Interrupt erkannt wurde.

## Werkzeuge

### •InterfaceBuilder

Mit dem Interface Builder wird eine Anwendung graphisch-interaktiv erstellt, vergleichbar mit einem Zeicheneditor, in dem eine Zeichnung aus beliebigen graphischen Elementen, wie etwa einem Kreis oder einem Rechteck, zusammengesetzt wird.

Dabei ist Interface Builder weit mehr als nur ein «Screen Painter», Ressourcen- oder Maskeneditor, denn neben der reinen Platzierung der graphischen Repräsentanten von Objekten, z.B. eines Schalters, Schiebepalkens oder Textfelds, wird direkt die Interaktion der verschiedenen Objekte bestimmt, z.B. bei Bewegen des Schiebepalkens verändert sich die Anzeige des Textfelds.

Dabei generiert Interface Builder keinen Code, sondern archiviert die verwendeten Objekte in einer Datei. Zur Laufzeit des Programms werden diese Objekte dann de-archiviert und quasi «wieder zum Leben erweckt».

Interface Builder ist vollständig erweiterbar, d.h. zu jedem Zeitpunkt können neue Objektbeschreibungen, sogenannte Paletten (siehe «ObjectWare»), hinzuge-laden werden und sind dann sofort verfügbar.

Interface Builder kann somit mit Fug und Recht als graphischer Objekteditor bezeichnet werden.

### •ProjectBuilder

Das Werkzeug zur Verwaltung aller Projektressourcen inklusive Schnittstellenbeschreibung, Klassenimplementierung und InterfaceBuilder User Interface Beschreibung.

### •HeaderViewier

Ein Klassenbrowser, der neben der Klassenhierarchie alternativ die gesamte Klassendokumentation und Schnittstellenbeschreibung anzeigt und Stichwortsuche erlaubt.

### •FileMerge

Graphisches Programm, das beliebige Dateien vergleichen und mischen kann und mittels seiner offenen Programmierschnittstelle einfach in ein Source Code Management System integrierbar ist.

### •DBModeler

Werkzeug zur Extraktion eines Datenbankschemas zur Verwendung im Database Kit.

### •Compiler

siehe Abschnitt «*Programmiersprachen*».

### •Edit und Debugger

Graphischer Quellcodeeditor, der u.a. die Programmierung im RTF Format inkl. Hypertext-Links erlaubt und den GNU Debugger transparent integriert.

**Peter Lipps**

Leiter Systems Engineering  
NeXT Computer Deutschland GmbH

## Quellennachweis:

Dieser Artikel erschien in Offene Systeme, Verlag GUUG e.V., München, Band 3, Nr 1, Februar 1994



## Literaturhinweise:

- [1] Third-Party Products  
Third-party products for NEXTSTEP  
*NeXT Computer, Inc.*, M6053, 4/93
  
- [2] ObjectWare  
A Catalog of Objects and Kits for NEXTSTEP  
*NeXT Computer, Inc.*, #1 M5014.01, 4/93
  
- [3] The NEXTSTEP Advantage  
A technical guidebook to object-oriented programming  
*NeXT Computer, Inc.*, #1 M5109.00, 3/93
  
- [4] NeXTSTEP Programming - STEP ONE: Object-Oriented Applications, ISBN 0-387-97884-4  
Simson L. Garfinkel and Michael K. Mahoney, *Springer Verlag*
  
- [5] NEXTSTEP Developers Library  
NEXTSTEP Development Tools and Technique,  
ISBN 0.201.63249.7  
NEXTSTEP Operating System Software,  
ISBN 0.201.63252.7  
NEXTSTEP Programming Interface Summary,  
ISBN 0.201.63253.5  
NEXTSTEP User Interface Guidelines,  
ISBN 0.201.63250.0  
NEXTSTEP General Reference, Vol. 1,  
ISBN 0.201.62220.3  
NEXTSTEP General Reference, Vol. 2,  
ISBN 0.201.62221.1  
NEXTSTEP OOP and the Objective C Language,  
ISBN 0.201.63251.9
  
- [6] NeXTSTEP vs. Other Development Environments  
Comparative Study  
*Booz-Allen & Hamilton*, M4882, 1/92

*Anm. der Redaktion:* Ich möchte Peter Lipps und Max Winiger ganz herzlich für die Überlassung dieses interessanten Artikels danken.



## Was sie schon immer über TeX wissen wollten - Teil 5

Die Erstellung schöner Tabellen gehört zu den interessantesten, aber auch schwierigsten Aufgaben im Satzbereich. Damit eine Tabelle ansprechend aussieht, ist der Abstand zwischen den Spaltenelementen passen zu wählen, sind bestimmte Teile durch andere Schriftarten hervorzuheben und die einzelnen Tabellenteile durch Linien zu gliedern.

### Keine Angst vorm Tabellensatz

Beim Tabellensatz gibt es zwei verschiedene Aufgabenstellungen: auf der einen Seite liegen die Daten vor, und diese sollen als Tabelle gestaltet werden, auf der anderen Seite liegt das Format als Vorgabe fest, etwa ein Fahrplan, und die Tabelle muss nur noch gefüllt werden.

Zum Satz von Tabellen gibt es in TeX zwei Möglichkeiten: Nutzung von Tabulatoren und die automatische Ausrichtung der Tabelle mittels des Befehls `\halign`, der für *horizontal alignment* steht.

Beim Satz mittels Tabulatoren wird die Breite der Tabellenspalten explizit vorgegeben. Die einfachste Form der Beitenangabe ist die Vorgabe der Spaltenanzahl:

```
\settabs 3 \columns
```

definiert Tabulatoren für 3 Spalten. Die Aktivierung der Tabulatoren geschieht mittels `\+ ... \cr`. Die Spalten zwischen `\+` und `\cr` werden mittels Tabulatoren gesetzt. Das Tabulatorsymbol ist dabei `'&'`. Die Eingabe

```
\settabs 4 \columns
\+ 1.Spalte & 2.Spalte & 3.Spalte \cr
\+ eins & zwei & drei \cr
\+ one & two & three \cr
```

ergibt 3 Tabellenspalten, die entsprechen mit den angegebenen Informationen gefüllt werden.

Im Gegensatz zum «Tab» einer normalen Schreibmaschine bewirken TeX-Tabs die Positionierung auf die jeweils nächste Spalte. Beispielsweise wird durch die Eingabe von `'&&'` auf den Anfang der dritten und nicht der zweiten Spalte positioniert! Eventuell kann es dadurch zu Überschreibungen kommen, da dabei nicht berücksichtigt wird, wie voll die beiden ersten Spalten schon sind.

Die zweite und meist praktischere Technik, um Spaltenbreiten zu definieren, geschieht mittels Vorgabe einer Musterzeile. Nach `\settabs` folgt einfach eine normale Tabulierungszeile mit `\+` am Anfang, `&` für jede Tabulatorposition und `\cr` am Schluss. Durch ein `\quad` können einzelne Spalten mit zusätzlichem Leerraum versehen werden. Beispiel:

```
\settabs\+Verfahren \quad & 100 Elemente
& 200 Elemente \cr
\+Verfahren & 100 Elemente & 200 Elemente \cr
\+Shell-Sort & 70 & 250 \cr
\+Heap-Sort & 50 & 100 \cr
\+Quick-Sort & 40 & 60 \cr
```

Verfahren	100 Elemente	200 Elemente
Shell-Sort	70	250
Heap-Sort	50	100
Quick-Sort	40	60

Diese Tabelle ist aber noch nicht besonders schön. Die Ausrichtung der Elemente in den Spalten soll rechtsbündig sein. Durch Einfügen von `\hfill` erreicht man eine solche Ausrichtung.

```
\settabs\+Verfahren\quad & 100 Elemente
& 200 Elemente &\cr
\+Verfahren & 100 Elemente & 200 Elemente &\cr
\+Shell-Sort & \hfill 70 & \hfill 250 &\cr
\+Heap-Sort & \hfill 50 & \hfill 100 &\cr
\+Quick-Sort & \hfill 40 & \hfill 60 &\cr
```



Damit der Befehl `\hfill` auch in der letzten Spalte wirkt, muss die Breite dieser Spalte bekannt sein. Daher wurde noch ein zusätzliches `'&'` vor `\cr` gesetzt!

Verfahren	100 Elemente	200 Elemente
Shell-Sort	70	250
Heap-Sort	50	100
Quick-Sort	40	60

Jedes Tabellenelement bildet für sich einen eigenen Bereich, der sonst nur durch Klammerung mit `'{'` und `'}'` erreicht werden kann. Damit können in einer Tabelle durchaus andere Schriftgrößen oder Schriftarten verwendet werden; diese sind dann jeweils nur auf ein Element beschränkt. Die rechtsbündige Ausgabe wurde oben vorgestellt – zentriert wird, indem auf beiden Seiten eines Tabellenelementes ein `\hfill` gesetzt wird. Das folgende Beispiel zeigt eine Zeile aus der obigen Tabelle, bei der die erste Spalte fett gedruckt und die Daten zentriert werden.

```
\+\bf Heap-Sort & \hfill 50 \hfill &
\hfill 100 \hfill &\cr
```

Der Schreibaufwand für eine solche Technik erscheint mit Recht ziemlich umfangreich. Eine komfortablere Technik, solche Tabelle auch automatisch zu setzen, wird im folgenden Abschnitt dargestellt. Noch eine Anmerkung: `\settabs`, `\+` und `\cr` benutzen intern den `\halign`-Befehl (siehe nächster Abschnitt). Jede einzelne Tabellenzeile ist im Prinzip eine Tabelle für sich allein. Damit wird die Tabelle, wenn sie nicht in einer `\vbox` verpackt ist, auch über Seitengrenzen hinweg umgebrochen.

### Automatischer Tabellensatz

Wirklichen Komfort beim automatischen Tabellensatz erreicht man erst durch den Befehl `\halign`. Dabei wird die gesamte Information, die zu einer Tabelle gehört, eingelesen und aus ihr die nötige Spaltenbreite bestimmt. Das grösste Element in jeder Spalte legt dann jeweils die Spaltenbreite fest.

Der Befehl `\halign` besitzt die folgende Syntax:

```
\halign{
  Musterzeile \cr Zeile 1 \cr Zeile 2 \cr ...
  Zeile n \cr
}
```

Die wichtigsten Elemente der Musterzeile sind `'#'` und `'&'`. (Frage: Wer hat gewusst, dass das `'#'`-Zeichen im deutschen Sprachraum *Kanalgitter* genannt wird?!) Dabei trennt `'&'` – wie bei `\settabs` – die einzelnen Spalten. `'#'` ist Platzhalter für die Stelle, an der das Tabellenelement eingefügt werden soll. Der Text links von `'#'` wird jeweils vor das Element, der Text rechts von `'#'` bis zum `'&'` hinter das Element eingefügt. Dies soll zunächst an unserem bekannten Beispiel demonstriert werden:

```
\halign{
  % Musterzeile
  \bf # \quad & \hfill # \quad & \hfill # \cr
  % Zeile 1
  Verfahren & 100 Elemente & 200 Elemente \cr
  \noalign{\smallskip}
  Shell-Sort & 70 & 250 \cr % Zeile 2
  Heap-Sort & 50 & 100 \cr % Zeile 3
  Quick-Sort & 40 & 60 \cr % Zeile 4
}
```

Verfahren	100 Elemente	200 Elemente
Shell-Sort	70	250
Heap-Sort	50	100
Quick-Sort	40	60

Dazu einige Anmerkungen: In der ersten Spalte wird automatisch auf die Schriftart *bold* umgeschaltet, die hinteren Spalten werden automatisch rechtsbündig gesetzt, indem von jedem `'#'` ein `\hfill` steht. Durch `\noalign{}` wird Information eingefügt, die sich nicht auf den Satz eines Tabellenelementes auswirkt, sondern auf die Tabelle als Ganzes.

Etwas merkwürdig am Befehl `\halign` ist das Format der Musterzeile. An dieses Konzept hat man sich erst einmal zu gewöhnen. Vor dem die Musterzeile been-



denden `\cr` darf als letztes Element nur ein `'#'` auftreten, kein `'&'`! Es darf also keine unvollendete Spalte begonnen werden, sonst wird der Fehler `«missing #»` gemeldet.

Die einfachste Form der Musterzeile ist in der folgenden Anweisung verwendet:

```
\halign{#&#&#&#\cr ...}
```

Dies erzeugt eine Tabelle mit 4 Spalten, bei denen jede so breit ist wie ihr entsprechend grösstes Element. Die Elemente werden jeweils in seiner Spalte linksbündig nebeneinander gesetzt. Die Ausgabe erfolgt dabei ohne weiteren Leerraum, da weder in der Musterzeile noch in den Eingabezeilen Leerraum vorgegeben wird. Durch Leerzeichen kann die Eingabe besser gegliedert werden. Insbesondere wird dann auch zwischen den einzelnen Spalten mindestens der Leerraum eines Spacezeichens gelassen.

Es können nun spaltenspezifische Einstellungen in die Musterzeile gesetzt werden. Selbst komplizierte Konstruktionen mit einer automatischen Umschaltung in den Mathematik-Modus sind möglich:

```
\halign{
  \it # \quad & $$$ \cr
  Minimum bei & \sqrt{\pi} \cr
  Maximum bei & \sqrt{\pi+1} \cr
}
```

$$\begin{array}{l} \textit{Minimum bei} \quad \sqrt{\pi} \\ \textit{Maximum bei} \quad \sqrt{\pi+1} \end{array}$$

Allerdings tritt auch manchmal die Situation auf, dass die Musterzeile mit ihren konstanten Einstellungen nicht ganz passt. Oft möchte man bei einzelnen Elementen auf die Vorgabe der Musterzeile verzichten. Hier hilft der Befehl `\omit`, der am Anfang eines Tabellenelements gesetzt, die Vorgabe durch die Musterzeile unterdrückt. Im vorherigen Beispiel wurde in der zweiten Spalte in den Mathematik-Modus umgeschaltet. In der folgenden Tabelle soll dies bei der letzten Zeile nicht geschehen:

```
\halign{
  \it # \quad & $$$ \cr
  Minimum bei & \sqrt{\pi} \cr
  Maximum bei & \sqrt{\pi+1} \cr
  Unstetigkeit & \omit\it nicht gefunden
  \cr
}
```

Hier wird im letzten Element ein `\omit` gesetzt; dies reduziert den Mustereintrag auf ein `'#'`.

## Spaltenausrichtung

Ein Standardfall beim Tabellensatz ist die Ausgabe von Zahlenkolonnen, die nach dem Dezimalpunkt oder Dezimalkomma ausgerichtet sind. Dies wird erst dann problematisch, wenn nicht alle Zahlen die gleiche Anzahl Vor- bzw. Nachkommastellen besitzen. Hier bietet die Verwendung von zwei Tabellenspalten, von denen die erste rechtsbündig und die zweite linksbündig gesetzt wird, einen Ausweg. An Stelle eines Kommas ist jeweils ein `'&'` einzugeben. Beispiel:

```
$$\vbox{
  \halign{
    % Musterzeile
    $\sin\#\^\circ$ \quad & \hfill $#,$
    & $$$ \hfill \cr
    % Zeilen 1 bis 4
    15 & 0 & 259 \cr
    30 & 0 & 5 \cr
    -15 & -0 & 259 \cr
    -30 & -0 & 5 \cr
  }
}$$
```

$\sin 15^\circ$	0, 259
$\sin 30^\circ$	0, 5
$\sin -15^\circ$	-0, 259
$\sin -30^\circ$	-0, 5

Der Mathematik-Modus wurde hier gewählt, um ein mathematisches Minus und keinen Trennstrich zu erhalten.



## Spaltenabstand

Häufig möchte man Tabellen auf eine bestimmte Breite bringen, indem die Spalten mit Leerraum gleichmässig aufgefüllt werden. Die Steuergrösse dafür ist die TeX-Variable `\tabskip`. Durch `\tabskip` bestimmter Leerraum wird bei einer Tabelle mit 4 Spalten an den folgenden Positionen gesetzt:

(A) *Spalte 1* (1) *Spalte 2* (2) *Spalte 3* (3) *Spalte 4* (E)

Dabei sind die Position (A) und (E) der Leerraum vor der ersten Spalte und nach der letzten Spalte. In der folgenden `\halign`-Anweisung sind sämtliche Stellen markiert, die anzeigen welcher `\tabskip`-Wert zu welchem Zeitpunkt benutzt wird:

```
(A)\halign{...#...(1)&...#...(2)&...#...(3)&...#...(E)\cr ...}
```

Das heisst, für den Leerraum vor der Tabelle wird der Wert verwendet, der zum Zeitpunkt des `\halign`-Aufrufs gültig ist. Für den Leerraum nach der letzten Spalte, der Wert, der zum Zeitpunkt des die Musterzeile beendenden `\cr` gültig ist. Der Leerraum zwischen den Spalten regelt sich jeweils zum Zeitpunkt des `'&'`. An den folgenden Beispieleingaben soll die Verwendung von `\tabskip` demonstriert werden:

```
$$\vbox{
  \halign{
    #&#&#\cr % Musterzeile: 3 Spalten
    \it PLZ &\it Ort & \it Vorwahl \cr
    8000 & Z\"urich & (01) \cr
    7000 & St.Gallen & (071) \cr
    ...
  }
}$$
```

Diese Zeilen erzeugen eine Tabelle, bei der der Abstand zwischen den Spalten von den Leerzeichen der Eingabe herrührt. Diese Leerzeichen werden bei der Berechnung der Spaltenbreiten berücksichtigt. Leerzeichen in der Musterzeile hätten übrigens die gleiche Wirkung.

<i>PLZ</i>	<i>Ort</i>	<i>Vorwahl</i>
8000	Zürich	(01)
7000	St.Gallen	(071)

In der folgenden Variante wird ein `\tabskip=15pt` gesetzt. Dadurch verändert sich die erste Zeile zu:

```
$$\vbox{\tabskip=15pt \halign{#&#&#\cr...
```

Elegant wird die Verwendung von `\tabskip` und `\halign`, wenn beim Befehl die gewünschte Breite der zu setzenden Tabelle angegeben wird. Die Breitenangabe geschieht durch `\halign to dimension`. Beispiele:

```
\halign to 200pt{...}
\halign to 15cm{...}
\halign to 0.8\hsize{...}
```

Es ist auch eine Breitenangabe möglich, die die zusätzliche Ausdehnung der Tabelle festlegt. Damit wird also der Leerraum definiert, der bei geeigneter `\tabskip`-Wahl gleichmässig verteilt wird. Beispiel:

```
\halign spread 4cm{...}
```

Um die Tabelle dann überhaupt auffüllen zu können ist der `\tabskip` mit einem «dynamischen» Anteil zu definieren, das heisst mit einer Längenangabe, um die das einzelne Spaltenelement verbreitert werden darf. Der so angegebene «plus»-Anteil ist die Füllmasse für die Tabelle. Die Anweisung

```
\tabskip=15pt plus 200pt
```

erlaubt die Auffüllung jeder Spalte um 200 pt. Unsere Tabelle kann so über die ganze Zeile gedehnt werden:

```
$$\vbox{
  \tabskip=15pt plus 200pt
  \halign to \hsize {#&#&#\cr...
```



## Gerahmte Tabellen

Sehr beliebt sind Tabellen in Kästchen. Die waagrechten Linien sind dabei der einfachere Teil:

```
\noalign{\hrule}
```

zieht eine passend lange waagrechte Linie. Die senkrechten Linien werden Stück für Stück aus Einzelementen jeder Zeile zusammengesetzt. Ein senkrechtes Linienelement heisst `\vrule`. Damit diese Elemente auch wirklich untereinander stehen und den Eindruck einer geschlossenen Linie bilden, sollte man extra Tabellenspalten für die senkrechten Striche einführen. Beispiel:

```
{\offinterlineskip \tabskip=0pt
 \halign{\strut
 \vrule #&          % Spalte 1
 \quad\bf #\quad & % Spalte 2
 \vrule #&          % Spalte 3
 \quad\hfil #\quad & % Spalte 4
 \vrule #&          % Spalte 5
 \quad\hfil #\quad & % Spalte 6
 \vrule #          % Spalte 7
 \cr               % Ende der Musterzeile
 \noalign{\hrule} % horizontale Linie
 &Verfahren && 100 Elem. && 200 Elem. &\cr
 \noalign{\hrule} % horizontale Linie
 & Shell-Sort && 70 && 250 & \cr
 & Heap-Sort && 50 && 100 & \cr
 & Quick-Sort && 40 && 60 & \cr
 \noalign{\hrule} }
}
```

Verfahren	100 Elem.	200 Elem.
Shell-Sort	70	250
Heap-Sort	50	100
Quick-Sort	40	60

Zuerst zur Beschreibung der Musterzeile:

Spalten 1, 3, 5, 7: senkrechte Linie  
 Spalte 2: Text in *boldface*  
 Spalten 4, 6: Text rechtsbündig

Bei diesem letzten Beispiel sind noch einige Befehle zu erklären: `\offinterlineskip` setzt den zusätzlichen Leerplatz den TeX zwischen einzelnen Zeilen ausgibt, auf Null, da sonst die senkrechten Linien nicht aneinanderstossen. `\strut` sorgt für gleich hohe Zeilen, indem praktisch ein unsichtbarer senkrechter Strich gesetzt wird, der die Maximalhöhe und Maximaltiefe eines Zeichen mit Unterlänge hat.

## Hilfsmittel beim Tabellensatz

Wird in einer Musterzeile statt eines `'&'` gleich `'&&'` eingegeben, so bedeutet dies, dass die nachfolgenden Einträge bei Bedarf immer weiter wiederholt werden. Bei

```
\halign{#&&#\cr...}
```

wird das Musterelement `'&#'` beliebig oft wiederholt. Bei

```
\halign{#&&\quad\hfill#&#\quad\cr...}
```

besteht das zu wiederholende Doppelement aus der Anweisung `'&\quad\hfill#&#\quad'`. Beginnt eine Musterzeile gar mit einem `'&'`, so wird der gesamte Eintrag wiederholt. Im folgenden Beispiel sei die Anwendung veranschaulicht:

```
$$\vbox{
 \halign{\&\hfill#\hfill\quad\cr
 $ x = $ & 1 & 2 & 3 & 4 & \dots\cr
 $\Gamma(x) = $ & 1 & 2 & 6 & 24 &
 \dots\cr
 }$$
```

$$\begin{array}{rcccccc}
 x = & 1 & 2 & 3 & 4 & \dots \\
 \Gamma(x) = & 1 & 2 & 6 & 24 & \dots
 \end{array}$$

Einige Befehle sind im Zusammenhang mit dem Setzen von Tabellen sehr interessant. Sie können sehr schön zur optischen Aufbereitung komplexer Tabellen verwendet werden. Stellvertretend möchte ich zum Schluss dieses Artikels eine Gruppe von Befehlen vorstellen, die ein Spaltenelement mit einem Muster bis zu seiner



gewünschten Breite auffüllen. So erzeugt `\dotfill` eine gepunktete Linie, `\hrulefill` eine waagrechte Linie, `\upbracefill` eine nach oben offene geschweifte Klammer, `\downbracefill` das nach unten offene Pendant, `\leftarrowfill` eine Pfeil mit Spitze links und `\rightarrowfill` einen Pfeil mit Spitze rechts. Die folgende Musterzeile benutzt den `\dotfill`-Befehl:

```

\begin{vbox}{\tabskip=40pt
\halign{#\dotfill\quad & ...}
}

```

Das erste Element wird damit jeweils bis zur geforderten Breiten von 40 pt mit Punkten aufgefüllt.

### Genug der Tabellen

Das war's für die heutige Folge. Auch diesmal konnte das gewählte Thema nicht vollständig abgehandelt werden, dazu existieren ganz einfach zuviele Befehle, die in irgendeiner Form für den Tabellensatz verwendet werden können. Der interessierte Leser sei der Vollständigkeit halber noch auf Literatur hingewiesen. Wer sich genauer mit den Befehlen zum Tabellensatz auseinandersetzen will, dem möchte ich Kapitel 22 – *Alignment* – aus dem TeX-Buch von Knuth empfehlen. Die genaue Referenzangabe findet sich am Ende des ersten Teils dieses Kurses.

In der nächsten Folge geht es um die Arbeitsweise von TeX. Zudem werfen wir einen Blick auf die Definition von eigenen Makros.

Dominik Moser



## News rund um TeX

An der 9. Mitgliederversammlung von DANTE, Deutschsprachige Anwendervereinigung TeX, in Kaiserslautern wurden die folgenden Aktualitäten rund um TeX bekanntgegeben.

### German-Style

Auf einigen *ftp*-Servern gibt es noch total veraltete Versionen von `german`, die überhaupt nicht mehr mit den aktuellen Anpassungen von TeX zusammenpassen. Solche Uralt-Makros müssen auf jeden Fall weggeworfen werden. Für die Verwendung mit den *dc*-Fonts ist nach wie vor `german3` (im  $\beta$ -Stadium) die gültige Version. Geplant ist, beide Versionen zu einer zusammenzufassen (2.5), die sowohl mit als auch ohne NFSS funktioniert.

Bei den Trennmustern wird darauf hingewiesen, dass sowohl `ghyphen.min` als auch `ghyphen.max` nicht mehr verwendet werden sollten, sondern nur noch `ghyphen3.tex`. Demnächst soll eine neue Version der deutschen Trennmuster erscheinen, die mit *cm*- und *dc*-Fonts arbeitet.

### NTS

Während des Treffens von DANTE e.V. 1992 wurde die Bildung einer Arbeitsgruppe namens «NTS» (für New Typesetting System) angekündigt. Diese Gruppe hat sich zum Ziel gesetzt, Mittel und Wege zu finden, die Philosophie und Paradigmen von TeX zu erhalten und zu verbessern. Nach anfänglichen Problemen hat man nun entschieden, bei der Arbeit an NTS zweigleisig vorzugehen: eTeX (*extended TeX*) als Erweiterung des vorhandenen Systems und NTS (*next generation typesetting system*) als Neuentwicklung, u.a. mit einer «vernünftigen» graphischen Oberfläche.

Fortsetzung folgt...



## Monitorprobleme, zum Zweiten

**In der letzten Ausgabe berichtete Klaus Kunze an dieser Stelle über seine Probleme mit dem Colormonitor von NeXT. Wer unser vereins-eigenes «Cinemascope-Modell» schon einmal gesehen hat, der weiss, dass vor allem der MegaPixel Display seine Altersprobleme hat. Diese Erfahrungen musste auch Felix Rauch (unser NiCE-Einkäufer) machen:**

### **Liebe NeXT-User**

Da dieses Problem ziemlich weit verbreitet scheint und ich erst kürzlich selbst unangenehme Überraschungen erleben musste, will ich Euch meine Erfahrungen nicht vorenthalten.

Es begann damit, dass mein 17" MegaPixel Colormonitor eines schönen Tages plötzlich kurz aber merklich zuckte. Die Breite des Bildes verringerte sich dabei für Sekundenbruchteile um ein bis zwei Zentimeter. Ich war darüber ziemlich verärgert, denn von einem anderen NeXT-Besitzer wusste ich, dass er das gleiche Symptom schon lange bei seinem eigenen Monitor beobachten konnte und es vorher sogar schon bei einem gewöhnlichen PC entdeckt hatte. Das Zucken wurde immer häufiger und auch stärker, bis sich das Bild auf einen Viertel seiner ursprünglichen Breite zusammengezogen hatte und auch so blieb. Mehr oder weniger zärtliches Klopfen brachte das Bild manchmal wieder zurück, manchmal aber begann es zu flackern und der Bildschirm gab ein bedrohliches Pfeifen von sich. Nun doch um meinen Monitor besorgt, schaltete ich alles ab und versuchte es erst einige Stunden später wieder. Leider kam dann nach wenigen Minuten gar kein Bild mehr...

Jetzt ging es darum, einen Weg zu finden, wieder an einen funktionstüchtigen Monitor zu kommen – am liebsten natürlich an den eigenen. Alles was mir NeXT-Vertreter *Sorbus* anbieten konnte, war ein Ersatzmonitor für rund 1500 Franken! Eine telefonische Nachfrage ergab, dass *Sorbus* aus vertraglichen Gründen den

Monitor nicht einmal aufschrauben darf um herauszufinden, was ihm fehlt. Nicht gerade ein Optimum an Kundenservice! Durch Nachfragen an den verschiedensten Orten kam ich schliesslich auf den «Bildschirm-Doktor». Dort konnte ich meinen Patienten am Morgen vorbeibringen und am Abend bereits wieder funktions-tüchtig und mit frischen Lötstellen abholen. Kostenpunkt: 350 Fr. Und weil ich damit so gute Erfahrungen gemacht habe, gebe ich hier für alle anderen Besitzer von kranken Bildschirmen seine Adresse an:

### **Neidhardt Elektronik**

Dufourstrasse 138  
8008 Zürich  
Tel.: 01/383 91 68  
Fax.: 01/382 21 03

(Reparatur von Farbmonitoren, Schaltnetzteilen und Geräten der Industrie-Elektronik. Konsultationen nur nach telefonischer Vereinbarung!)

Seine Diagnose: Durch das Ein- und Ausschalten und die damit verbundenen Temperaturunterschiede werden die Lötverbindungen mit der Zeit schlechter, weil sich die Drähte ausdehnen bzw. zusammenziehen. Wird mit der Zeit ein Kontakt ganz unterbrochen, kann es zu Folgeschäden kommen und noch empfindlichere Teile durchbrennen. Soweit kam es bei mir glücklicherweise nicht. Ich habe den Doktor um einige Tips im Umgang mit Monitoren gebeten:

- Was tun, wenn's zu Zucken beginnt?  
Sobald sich die ersten Anzeichen eines Defekts zeigen, nicht zögern, sondern sofort in die Reparatur geben. Sonst kann es nur noch schlimmer kommen und mehr zerstört werden (siehe oben).
- Wie häufig sollte man einen Monitor Ein- und Ausschalten?  
Bei dieser Frage muss man sich für einen Kompromiss entscheiden: Es gibt Teile, die durch häufiges An- und Abschalten kaputt gehen und solche, deren Lebensdauer durch zu langen Betrieb (sprich hohe Temperaturen) verringert werden. Sein Tip: Wenn man den Bildschirm für länger als drei Stunden nicht mehr benötigt, sollte man ihn ganz abschalten.



- Welches ist der ideale Bildschirmschoner?

Erstaunlicherweise sollte man keinen Schoner wählen, der den ganzen Schirm Schwarz schaltet, sondern einen, der den ihn beschäftigt, also z.B. farbige Linien zieht oder Fische herumschwimmen lässt. Zeigt ein Monitor nämlich längere Zeit nichts an, so sammeln sich an der Kathode der Röhre (also an dem Teil, der die Elektronen auf die Leuchtstoffschicht schießt) viele Ladungen an und lassen diese mit der Zeit «taub» werden. Folge: Das Bild wird dunkler.

Ich hoffe, mit diesen Tips jemandem geholfen zu haben. Es bleibt noch anzumerken, dass ich mit Herrn Neidhardt keinerlei geschäftliche Beziehungen pflege, sondern nur ein zufriedener Kunde bin. Zudem kann ich keine Garantie für die Richtigkeit seiner Tips übernehmen.

**Felix Rauch**  
felix@nice.ch

## NiCE-Markt

MediaStation zu verkaufen:

*«An integrated multimedia database and authoring system that allows you to capture, store, process, organize and present high-resolution graphics, scanned images, digital audio, video, animation and text.»*

(Originalverpackt, Motorola-Version)

NP: \$ 995.-      VP: SFr. 50.-

**Martin Beerli**  
Tel.: 01/362 95 72 (tagsüber)

## «Sicherer» Kode jetzt doch entschlüsselt

**NiCE-Mitglied Hans Lange aus Bonn hat Uns einen Zeitungsartikel der *Frankfurter Allgemeinen Zeitung* vom 8. Juni 1994 zukommen lassen. Darin geht es um das RSA-Public-Key Verfahren, das im Artikel «*Die Kryptographie erobert den Alltag*» im *PowerKey 1/94* beschrieben wurde.**

(Sinngemäss in Auszügen wiedergegeben)

Vier Mathematikern ist es gelungen, einen für sicher gehaltenen kryptographischen Kode zu entschlüsseln. Besondere Brisanz erhält dieser Erfolg dadurch, dass das unter dem Namen RSA bekannte Kodierungsverfahren auch von Regierungen und vermutlich auch von Geheimdiensten verwendet wird.

...

Die Sicherheit dieses Kodierungsverfahrens beruht auf der ausserordentlichen Schwierigkeit, den weit mehr als hundertstelligen öffentlichen Schlüssel in Primfaktoren zu zerlegen. All jene, die glaubten, die Lösung dieser Aufgabe für eine Zahl mit 129 Stellen liege jenseits heutiger Rechnerkapazitäten, musste sich eines Besseren belehren lassen. In nur acht Monaten gelang es den Wissenschaftlern, den Kode mit Hilfe eines mathematischen Algorithmus zu knacken. Allerdings bedurfte es der Mithilfe von 600 freiwilligen Mitarbeitern aus 25 Ländern, die über das Internet Teilaspekte der gewaltigen Aufgabe zugewiesen bekamen. Potentiellen Anwendern des Kodierungsverfahrens raten die Wissenschaftler, auf längere Schlüssel auszuweichen...

*Anm. des Autors:* Wie ich schon in Schlusswort meines Artikels geschrieben habe, ist es bei jedem Verfahren fast nur eine Frage des Aufwandes, ob man den entsprechenden Code knacken kann. Und acht Monate scheint mir doch ein ziemlicher Aufwand, zumal z.B. in PGP eine Schlüssellänge von 128 Bit als tiefste Sicherheitsstufe gilt.

dm

# Impressum



**Herausgeber:** NiCE – NeXT User Group

**PowerKey** ist das Magazin der NiCE und erscheint vier mal jährlich. Ein Abonnement ist in der Mitgliedschaft bei der NiCE enthalten.

**PowerKey** wird vollständig auf NeXT-Computern mit *WriteNow* erstellt.

Auflage: 250 Exemplare • Einzelverkaufspreis: Fr. 7.–

## Redaktion:

*Verantwortlicher Redaktor:* Dominik Moser (dm)

*Mitarbeiter dieser Ausgabe:* Neil Franklin, Feliz Rauch

“Wir bemühen uns, sowohl die männliche als auch die weibliche Schreibform zu verwenden. Wo wir dies zugunsten einer besseren Lesbarkeit nicht tun, beziehen sich sämtliche Aussagen auf Männer und Frauen.”

## Redaktionsadresse:

Dominik Moser, Dörflistrasse 41, 8942 Oberrieden

Anfragen und Inserate von Mitgliedern bitte nur schriftlich.

Adressänderungen bitte an den Aktuar.

## Anzeigenpreise:

8 × 5 cm Fr. 60.–, 8 × 10 cm Fr. 100.–, 16 × 10 / 8 × 20 cm Fr. 175.–, 1 Seite A4 Fr. 300.–, Mengenrabatt bereits ab 2 Ausgaben!

Einmalige, nicht gewerbmässige Inserate von Mitgliedern gratis.

## Copyright:

Copyright aller Artikel bei NiCE, ausgenommen Artikel vom Internet (bezeichnet) bei den entsprechenden Autoren. Die gewerbliche Nutzung, insbesondere der Programme, Schaltpläne, gedruckten Schaltungen und Adressen von Mitgliedern, ist nur mit schriftlicher Genehmigung des Herausgebers zulässig. Nachdruck, auch auszugsweise, nur mit schriftlicher Genehmigung des Herausgebers. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Für unverlangt eingesandte Manuskripte übernimmt die Redaktion keine Haftung. © 1994 NiCE – NeXT User Group.

## Haftung:

Der Herausgeber lehnt jegliche Haftung für direkte und indirekte Schäden oder Folgeschäden ab. Für abgedruckte Tips und Anleitungen kann keine Garantie übernommen werden. Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden.

## NiCE – NeXT User Group

Vereinsadresse: NiCE – NeXT User Group  
Rechenzentrum  
ETH Zentrum  
8092 Zürich

PC-Konto: 80-46102-05

## Vorstand

Präsident: Neil Franklin  
Morgenweg 8, 8404 Winterthur

Vizepräsident: Christian Limpach  
Mainaustr. 44, 8008 Zürich

Aktuar: Albin Mächler  
Jonas Furrer-Str. 97, 8400 Winterthur

Kassier: Werner Burri  
Chisweg 17, 5313 Klingnau

Redaktor: Dominik Moser  
Dörflistr. 41, 8942 Oberrieden

Verleger: Peter Burgdorfer  
Austr. 51, 8953 Dietikon

Beisitzer: Adriano Gabaglio  
Brunnmattstr. 22a, 6010 Kriens

Beisitzer: Patrik Lori  
Schumacherweg 44, 8046 Zürich

e-mail Vorstand Vorstand bzw. <Vorname>.<Name>  
oder einzeln: @nice.usergroup.ethz.ch

Mailbox: Tel. 01 251 20 02  
# **call nice**  
nice login: **mailbox**

**PowerKey 3/94 erscheint Ende September 1994**

Redaktions- und Anzeigenschluss: 15. September 1994