PowerKey 3/94

Inhalt

Editorial

Liebe Leserin, lieber Leser

Wer andern eine selbst hinein! So oder ähnlich dürfte wohl der aufmerksame und zynische Leser auf das Editorial der letzten Ausgabe reagiert haben. Da war doch genau an dieser Stelle die Rede von «*knapp einem Monat*» zwischen dem Erscheinen der Ausgaben 1/94 und 2/94 – dabei hat es im Endeffekt wohl etwas länger gedauert. Das lag grösstenteils daran, dass Hardware-, genauer gesagt Monitorprobleme, unseren Zeitplan leicht orkanartig durcheinanderwirbelten! Ich möchte mich an dieser Stelle, im Namen von Redaktor und Verleger, für die entstandene Verzögerung entschuldigen. Soll nicht wieder vorkommen – zumal habe wir uns fest vorgenommen, dass es nicht wieder vorkommen soll.

Nach meinem wiederholten Aufruf in der letzten Ausgabe betreffend Mitgliederartikel haben sich erfreulicherweise ein paar wenige von Euch gemeldet. Nichts desto trotz, stammen die meisten Artikel, auch in dieser Ausgabe wieder, von Vorstandsmitgliedern. Ich möchte Euch einmal mehr dazu ermuntern aktiv an der Gestaltung des ***PowerKey*** teilzunehmen, indem Ihr Eure Artikel, Testberichte, Kommentare oder Erfahrungen der Redaktion zur Verfügung stellt.

Jetzt noch zu einem erfreulicheren Thema. Das bereits einmal angekündigte neue Informationssystem der NiCE – die NiCE Postkarte – hatte mit der Einladung zur NiCE-Grillparty ihren ersten Auftritt. In Zukunft soll mit Hilfe dieser Postkarten aktuell über Aktionen oder Veranstaltungen der NiCE informiert werden. Damit haben wir neben dem ***PowerKey***, das ja vorerst nur noch viermal jährlich erscheinen soll, ein weiteres Publikationsorgan, das es uns erst noch erlaubt, kurzfristiger und somit gezielter zu informieren.

Wie aktuell das ***PowerKey*** ist, konnte man erst vor wenigen Tagen wieder über's Internet erfahren. In der Ausgabe 6/93 schrieb Neil Franklin in seinem Kommentar zum Thema Zusammenarbeit zwischen NeXT und Sun: «*Ich bin sicher, in ein bis zwei Jahren bringt NeXT OpenStep NT heraus, um die Firma zu retten. NT wird schnell die Hauptplattform für OpenStep werden und OpenStep die Hauptsoftware von NeXT.*»

Nun will ein findiger Schreiberling der Zeitschrift PC Week herausgefunden haben, dass NeXT tatsächlich an einen OpenStep-Port für Windows NT und Chicago (das neuerdings *Windows 95* heissen soll) denkt. Zumindest beruft er sich auf eine Rede, die Steve Jobs in New York gehalten haben soll. Das Internet reagierte mit Wohlwollen auf die Neuigkeit und die offiziellen Stellen von NeXT liessen verlauten, dass man lediglich «mögliche Plattformen für einen OpenStep-Port evaluiere».

Vielleicht sollten wir diesen Herren eine Mitgliedschaft bei der NiCE nahelegen. Damit würden sie automatisch in den Genuss des ***PowerKey*** kommen und könnten dort jeweils nachlesen, was sie in der näheren Zukunft vielleicht zu entscheiden gedenken.

Nun aber zum Inhalt: In der aktuellen Ausgabe von ***PowerKey*** erscheint ein Bericht von der diesjährigen NEXTSTEP Expo 94, die Ende Juni in San Francisco stattfand. Daneben findet Ihr die Fortsetzung von Neil Franklin's Artikel zum Thema Internet und SLIP. Den Abschluss bilden ein Softwarebericht über Virtuoso und NeXT-Oberon, Teil 6 des TeX-Kurses und die nicht ganz ernst gemeinte Frage «Do you speek C?».

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

Dominik Moser, Redaktor

NiCE Internals

Veranstaltungen

NiCE-Meetings

Wann: jeden zweiten Dienstag im Monat
Beginn 19.00 Uhr (bis ca. 21.30)
Wo: ETH Zürich, Hauptgebäude
Raum wird noch bekanntgegeben!
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.

Achtung!

Wegen mangelndem Angebot an Vortragsthemen ist es uns zur Zeit nicht möglich, im voraus über die Durchführung der NiCE-Meetings zu berichten. Meetings werden neu kurzfristig per Postkarte oder Mail angekündigt! Fällt ein Meeting aus, findet automatisch eine talk&copy Veranstaltung im Informatik-Gebäude (IFW) der ETH statt!

NiCE-Agenda 1994

11. Oktober: Meeting
Themen noch nicht bekannt

25. Oktober: talk & copy

8. November: Meeting
Themen noch nicht bekannt

22. November: talk & copy

13. Dezember: Meeting
Themen noch nicht bekannt

27. Dezember: Fällt wegen Weihnachtsferien aus!

FTP Server

Auf dem NiCE ftp-Server befinden sich die neusten Applikationen, Tools und Daten direkt ab dem grossen ftp-Archiv in München. Zugriff für Mitglieder entweder anonym über Internet oder persönlich während jeder talk&copy Veranstaltung.

Wichtig: Nachdem unsere Archiv-Harddisk für ein paar Wochen in Reparatur war, ist der NiCE-Server wieder voll einsatzbereit!

Missing Fax?

Wer hat uns ein Fax geschickt? Vor kurzem haben wir ein völlig zerstückeltes Fax (aus Deutschland?) bekommen, in dem ein *Hans* Fragen zum Zyxel-Modem stellt. Damit wir die Fragen beantworten können, brauchen wir unbedingt das komplette Fax! Also bitte noch einmal absenden. Danke!

NEXTSTEP EXPO 1994

Vom 21. bis 23. Juni 1994 fand im Moscone Center in San Francisco (USA) die NEXTSTEP EXPO 1994 statt. Vieles von dem was im Vorfeld der EXPO bereits bekannt war, wurde nun offiziell angekündigt bzw. vorgestellt. Die wichtigsten Stichworte für die nähere Zukunft lauten daher erwartungsgemäss OpenStep, DEC, NEXTSTEP 3.3, PDO und EOF.

OpenStep und SunSoft

Am 5. April dieses Jahres verkündete NeXT zusammen mit SunSoft die Verfügbarkeit des *OpenStep Developer Starter Kit*, der Entwicklungsumgebung für OpenStep. Damit soll es Entwicklern möglich sein, verteilte Applikationen mit nur einem Bruchteil des bisherigen Aufwandes zu schreiben. Applikationen welche mit dem OpenStep Developer Starter Kit entwickelt wurden, können später problemlos auf die Solaris-basierende Implementation von OpenStep portiert werden, welche im Rahmen von SunSoft's *Distributes Object Environment* (DOE) Projekt Anfang 1995 ausgeliefert werden soll. Die Interface-Spezifikation von OpenStep wurde Ende Juni veröffentlicht und an Standardisierungsbehörden wie die *Object Management Group* (OMG) und X/Open überwiesen.

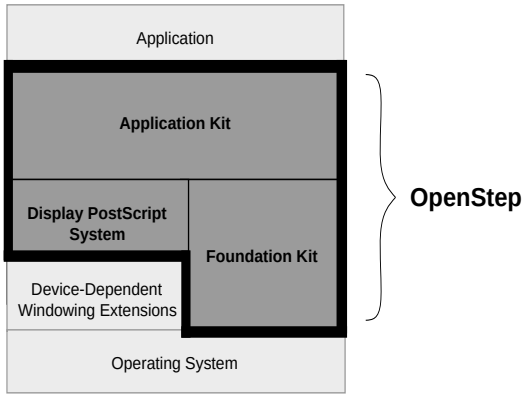


Fig. 1: Genereller Aufbau von OpenStep

OpenStep soll gemäss NeXT und SunSoft zum kommenden Industriestandard für ein betriebssystemunabhängiges, objektorientiertes *Application Programming Interface* (API) werden, das auf den unterschiedlichsten Host-Betriebssystemen laufen soll. Applikationen, die sich OpenStep bedienen, sollen über alle Implementationen von OpenStep hinweg portabel sein. Figur 1 zeigt den generellen Aufbau von OpenStep, Figur 2 die Eingliederung von OpenStep in Solaris.

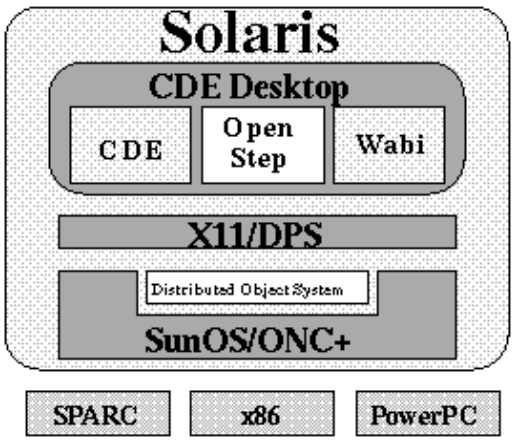


Fig. 2: OpenStep unter Solaris

OpenStep ermöglicht ein verteiltes, objektorientiertes Konglomerat zwischen Systemen mit unterschiedlichen Betriebssystemen und Prozessoren wie x86, PA-RISC, SPARC, Alpha AXP und PowerPC, was dem Softwareentwickler einen von Anfang an grösseren Markt öffnet und die Plattformabhängigkeit minimiert.

Digital Equipment Corporation

Wie im Vorfeld der diesjährigen EXPO bereits bekannt wurde hat NeXT mit der Digital Equipment Corporation (DEC) nach Sun Microsystems und Hewlett-Packard (HP) bereits den dritten wichtigen Verbündeten gewinnen können. Zitat Enrico Pesatori, Vizepräsident der Digital Equipment Corporation's Systems Business Unit:

«*Digital has evaluated object-oriented technology shipping today and on the horizon and Digital has chosen NeXT. OpenStep will play a strategic role for Digital, allowing us to offer customers this premier technology on our high performance Alpha AXP desktop systems. OpenStep on Alpha AXP is a key component to our*

object-oriented product strategy and provides Digital customers with the foundation for complete client/server software solutions based on NeXT technology, including NEXTSTEP for Intel on the client side and Portable Distributed Objects (PDO) and NetInfo running on our UNIX-based servers.»

Wie Steve Jobs bekanntgab, wird Digital OpenStep auf ihr Unix-Derivat OSF/1 für Alpha AXP portieren. Dabei soll es zu einer Verschmelzung von OpenStep und *ObjectBroker* – der Grundlage von Digital's *Common Object Model* (COM) – kommen. Dies wird es jedem von ObjectBroker verwalteten Objekt ermöglichen, mit Objekten aus der NeXT-Umgebung zu interagieren. Digital's ObjectBroker wird heute bereits für Systeme wie Windows, Macintosh, ULTRIX, SunOS, HP-UX und AIX ausgeliefert.

Erste konkrete Produkte sollen Ende Jahr erhältlich sein und direkt von Digital vertrieben werden. Ebenfalls Ende Jahr will NeXT ihre PDOs für Alpha AXP ausliefern. Mit der vollständigen Portierung will Digital 1995 dann eine Reihe von Applikationen ähnlich denen von NEXTSTEP 3.2 anbieten, wie etwa den Workspace Manager, NeXTmail und Entwicklungswerkzeuge aus NEXTSTEP Developer. Für seine Desktop-PCs bietet Digital schon heute NEXTSTEP (for intel) an.

NEXTSTEP 3.3

Basierend auf dem Feedback von Kunden wurden über 1000 Verbesserungen an der bestehenden Version von NEXTSTEP vorgenommen. In den Kernel sind Features von Mach 3.0 und BSD 4.4 eingeflossen, was vor allem Performance und Stabilität verbessern sollte. Der neue Release beinhaltet hauptsächlich Verbesserungen in Sachen Skalierbarkeit und Benutzerfreundlichkeit. Vor allem aber bietet NEXTSTEP Release 3.3 mehr Unterstützung für Intel-basierende PCs. Standards wie PCI und ISA *Plug & Play* werden genauso unterstützt wie PCMCIA, 8-Bit Farbe und *Advanced Power Management*, letztere sind für den Betrieb von NEXTSTEP auf portablen Computern interessant. Release 3.3 enthält über 50 Treiber für die unterschiedlichsten Peripheriegeräte, was die Konfiguration deutlich erleichtert und NEXTSTEP auf einer breiteren Hardwarebasis laufen lässt. Zudem wurden Änderungen an den Administrations-Tools angekündigt, so z.B. am NetInfo-Manager, UserManager und am SimpleNetworkStarter. Neue Netzwerk-Tools sollen die Verwaltung grosser Netzwerke effizienter und weniger kostenintensiv machen.

Deutlich verbessert wurde NeXTmail. Dem Benutzer wird eine grössere Flexibilität in der Speicherung und Verwaltung seiner Mail gegeben. Die lange ersehnte Unterstützung von *Multipurpose Internet Mail Extensions* (MIME) erlaubt es dem Mailteilnehmer neben der normalen ASCII-Mail und dem NeXT-eigenen Format auch eine andere populäre Multimedia-Mail über die gleiche intuitive Schnittstelle zu senden und zu empfangen.

Ebenfalls enthalten sein wird eine neue Version von SoftPC von Insignia Solutions Inc. SoftPC 4.0 unterstützt eine Reihe von *enhanced mode* DOS- und Windows-Applikationen, wie zum Beispiel WordPerfect 6.0 und Aldus PageMaker 5.0.

NeXT Computer, Inc.
900 Chesapeake Drive
Redwood City, CA 94063

NEXTSTEP Release 3.3 soll im vierten Quartal dieses Jahres verfügbar sein. Der Preis wird 795 Dollar betragen, ein Update kostet 199 Dollar.

Enterprise Object Frameworks

NeXT unterstrich in San Francisco auch sein Engagement im Business-Bereich: Ein neues Produkt namens *Enterprise Object Frameworks* (EOF) soll die Vorteile der objektorientierten Programmierung auch Entwicklern von Datenbankapplikationen zugänglich machen. Mit EOF sind sie erstmals in der Lage, wiederverwendbare Objekte zu schaffen, die einen direkten Zusammenhang zwischen der Logistik von Firmenabläufen und dem Inhalt von Datenbanken herstellen lassen.

Diese Objekte sind unabhängig von der darunterliegenden Datenbankstruktur und können über das gesamte (heterogene) Netzwerk einer Firma hinweg genutzt werden. Laut Aussage von Jnan Dash, Vizepräsident von Oracle Corporation, schliesst EOF die Lücke zwischen den mächtigen relationalen Datenbanken und der objektorientierten Modellierung von Geschäftsprozessen. Durch die konsequente Verwendung von EOF sollen sich die Entwicklungs- und Unterhaltskosten für Datenbankapplikationen drastisch senken lassen.

Traditionellerweise verwendeten Entwickler bis anhin eine 4GL-Sprache. Aus einem bestehenden Datenstamm wurden die darzustellenden Werte ausgewählt und zu einer Bildschirmmaske (eine Art Datenfenster) zusammengestellt. Geschäftsrelevante Vorgänge, wie zum Beispiel Zinsberechnungen oder Abschreibungen, waren dabei fest mit der Bildschirmmaske verbunden. Eine neue Auswertung basierend auf den gleichen Daten erforderte genauso eine neue Bildschirmmaske, wie die Änderung von Geschäftsprozessen. EOF bietet da einen alternativen Ansatz: Ausgehend von der Analyse eines Geschäftsvorganges erstellt der Entwickler direkt wiederverwendbare Softwarekomponenten, die sowohl Geschäftsdaten als auch Geschäftsprozesse zur Verarbeitung dieser Daten beinhalten. Die Visualisierung, entsprechend den oben erwähnten Bildschirmmasken, beschränkt sich nunmehr auf die Darstellung von sogenannten *Enterprise Objects*, da die Geschäftsprozesse nicht in der Visualisierung, sondern in den Objekten selbst enthalten ist. Änderungen betreffen somit die Objekte, nicht aber die verschiedenen Bildschirmmasken!

EOF besteht aus drei Modulen: Dem *Enterprise Object Modeler* (vergleichbar mit dem DB Modeler), dem von den Applikationen benötigten *Framework* (die Laufzeitumgebung) und diversen Adaptern für kommerzielle Datenbanken. Zur Zeit sind Adapter für Oracle und Sybase verfügbar. EOF wurde während der EXPO an über 1000 Konferenzteilnehmer verteilt. Die definitive Auslieferung erfolgt im vierten Quartal dieses Jahres zu einem Preis von 299 Dollar.

Canon object.station 41

Bereits Anfang März dieses Jahres wurde bekannt, dass Canon Computer System Inc spezielle, auf NEXTSTEP optimierte Workstations produzieren wird. Gemäss der nun vorliegenden Spezifikation basiert die *Canon object.station* auf einem 100 MHz Intel 486DX4 mit 16 KByte 1st-level und 256 KByte 2nd-level write-through Cache. Durch einen Upgrade-Sockel soll die spätere Verwendung eines Pentium möglich sein. Auf dem Mainboard befinden sich neben 16 MB RAM in 72-pin SIMMs (erweiterbar auf 96 MB) ein 32-Bit Fast SCSI-2 Controller sowie Ethernet; letztere sind über einen VL-Bus verbunden. Drei ISA-Slots und ein kombinierter ISA/VL-Slot stehen für Erweiterungen offen. Canon's *Proprietary Video Architecture* ist mit 2MB VRAM bestückt und bietet bei 72 Hz und 16-Bit Farbe maximal 1280×1024 Pixel. Bei der NEXTSTEP-typischen Auflösung von 1120×832 Pixel ist gar eine Bildwiederholrate von 90 Hz möglich. Der Sound kommt von einer SoundBlaster-kompatiblen Audioeinheit in CD-Qualität.

Fig. 3: Canon object.station 41

An Schnittstellen steht das ganze Spektrum von Fast SCSI-2, über RJ-45 (für Ethernet), Stereo In/Out, Video sowie die gewohnten PC-Schnittstellen zur Verfügung. Ausgeliefert wird die object.station standardmässig mit einem 3.5" 1.44 MByte Diskettenlaufwerk, einer PS/2-Maus und einer Tastatur mit NeXT-spezifischer Tasten-

belegung in einem Desktopgehäuse (siehe Figur 3). Als interne Massenspeicher stehen zwei Fast SCSI-2 Harddisk mit Kapazitäten von 500 MByte oder 1 GByte sowie ein Double-Spin CD-ROM Laufwerk zur Auswahl. Bei den (Farb-) Monitoren hat man die Wahl zwischen Bildschirmdiagonalen von 17" und 21". Vorinstalliert sind in der Grundausstattung nur die User-Version von NEXTSTEP und Indignia's SoftPC Demo. Auf Wunsch (und gegen bare Münze) gibt's zusätzlich DOS, Windows oder die Developer-Version von NEXTSTEP. In den USA geht die Canon *object.station* ab sofort für 8900 Dollar über den Ladentisch. Die Preise für den schweizerischen Markt waren zur Zeit des Redaktionsschlusses noch nicht festgelegt.

Fazit

Die Ausgangslage von NeXT für die anstehende Auseinandersetzungen in der Objektarena hat sich damit weiter verbessert. Mit drei Branchengrössen im Rücken dürfte es leichter fallen, die Vorbehalte zu zerstreuen, die NeXT von vielen grösseren Unternehmen derzeit noch entgegengebracht werden. Im Unterschied zu den Hauptkonkurrenten Taligent und Microsoft, die noch tief in der Entwicklungsphase stecken, verfügt NeXT bereits über ein brauchbares Produkteportfolio.

Dominik Moser, Redaktor

Anm. der Redaktion: «Trademarks mentioned belong to their respective owners.»

NEXTSTEP und NEXTSTEP Developer für HP's PA-RISC ausgeliefert

Wie NeXT Computer, Inc. und die Hewlett-Packard Company am 17. August in Redwood City, Californien bekanntgaben, ist NEXTSTEP und NEXTSTEP Developer Version 3.2 ab sofort für die auf HP's PA-RISC basierenden Workstations HP 9000 Modell 712, 715, 735 und 755 verfügbar.

Wie die beiden Firmen verlauten liessen, wollen sie mit den neuen Produkten vor allem die Finanz- und Telekommunikationsmärkte anvisieren, die heute bereits verstärkt HP-Workstations einsetzen. Auf speziellen Kundenwunsch hin, soll NEXTSTEP ab sofort auch auf HP-Workstations vorinstalliert werden. Für HP-Server bietet NeXT auch seine *Portable Distributed Objects* (PDO) sowie NetInfo unter HP-UX an.
Viele der angesprochenen Banken, Handelsinstitute und Telecom-Firmen benutzen NEXTSTEP – als Plattform für ihre Applikationen oder zur Entwicklung – und Hardware von HP getrennt voneinander. Die leistungsfähigen Rechner der Reihe HP 9000 lassen die Applikationen nun noch schneller laufen und sichern den Unternehmungen so einen Wettbewerbsvorteil.

Wie Steven P. Jobs mitteilte, erfolgt die Auslieferung termingerecht, wie vor einem Jahr angekündigt. NEXTSTEP für HP's PA-RISC ist eine vollständige Portierung des NEXTSTEP-Betriebssystems in der Version 3.2, wie es bereits für Prozessoren von Motorola und Intel erhältlich ist. NEXTSTEP Developer enthält zusätzlich zu der bekannten graphischen, objektorientierten Entwicklungsumgebung auch das sog. *NEXTSTEP Object Framework*, die Basis von OpenStep. Softwarefirmen wie z.B. Lighthouse Design konnten so ihre komplette Produktelinie innert eines Monats portieren. Kostenpunkt für den PA-RISC-Port: ca. 1500 Franken.

(dm)

Verbindung mit dem Internet via SLIP und EUnet

Haben Sie den Artikel «*Das Internet – was ist das?*» im letzten *PowerKey* gelesen und sich dabei gedacht, wie schön es doch wäre, wenn Sie zuhause an ihrem NeXT einen eigenen Internet-Anschluss hätten? Dieser Artikel beschreibt die Installation und Konfiguration der dazu notwendigen Software.

Achtung
Diese Installationsanleitung geht davon aus, dass kein bestehendes Netz (z.B. Ethernet zu lokalem PC) vorhanden ist. Wer ein solches in Betrieb hat muss die Änderungen im Vorgehen selber herausfinden. Ich nehme an, dass sie hauptsächlich im Bereich von *Host-Manager.app* liegen werden, habe dies aber mangels Zeit und Möglichkeiten nicht geprüft.

Bei EUnet anmelden

Um via SLIP mit dem Internet in Verbindung zu treten muss man bei EUnet einen SLIP-Account bestellen. Dieser muss vor der Installation aktiviert worden sein. Ausserdem braucht man für die Installation Angaben von EUnet, wie zum Beispiel den eigenen Accountnamen, IP-Nummern und Passwörter.

Die benötigte Software

NeXT ist der Meinung, dass jeder, der ans Internet will, mit Ethernet arbeiten wird (dies ist in den von NeXT anvisierten «*custom application*» Umgebungen wohl auch weitgehend der Fall), daher ist die für SLIP benötigte Software im NEXTSTEP-Release nicht enthalten. Man muss sie sich selber besorgen, im Gegensatz zu UUCP, wo NeXT alle nötigen Programme mitliefert, und man selbst nur konfigurieren muss. Am einfachsten besorgt man sich die Software, indem man sie an einer *talk&copy* Veranstaltung vom *nice*-Server herunterlädt.

Es sind dies die folgenden Dateien:

```
~ftp/pub/Unix/communication/SLIP_920904-A.*
~ftp/pub/Mail/apps/PopOver.*
```

Die SLIP Installation

Zuerst sollte man sich als `root` einloggen. Von jetzt an muss man sehr vorsichtig sein, da die Sicherheitsvorkehrungen von NEXTSTEP betreffend unbeabsichtigter Beschädigung/Löschung von wichtigen Systemdateien nicht mehr aktiv sind.

1. Dateien auspacken und installieren
Zuerst *SLIP.920904-A.N.b.tar.gz* auspacken, anschliessend *SLIP_920904-A.pkg* installieren. Die Dateien werden in einem neugeschaffenen Verzeichnis namens */usr/dialupip* abgelegt.

2. Dateien modifizieren
Anm. der Redaktion: Bei der Darstellung der zu ändernden Dateien kommt es leider häufiger vor, dass einzelne Zeilen zu breit für eine ***PowerKey***-Spalte sind. Für diesen Artikel gilt folgende Notation: Zeilen, welche zu lang sind, werden aufgeteilt und an der Trennstelle durch ein Returnzeichen (↵) markiert. Beim Abtippen der Zeilen, ist das Returnzeichen durch ein Leerzeichen zu ersetzen und die darauffolgende Zeile anzuhängen. Die im Punkt 2.1 angegebene Zeile muss in der endgültigen Form so aussehen:

```
slip0:…:slipsrv.scrip 012914687:/usr/…
```

2.1 /usr/dialupip/config/diald.conf:

```
slip0:chsun.eunet.ch:cua#38400:slipsrv.scrip↵
012914687:/usr/dialupip/log/trans@0:slipsrv.a
ccess
```

Falls das Modem an Serial B angeschlossen ist oder 38400 Baud nicht verkraftet muss entweder `cua` durch `cub` oder `38400` durch die richtige Geschwindigkeit ersetzt werden. Wie man die richtigen Werte ermittelt steht in meinem Artikel «*E-Mail Verbindung mit der nice*» im ***PowerKey*** 3/93. Wer ein richtiges NeXT-Modemkabel hat kann auch `cufa` oder `cufb` verwenden, mein Mac-Modemkabel versagte dabei leider.

2.2 /usr/dialupip/config/slipsrv.access:

```
inactivity 15
uptimeinterval 15
```

Die Zeile mit `inactivity` gibt die Zeit zwischen dem letzten Datentransfer und dem Auflegen des Modems (in Sekunden) an. Für Mail und News ist 15 ideal (keine Telefonverschwendung), für Telnet und FTP ist dagegen 300 (5 Minuten) ideal, damit während Denkpausen nicht aufgelegt wird. Leider kann man dies nicht selektiv einstellen. Ich lasse normalerweise 15 eingestellt und stelle auf 300 um, wenn ich dies brauche.

2.3 /usr/dialupip/config/config.slip:

```
SLIP0LOCAL=###.###.###.###  (eigene IP-Nummer)
SLIP0REMOTE=146.228.10.15
SLIP0NETMASK=255.255.255.0
SLIP0CONFIG=SLIP
SLIP0DEFAULT=YES
```

2.4 /usr/dialupip/config/slip-srv.script.tcl:

```
log "slip-srv.script.tcl for interface↵
$interface site $site"
log "Now invoking hooks.tcl script."
source "/usr/dialupip/config/hooks.tcl"
log "Now invoking dial-zyxel.tcl script."
source "/usr/dialupip/config/dial-zyxel.tcl"
log "Now invoking login-eunet.tcl script."
source "/usr/dialupip/config/login-eunet.tcl"
```

Wer kein ZyXEL hat muss sich entweder eins zutun oder in der 5. Zeile `zyxel` durch `modem` ersetzen (resultiert in einem etwas langsameren Verbindungsaufbau und keiner optimalen Ausnutzen eines ZyXEL's falls man dies doch mit einem solchen macht).

2.5 /usr/dialupip/config/dial-zyxel.tcl oder *
/usr/dialupip/config/dial-modem.tcl:*
Die mitgelieferte Datei benutzen.

2.6 /usr/dialupip/config/login-eunet.tcl:

```
# TCL script for use with tcldiald.
# Used to put unix host chsun.eunet.ch into
# SLIP mode.
set timeout 20
log "Begin chsun.eunet.ch login"
parity ZERO
sleep 3
expect timeout {error "waiting for login"}↵
"*ogin:*"
xmit "###\r"     (###=Loginname gem. EUnet)
sleep 1
expect timeout {error "waiting for password"}↵
"*assword:*"
xmit "###\r"     (###=Loginpasswort gem. EUnet)
sleep 1
expect timeout {error "waiting for SLIP↵
startup"} "*starting-slip*"
log "Entering SLIP mode"
return "Connected"
```

2.7 /etc/resolv.conf: Neue Datei erstellen!

```
nameserver 146.228.10.15
nameserver 192.76.144.66
```

Achtung Unix Kenner: Ja keine Zeile der Sorte

```
domain nice.ch
```

einsetzen, sonst baut *sendmail* Mist und versucht Mail an **@nice.ch* über's nicht vorhandene Ethernet zu verschicken!

2.8 /etc/syslog.conf: Zusätzliche Zeile!

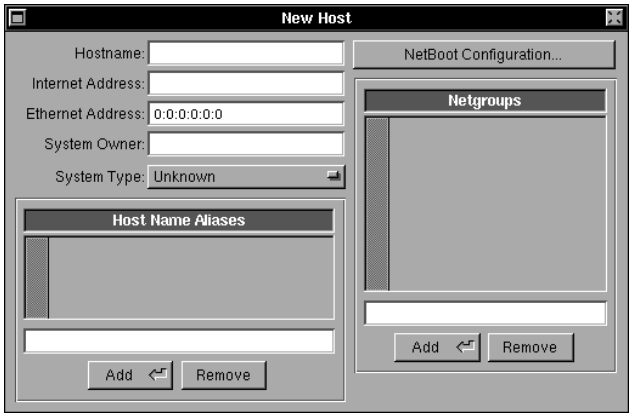
```
local3.debug    /usr/dialupip/log/syslog
```

3. Netinfo mit HostManager «erschlagen»

Alle NeXT's haben ein zentrales System-Utility namens *NetInfo* um Sachen wie Usernamen etc. zu verwalten. Leider hat NeXT, in der Absicht benutzerfreundlich zu sein, einen kapitalen Bock geschossen. Auf Systemen ohne Netz verwendet *NetInfo* automatisch eine eigene Datenbank für jede Maschine. Bei vernetzten Maschinen (LAN) wird dagegen nur eine einzige Datenbank für das ganze Netz verwendet und auf den Betrieb ohne lokale Datenbank umgestellt (dies ist bei allen Maschinen ausser einer ja auch der Fall). Bei einem über SLIP (WAN) angeschlossenen NeXT zuhause braucht jedoch wieder jede Maschine ihre Datenbank (sie ist quasi ein Einmaschinen-LAN). Man muss nun *NetInfo* so einstellen, dass trotz aktivem Netz eine eigene Datenbank verwendet wird.

Unglücklicherweise ist die Option hierfür an einem unerwarteten Platz versteckt (im Programm *HostManager*, welches für einen erfahrenen Unix Hacker aussieht wie ein GUI Editor zu den Dateien */etc/hostconfig* und */etc/hosts*). Von den 4 Monaten, die ich für die erfolgreiche Installation von SLIP aufgewendet habe, gingen allein 2 auf Kosten regelmässiger Abstürze wegen des falsch arbeitenden NetInfo-Systems verloren.

Dazu *HostManager.app* (in */NextAdmin*) starten und **Hosts** → **New** aufrufen und zwei neue Einträge generieren:



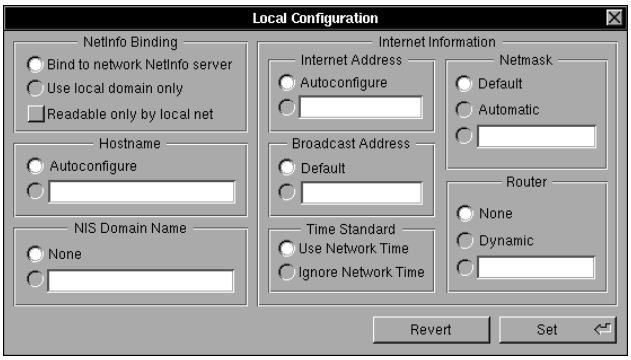
1. Eintrag:

```
Hostname          ###   (### = eigene Maschine)
Internet          ###.###.###.###
                  (### = eigene Adresse von EUnet)
Host Name Alias   ###.nice.ch
```

2. Eintrag:

```
Hostname          chsun
Internet          146.228.10.15
Host Name Alias   chsun.eunet.ch
                  mail.eunet.ch
```

Dann im Hauptmenu **Local...** aufrufen und ausfüllen:




```
Hostname            ###
Internet Address    ###.###.###.###
Broadcast Address   Default
Netmask             Default
```

Dies führt nach mehreren Abfragepanels (jeweils OK klicken) zur Generierung einer eigenen NetInfo-Datenbank. Am Schluss wird die Maschine neu gestartet.

Danach in der Datei */etc/hostconfig* die unerwünschten Seiteneffekte der Änderungen durch *HostManager* korrigieren:

```
HOSTNAME=###        (### = eigene Maschine)
INETADDR=-NO-
ROUTER=-NO-
IPNETMASK=-NO-
IPBROADCAST=-NO-
NETMASTER=-NO-
YPDOMAIN=-NO-
TIME=-NO-
```

(Überall `-NO-` weil wir kein Ethernet benutzen).
Zum Schluss noch in */etc/rc.local* nach der bestehenden Zeile:

```
# Run your own commands here
```

zusätzlich

```
sh /usr/dialupip/config/rc.slip>↵
/dev/console 2> &1
```

einfügen.

Root

Die Mail-Installation «Senden» (SMTP)

Um Mail über SLIP senden zu können muss man im Gegensatz zu UUCP 3 Schritte machen. Zuerst die von der Anwendung unabhängige SLIP-Verbindung (siehe oben) und dann die eigentliche Konfiguration der beiden Mail-Programme (für Senden und Empfangen).

/etc/sendmail/sendmail.cf
Als Basis wird die Datei *sendmail.mailhost.cf* verwendet. Dazu zuerst die Datei *sendmail.cf* (eigentlich ein Link) löschen, anschliessend *sendmail.mailhost.cf* in *sendmail.cf* kopieren. Die Datei sieht zwar äusserst wüst aus (vor allem gegen den Schluss), kann aber ohne Probleme geändert werden, so lange man sich genau an die Anleitung hält.
Folgende Zeilen sind zu ändern:

Alt	→	**Neu**
`DMuucp`	→	`DMddn`
`DR mail-relay`	→	`DRmail.eunet.ch`
`CR mail-relay`	→	`CRmail.eunet.ch`
`Odbackground`	→	`Odqueue`

Nach der Zeile `S1` einfügen:

```
# Username umbenennen
R$*    $:$>8$1
S8
Rneil  ###@dial.eunet.ch (###=POP Name gem. EUnet)
Rroot  ###@dial.eunet.ch
```

Bei der Zeile:

```
R$*<@$+>$*  $:$1<@$[$2$]>$3  find canonical
hostname
```

am Anfang ein # einfügen (um unnötige Telefonanrufe zu vermeiden). Bei der Zeile:

```
#R$*<@$*.$+>$*  $#$M   $@$R $:$1<@$2.$3>$4
user@any.domain
```

das # entfernen und bei der Zeile:

```
R$*<@$+.$->$*  $#ddn $@ $2.$3 $:$1<@$2.$3>$4
user@any.domain
```

am Anfang ein # einfügen (um Mail im Zweifelsfall an EUnet zu schicken).

Die Mail Installation «Empfangen» (POP)

1. Dateien auspacken und installieren

- *PopOver.1.1.NI.bd.tar.gz* auspacken → *PopOver.app*, *popOver*, beide nach */LocalApps* kopieren.

2. Dateien modifizieren
GUI Version *PopOver.app* starten.
Menu **Mail Hosts**…, dann **Add**…, dann eintragen:


```
Mail host:  mail.eunet.ch
Username:   ###     (###=POP Login Name)
Password:   ###     (###=POP Login Passwort)
```

Delete Mail aktivieren (sie wollen die Mails ja nur ein mal herunterholen). Nach OK entsteht im Home-Verzeichnis eine Datei *.popHosts.* Danach wird im Betrieb nur noch die Kommandozeilenversion *popOver* benützt.

3. Installation beenden
Maschine neu starten, damit die geänderten Dateien gelesen werden.

Mail testen

Danach testet man das Ganze, indem man eine Mail an die von EUnet gegebene POP-Mail Adresse schreibt und **Deliver** drückt. Bevor man sie absenden kann, muss die SLIP Verbindung zuerst mit *slipup* gestartet werden. In einer Shell gibt man dazu die beiden folgenden Zeilen ein:

```
/usr/dialupip/bin/slipup
/usr/lib/sendmail -q
```

Um die Mail zu holen (nach einer Wartezeit von ein paar Minuten) erneut mit

```
/usr/dialupip/bin/slipup
```

die Verbindung starten und anschliessend */LocalApps/popOver* verwenden, um die wartenden Mail's abzuholen.

Datentransfer automatisieren

Das beim obigen Testen verwendete Prozedere ist recht mühsam, weshalb man es am besten automatisiert. Idealerweise sollte anfallende Mail immer dann transferiert werden, wenn die SLIP Leitung ohnehin aktiv ist (z.B. für FTP oder Telnet), um kostenintensive Telefonverbindungen zu vermeiden und die «Zustellungshäufigkeit» der eigenen Mail zu steigern. Dazu modifiziert man */usr/dialupip/config/hooks.tcl*. Nach den Zeilen:

```
proc event_linkup {why} {
   global interface site
```

fügt man folgendes ein (### = Username):

```
exec /usr/lib/sendmail -q &
exec su ### -c /LocalApps/PopOver.app/popOver &
```

Danach muss ebenfalls neu gebootet werden damit die Änderung wirksam wird.

Will man nur einen Mailtransfer auslösen ohne sonstige Internet-Aktivitäten kann man dies einfach mit dem Starten von */usr/dialupip/bin/slipup* (ohne weiteren Befehle) explizit machen. Ich habe dazu eine Kopie unter dem Namen *DoMail.daemon* (Name historisch vom UUCP her) in mein Home-Verzeichnis getan.

NiCEmail anpassen

Wenn man SLIP so eingestellt hat gehen bereits alle Mail's über SLIP hinaus. Damit man aber auch NiCEmail empfangen kann muss auf der *nice* der NiCEmail-Eintrag angepasst werden. Am besten schickt man dazu die von EUnet erhaltene POP-Adresse zusammen mit einer kurzen Nachricht an den Mailadmin.

Fazit

Nach all dem sollte einem erfolgreichen Einsatz von SLIP nichts im Wege stehen. Ich benütze es seit Januar für Archie/FTP und seit Mitte Februar als meine einzige Verbindung (kein UUCP mehr).

Neil Franklin
Neil.Franklin@nice.ch

In eigener Sache

Traurig sieht es zur Zeit um die NiCE-Meetings aus! Auch nach dem Aufruf im Editorial der letzten Ausgabe hat sich niemand beim Vorstand gemeldet, der selbst ein NiCE-Meeting gestalten will bzw. auf Seite des Vereins für die Koordination zuständig sein will.

Wir müssen uns langsam ernsthaft fragen, ob wir die monatlichen NiCE-Meetings nicht auf längere Sicht abschaffen und nur noch eine talk&copy-Veranstaltung pro Monat durchführen wollen. Derzeit ist die Situation alles andere als befriedigend: Im ***PowerKey*** können sehr wohl die Daten der (möglichen) NiCE-Meetings angeben werden, nicht jedoch die Themen bzw. ob sie überhaupt durchgeführt werden. So ist der Vorstand gezwungen, die NiCE-Meetings fast schon routinemässig in talk&copy's umzuwandeln, was nicht nur uns unzufrieden stimmt, zumal das Interesse an zwei gleichen Veranstaltungen pro Monat stark nachzulassen scheint.

Es sei hier an dieser Stelle noch einmal betont: Wenn keine spezielle Ankündigung zu einem NiCE-Meeting erscheint – entweder per Mail oder per Postkarte – dann findet kein Meeting statt! Was dann jedoch bis auf weiteres automatisch stattfindet, ist ein talk&copy im IFW-Gebäude der ETH.

Aufruf:
Wer sich zum Thema NiCE-Meetings eine eigene Meinung leisten kann, der soll sie doch bitte dem Vorstand per Mail, Brief oder Fax mitteilen. Es würde uns stark interessieren was ihr zu dem Thema zu sagen habt.

(dm)

HP-Gecko Aktion

Für alle ETH-Angehörigen und immatrikulierte Studenten findet noch bis Ende Jahr eine Verkaufsaktion der HP Schweiz (nicht der NiCE!) statt.

Folgende drei Angebote stehen zur Auswahl:

NeXT
712/60 PA-RISC CPU (79 SPECfp92)
32 MB RAM (8 MB SIMM)
1050 MB internal Harddisk
3.5" 1.44 MB Floppy Drive
17" Color Monitor 1280×1024
Keyboard und Mouse
660 MB SCSI CD-ROM Drive
HP-UX Revision 9.0 (vorinstalliert)
Fr. 8499.- (Nettopreis ohne NEXTSTEP, inkl. Wust)

Oberon
gleiche Leistungsdaten wie NeXT, jedoch nur mit 16 MB RAM, 525 MB Harddisk und einem 17" Color Monitor (1024×768)
Fr. 6939.- (Nettopreis inkl. Wust)

Speedy
gleiche Leistungsdaten wie NeXT, jedoch mit 712/80 PA-RISC CPU (122 SPECfp92)
Fr. 11944.- (Nettopreis inkl. Wust)

Interessenten melden sich ETH intern bei:

Herrn M. Gablinger
RZ G 8.2, int. 5801
gabling@ks.id.ethz.ch

oder bei:

GSE
Schuppistrasse 2a
8042 Zürich
Tel. 01/312 37 00
Fax 01/312 37 46

Eine Demo-Maschine steht im E-Stock des RZ (anschliessend ans IFW-Gebäude, Adresse siehe Veranstaltungskalender).

Testbericht: Althys Virtuoso

Die DTP-Revolution konnte in dieser Form nur einmal stattfinden. Sie ist mehr als ein Jahrzehnt alt und mittlerweile untrennbar mit dem Macintosh verknüpft. Das mag man bedauern oder nicht, deutlich wird diese Tatsache nach einem kurzen Blick in Satzstudios, Druckereien, Werbeagenturen und Grafikateliers weltweit: «Mac all over the place».

NeXT und Grafik — die Situation

NeXT trägt seinen Namen wohl unter anderem, weil Steve Jobs überzeugt war, ein Schritt weiter in technologischer Richtung würde automatisch diese riesige Industrie, genannt grafisches Gewerbe, folgen lassen. Betrachtet man diesen Gedanken nachträglich im Bewusstsein der tatsächlichen Entwicklung, muss man von Naivität, vielleicht sogar von Grössenwahn sprechen. Es sah so aus, als könnte die Sache funktionieren, solange NeXT tatsächlich nichts war als eine technische Verbesserung des Mac. Leidgeprüfte Pioniere, die versucht haben, mit ihren schwarz glänzenden Schuhen das graue Mainland des DTP zu betreten, wurden immer wieder eines schlechteren belehrt: Nicht nur ist NeXT ein absoluter Nobody im Markt, die vielgepriesene Fähigkeit, DOS- und Mac-Formate zu lesen und zu schreiben bringt auch wesentlich weniger Vorteile als erhofft. Nur schon die Belichtung von Grafiken, die am NeXT entstanden, ist immer wieder ein Abenteuer, das von unschönen Überraschungen wimmelt. Mal kommt der Film anstelle der deutschen Umlaute mit exotischen Satzzeichen aus dem Satzstudio zurück, mal zeigt sich das maclastige Entwicklungsbüro ausserstande, eine simple Illustrator-Grafik zu öffnen, um eine letzte winzige Korrektur anzubringen.

Zusätzlich macht sich schmerzhaft bemerkbar, dass die Entwicklung von NeXT-Software für den Grafikbereich offensichtlich fast vollkommen stagniert. *Illustrator* ist seit der Version 3.0 (der ersten auf NEXTSTEP) nicht mehr weiter verbessert worden, obwohl man auf Mac mittlerweile eine stolzgeschwellte 5 präsentiert, die all das kann, was NeXT kann, und… mehr!

Application Icon von Virtuoso

In diese Situation hinein kam vor einiger Zeit *Virtuoso*, gemacht von Althys, den Eltern des Illustrator-Erzfeindes Freehand. Virtuoso kann Illustrator, EPS, Freehand sowie ein eigenes Format schreiben und lesen, ist also vielversprechend in Bezug auf Anschlussmöglichkeiten. Aber was für ein Programm ist Virtuoso eigentlich genau?

Auf dem Weg zum Hybrid

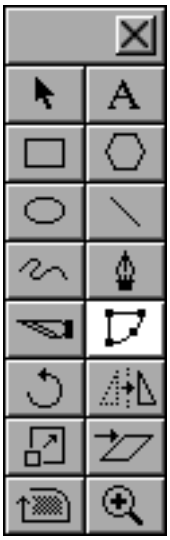
Virtuoso tritt, zumindest in seiner neuesten Version, eindeutig aus dem Schatten seines Rivalen Illustrator, denn es durchstösst die Grenze zwischen Vektorgrafikprogramm und Seitenlayoutapplikation. (Freehand tut übrigens gleichzeitig dasselbe auf der Mac-Plattform.) Durch die Fähigkeit, mehrseitige Dokumente zu erstellen, durch die gute Geschwindigkeit beim Erfassen von Text und durch die reibungslosen Importmöglichkeiten für TIFF, EPS und andere Grafiken, durch variable Hilfslinien und ein durchdachtes Layer-Konzept kombiniert Virtuoso die Möglichkeiten eines Zeichenpakets mit dem (fast kompletten) Komfort eines Satzprogramms im Stil von Pagemaker, XPress etc.

Aufbau

Virtuoso hat eigentlich zwei «Herz-Elemente». Da ist zunächst einmal die bewährte Werkzeugkiste, aus der diverse Tools gepflückt werden können. Gute Idee: Das geht nicht nur per Maus, sondern auch per Tastenklick. Ein schlichtes A zum Beispiel macht das Text-Werkzeug zugänglich.

Daneben gibt es ein zentrales Inspector-Panel, das aus sechs Unterabteilungen besteht:

- Positionsabteilung zum Ablesen und numerischen Eingeben der genauen Lage, Höhe und Breite sämtlicher Elemente, auch Textblocks oder importierte TIFF's etc.
- Layer-Werkzeug zum Kreieren, Benennen und Umverteilen beliebig vieler Schichten. Dazu später mehr.
- Ein Fill-Werkzeug, das z.T. in Zusammenarbeit mit NEXTSTEP-typischen Elementen wie dem Colorwell das Bestimmen zahlreicher Füllungen erlaubt. Dazu gehören auch diverse vorgefertigte Muster und die beliebten Verläufe.
- Ein Line-Werkzeug, das ähnlich funktioniert, sich aber auf die Umrandungslinien von Objekten bezieht.
- Ein Satz-Werkzeug, das eine Fülle von Textmanipulationen erlaubt, sehr schön übersichtlich angeordnet, und zwar nach Prioritäten, wie sie für Profis in Frage kommen (die berüchtigten Schlagschatten und andere Monstereffekte macht Virtuoso auch, aber sozusagen nur, wenn es unbedingt sein muss. Es verlangt von seinen Mit-Virtuosen, sich erst mal um das Grundlegende zu kümmern.).
- Ein Dokumenten-Tool, das sich um Masseinheiten, Seitenformat, Seitenzahl und -reihenfolge, also die grundlegenden Einstellungen des Files kümmert.



Toolbox

Das klingt zunächst komplex, besonders wenn man merkt, dass all diese Abteilungen wiederum Unterabteilungen haben. Aber die Sache ist sehr gut durchdacht und offensichtlich auf die Bedürfnisse von lebendigen Menschen ausgerichtet, und so funktioniert der Gebrauch des Inspektors nach kurzer Eingewöhnungszeit automatisch. Anfangs sucht man noch gelegentlich die Menüs nach Dingen ab, die man nach Erfahrungen mit anderen Programmen dort vermutet (z.B. das Seitenlayout), aber bald hat man alles sehr gut im Griff.

Schichtwechsel

Was hat es nun mit den Layers auf sich? Ein Grafikdokument, besonders eine Illustration, besteht immer aus Schichten von Elementen, die übereinanderliegen, deren Reihenfolge also entscheidend dafür ist, was man nachher sieht. Je nach «Vielschichtigkeit» (das Wort sagt alles…) kann es äusserst schwierig werden, diese Schichten zu managen. Vor allem im fortgeschrittenen Zustand würde man oft gern eine ganze Gruppe von Elementen in ihrer Position ändern, ausblenden oder deren Attribute versuchsweise abändern. Wenn man in Virtuoso von anfang an konsequent mit den Layers arbeitet, und jedes neue Element einem solchen Layer zuteilt, wird all das zum Kinderspiel. Durch die Möglichkeit, die Layer zu benennen (zum Beispiel Fotos, Rahmen, Textblocks, Bildunterschriften, Hintergrund, Hilfslinien etc.), bleiben auch komplexe Dokumente transparent und leicht zu managen. Selbstverständlich kann man auch Layer löschen oder nach belieben während der Arbeit neue kreieren. Auch Illustrator kann ausblenden und «locken» (verriegeln), aber wenn man diese Funktionen rückgängig macht, wird ohne Unterschied alles wieder eingeblendet oder entriegelt, was jemals von der Bildfläche gezaubert oder festgenagelt wurde. In Virtuoso kann man alle Layer für sich betrachten und bearbeiten. Zusätzlich zur Arbeitserleichterung, die das bringt, lässt sich so die Geschwindigkeit erheblich verbessern, indem man z. B. Fotos nur dann sichtbar behält, wenn man sie wirklich braucht (schnellerer Bildschirmaufbau).

Illustrator verschrotten?

Nach so viel Lob könnte man meinen, der alte Illustrator gehöre komplett zum alten Eisen. Nicht ganz! Gerade durch seine Schlichtheit ist mir das Adobe-

Produkt, zumindest in der Entwurfsphase, viel lieber als der Herausforderer. Das Handling der berühmten Bezierkurven ist nach wie vor unerreicht. Und Illustrator bleibt die erste Wahl für Künstlerinnen und Illustratoren, die möglichst alles selbst machen wollen. Ideal ist eine Kombination aus beiden Programmen, besonders, weil auch Virtuoso die oben beschriebenen Belichtungsprobleme kennt. Ein kleiner Test (kann ein beliebiges DTP-Satzstudio ein Virtuosofile, gespeichert auf Mac-Diskette als Freehand-Dokument, öffnen?) ging mir schon prompt in die Hose. Da ist man froh, wenn man auf die mühsam erarbeitete Illustrator-Form zurückgreifen kann. Klar: Features, die Illustrator nicht hat, gehen bei der Umwandlung des Formates verloren, zum Beispiel die weiter oben gerühmte Layertechnik. Also: Unbedingt Originalfile behalten und erst umwandeln, wenn alles stimmt!

Johannes Labusch

NiCE-Markt

Ich verkaufe ab sofort:

- WordPerfect 1.0.1 for NEXTSTEP Fr. 400.-

(Intel+Motorola, originalverpackt)

Martin Beerli
Büchnerstr. 19
8006 Zürich

Tel.: 01/362 95 72 (tagsüber)
01/362 76 71 (18–20 Uhr)

Oberon V4 – Beta zum Schnuppern

Vor etwa eineinhalb Jahren nahmen Adriano Gabaglio und ich eine Semesterarbeit an der ETH in Angriff, um *Oberon* – damals noch Version 2 und etwas mehr – auf schwarze Hardware zu portieren. In der Zwischenzeit habe ich V4 integriert und auf Basis des Windows-Compilers eine Version für PCs gemacht (für Gecko-User: Der HP-Port würde nicht viel kosten, da das ganze schon ausgemistet und *bi-endian* ist; falls Interesse da ist, kontaktiert mich).

Wer sich jetzt fragt, was Oberon überhaupt ist, hier die Antwort: Oberon ist der Name eines ETH-Projektes hinter dem hauptsächlich die Proff. Wirth und Gutknecht stehen. Ziel war es, ein schlankes und portables Workstation-Betriebssystem zu schaffen, das sich speziell für den Unterricht eignet. Neben dem Betriebssystem Oberon war die Sprache Oberon ein weiteres Resultat: Diese Sprache (in unserem Fall *Oberon-2*) ist eine Weiterentwicklung von Modula-2 (und somit auch indirekt von Pascal) die mit objektorientierten Konzepten und der Fähigkeit Low-Level-Module zu schreiben angereichert wurde. Ursprünglich wurde Oberon auf der *Ceres* implementiert. Das ist eine Workstation auf Basis des NS32xxx-Prozessors die ebenfalls an der ETH entwickelt wurde. Mittlerweile läuft Oberon als eingebettetes System auf fast allen Rechnern von Rang und Namen: Ceres (Native, NS32k), Macintosh, DOS, Windows, Solaris (Sparc), Linux, Silicon Graphics, Decstation (Mips/Ultrix), RS/6000 (AIX), HP 9000/700 (HP/UX), Amiga und Next.
Wer sich für System und Sprache interessiert – oder das Vergnügen hat Übungen damit zu schreiben – kann sich eine Beta-Version von Oberon V4 vom NiCE-Server runterkopieren (*~ftp/NiCE*). Wie das ganze läuft, welche Macken es auf schwarzer und auf weisser Hardware hat, steht im *Readme*-File. Wer Oberon noch nicht kennt muss sich durch die empfohlenen Berichte kämpfen oder sich eins der Bücher besorgen.

George Fankhauser
<gfankhau@iiic.ethz.ch>

Was sie schon immer über TeX wissen wollten – Teil 6

Eine wichtige Arbeitserleichterung bei der Erstellung von TeX-Dokumenten ist die Abkürzung von häufig eingegebenen Textstücken durch Makros. Doch Makros bieten weit mehr als nur Textersetzung – sie ermöglichen es, neue Befehle zur Textsetzung zu schaffen, um so die vielfältigen Einsatzmöglichkeiten von TeX noch zu erweitern.

Einfache Makros

Wenn in einem Text zum Beispiel häufig der Vektor $(x_1, \ldots, x_n)$ gesetzt werden muss, ist es schon sehr lästig, stets `$(x_1,\ldots,x_n)$` einzugeben. Der Makrobefehl `\def` hilft dabei:

```
\def\XV{(x_1,\ldots,x_n)}
```

bewirkt, dass `\XV` die Abkürzung des längeren Ausdrucks ist. Wenn `\XV` in der Eingabe gefunden wird, expandiert TeX dies zu der dazugehörenden Zeichenfolge `(x_1,\ldots,x_n)`.

Die Wirkung ist genauso, wie wenn diese Zeichen im Eingabetext stehen würden. Die geschweiften Klammern, die um den Definitionsteil herumstehen, werden jedoch nicht eingesetzt. Nur die zur Definition nötigen äusseren Klammern werden entfernt! Zwischen den beiden Definitionen

```
\def\zb{\bf zum Beispiel}
\def\ZB{{\bf zum Beispiel}}
```

besteht somit der wichtige Unterschied, dass im ersten Fall auch noch nach dem Text «*zum Beispiel*» weiter in der Schriftart `\bf` (boldface) gesetzt wird. Der Aufruf `\zb` wird zu `\bf zum Beispiel` expandiert, `\ZB` dagegen zu `{\bf zum Beispiel}`.

Makros mit Parametern

Am häufigsten werden Makros verwendet, die Parameter besitzen. Will man ein allgemeines Makro für *x,y,z*-Vektoren aus den obigen Beispiel, so empfiehlt sich:

```
\def\vector#1{(#1_1,\ldots,#1_n)}
```

Dann liefert der Aufruf `$\vector y$` als Ergebnis

$$(y_1, \ldots, y_n)$$

Makros dürfen bis zu 9 Parametern besitzen. Diese müssen aufsteigend mit `#1`, `#2`, `#3`, …, `#9` bezeichnet werden. Beim Aufruf eines Makros muss TeX entscheiden, wieviel des nachfolgenden Textes jetzt für die einzelnen Parametern übernommen werden muss. Wird ein Makro ohne spezielle Trennzeichen definiert, was der Normalfall ist, so übernimmt TeX immer das nächste *token*. Dies kann aber auch wieder der Aufruf eines Makros sein. Ein Makro

```
\def\test#1#2#3{#1#1#2#2#3#3}
```

das mit `\test abcde` im Eingabetext aufgerufen wird, übernimmt für

```
#1  ←  a
#2  ←  b
#3  ←  c
```

Das Ergebnis des Makroaufrufes und des nachfolgenden Textes entspricht also der Eingabe *aabbccde*. Die Zeichen ‘de’ gehen als normale Zeichen an TeX weiter. Wird zusätzlich noch ein Befehl

```
\def\abc{Alphabet}
```

definiert, bewirkt der Befehl `\test\abc abc` eine Parameterbesetzung:

```
#1  ←  \abc
#2  ←  a
#3  ←  b
```

Anschliessend wird in der weiteren Abarbeitung weiter expandiert:

```
#1  ←  \abc  ←  Alphabet
```

Also wird die gesamte Eingabe zu der Zeichenfolge *AlphabetAlphabetaabbc*.

Will man mehr als ein Zeichen oder einen Befehl in einem Parameter übergeben, so ist zu klammern. Der Aufruf `\test a{bcde}{fg}hij` ermittelt die drei Parameter zu

```
#1  ←  a
#2  ←  bcde
#3  ←  fg
```

Die Zeichen ‘hij’ sind Text der normalen Eingabe und werden nicht durch das Makro behandelt.

Nun kann man bei der Definition eines Makros auch selbst angeben, durch welche Zeichen die einzelnen Parameter getrennt werden sollen. Die Definition

```
\def\TEST #1 #2 #3 {#1#1#2#2#3#3}
```

für `\TEST` unterscheidet sich von `\test` darin, dass die einzelnen Parameter durch Leerzeichen getrennt werden müssen. Der `\test` entsprechende Aufruf für das letzte Beispiel lautet somit

```
\TEST a bcde fg hij
```

Bei der Definition des trennenden Textes braucht man sich aber nicht auf Leerzeichen zu beschränken. Durch

```
\def\restbold#1.{{\bf#1.}}
```

wird der nachfolgende Text bis zum nächsten Punkt als Parameter übernommen. Allerdings ist zu bemerken, dass der Trenntext ausserhalb von Klammerstrukturen stehen muss. Bei der Benutzung in folgendem Text

```
\restbold Ein Text {soll … hier} folgen.
```

wird für den ersten (und einzigen) Parameter

```
#1  ←  Ein Text soll … hier folgen
```

eingesetzt. Der Trenntext (der abschliessende Punkt) gehört nicht zum Parameter!
Auch solche Strukturen sind möglich:

```
\def\meintest #1ABC#2.#3${…}
```

Die Parameter bestehen dann aus aus dem Text bis zum ersten Auftreten von ‘ABC’. Der zweite Parameter besteht aus den Zeichen zwischen ‘ABC’ und dem nächsten Punkt. Der dritte aus den Zeichen zwischen Punkt und dem nächsten Dollar.

Makros innerhalb von Makros

Innerhalb von Makros können selbst wieder neue Makros definiert werden. Ob diese dann nach der Abarbeitung des äusseren Makros noch bekannt sind, hängt von der Blockstruktur ab. Beliebt ist es, lokale Hilfsmakros zu definieren, die hinterher nicht mehr vorhanden sind.
Bei der Definition von Makros innerhalb anderer ist auf die Verwendung von ‘#’ zu achten. Es ist für innere Makroparameter eines neuen internen Makros ein zusätzliches ‘#’ zu setzen. Beispiel:

```
\def\MeineMatrix#1{{\def\vektor##1{#1_##
1,\ldots,#1_n}$$\pmatrix{\vektor1\cr
\vektor2\cr \vektor3\cr}$$}}
```

Der Aufruf `\MeineMatrix{\sin\alpha}` liefert:

$$\begin{pmatrix} \sin\alpha_1,\ldots,\sin\alpha_n \\ \sin\alpha_2,\ldots,\sin\alpha_n \\ \sin\alpha_3,\ldots,\sin\alpha_n \end{pmatrix}$$

Soll jedoch umgekehrt zum bisherigen Verhalten ein Makro, das innerhalb einer inneren Klammerstruktur definiert wird, auch nach Verlassen derselben bekannt sein, muss man ein `\global` davor setzen. Beispiel:

```
{{{{\global\def\ABC{abcdefghijklmnopqrst
uvwxyz}}}}
```

Das Makro `\ABC` ist auch nach Abarbeitung des Klammergebirges bekannt. Das gleiche gilt für

```
\def\initABC{{\global\def\ABC{abcdefghij
klmnopqrstuvwxyz}}
```

Nach dem Aufruf von `\initABC` ist das Makro `\ABC` global bekannt.

Will man die aktuelle Bedeutung irgendeines Befehls oder Makros speichern oder einen Befehl umbenennen, hilft der `\let`-Befehl weiter. Durch

```
\let\INITABC=\initABC
\def\initABC{{\global\def\ABC{abcdefghij
klmnopqrstuvwxyz}}
```

stehen anschliessend zwei Befehle zur Verfügung: `\INITABC` mit der alten Bedeutung von `\initABC` und ein neues `\initABC`.

Expandieren von Makrobefehlen

Bei der Eingabe eines Makros werden die im Definitionsteil stehenden Befehle nur gespeichert. Sie werden bis auf ganz wenige Ausnahmen noch nicht interpretiert. Es wird auch noch nicht geprüft, ob diese Befehle überhaupt existieren. Erst zur Ausführungszeit werden die Eingabebefehle der Reihe nach ausgewertet. Dies bedeutet insbesondere, dass diese auch mit ihrer zu diesem Zeitpunkt gerade gültigen Bedeutung verwendet werden.

Allerdings bietet der Befehl `\edef` (für *expanded definition*) nun gerade die Möglichkeit, den Inhalt eines Makros schon bei der Definition expandieren zu lassen. Damit wird die aktuelle Bedeutung zum Zeitpunkt der Definition ausgewertet und als Definitionstext eingetragen. Jedoch müssen alle zu expandierenden Makros zu diesem Zeitpunkt bereits bekannt sein. Beispiel:

```
\def\text{ALT}
\edef\etest{\text}
\def\test{\text}
\def\text{NEU}
```

Danach werden die Aufrufe wie folgt ausgewertet:

```
\text  →  NEU
\test  →  NEU
\etest →  ALT
```

Nun ist es aber bisweilen so, dass ein Teil der Befehle sofort, der andere aber erst später expandiert werden soll. Dazu gibt es den Steuerbefehl `\noexpand`, der die Interpretation des folgenden Befehls unterdrückt. Die Wirkungsweise sei an dem etwas abgewandelten letzten Beispiel dargestellt:

```
\def\text{ALT}
\edef\etest{\text, \noexpand\text}
\def\text{NEU}
```

Dies führt zur Ausgabe:

```
\text  →  NEU
\etest →  ALT, NEU
```

Auch dem Befehl `\edef` kann ein `\global` vorangestellt werden. Für die Befehlsfolge `\gobal\edef` gibt es den abkürzenden Befehl `\xdef`; die Befehlsfolge `\global\def` lässt sich mit `\gdef` kürzer schreiben. Noch zwei weitere Befehls zur Definition von Makros sind vorhanden. Mit `\outer` wird ein Makro charakterisiert, dass nur aus dem äussersten Eingabeniveau aufgerufen werden darf. Damit werden Aufrufe in Makros, als Makroparameter oder in Boxen als Fehler interpretiert. Es typisches Beispiel für einen solchen Befehl ist das `\bye`-Kommando.
Beim Aufruf eines Makros sind im Normalfall nur kurze Parameter erlaubt. So werden z.B. keine ganzen Absätze akzeptiert, es sei denn bei der Definition wurde dies mit `\long\def`… angegeben.

Abfragen

Die bisherige Abarbeitung eines Makros war recht statisch: Stets wurde bei gleichem Aufruf auch ein identisches Ergebnis erzeugt. Häufig möchte man jedoch abhängig von äusseren Bedingungen unterschiedliche Verhaltensweisen erzeugen. Ein Beispiel ist eine Defi-

nition der Seitenüberschrift, die für gerade und ungerade Seitennummern verschiedene Ergebnisse erzeugt. Dazu gibt es Abfragebefehle mit der Syntax:

`\if`<*Bedingung*><*true-Teil*>`\else`<*false-Teil*>`\fi`

Falls die Bedingung erfüllt ist, werden nur die Befehl des «true-Teils» abgearbeitet, sonst nur die Befehle des «false-Teils». Wichtig ist dabei, dass die Makros, die in einem der beiden Teile stehen, nur dann expandiert werden, wenn dieser auch wirklich durchlaufen wird.

Es gibt eine Reihe verschiedener Bedingungen, die verwendet werden können. Die Folge `\if`<*Bedingung*> gilt jeweils als einzelner TeX-Befehl. Zwei verschiedene Abfragetypen sind vorhanden: der Grössenvergleich und die Abfrage eines bestimmten Zustandes.

Die folgenden sechs Abfragen prüfen, ob die linke Zahl bzw. Dimension gleich, grösser oder kleiner der rechten ist.

```
\ifnum zahl1 = zahl2
\ifdim dimension1 = dimension2
\ifnum zahl1 > zahl2
\ifdim dimension1 > dimension2
\ifnum zahl1 < zahl2
\ifdim dimension1 < dimension2
```

Beispiele:

```
\ifnum\pageno=1 …
\ifnum 17>\count0 …
\ifdim\leftskip>\hsize …
\ifdim\ht0>1cm …
```

Ein Anwendungsbeispiel: Die Seitennumerierung soll für die erste Seite nicht ausgegeben werden.

```
\footline={\ifnum\pageno=1\hss\else\hss
\folio\hss\fi}
```

Eine zweite Gruppe enthält Zustands- und Vergleichsabfragen, mit denen insbesondere die Identität von Objekten geprüft werden kann. `\ifodd` prüft, ob die nachfolgende Zahlenangabe ungerade ist. `\if` prüft, ob die nächsten 2 Zeichen übereinstimmen. (Dabei werden evtl. folgende Makros expandiert!) `\ifx` prüft ohne volle Expandierung ob die beiden folgenden Sequenzen übereinstimmen. Damit können Makrodefinitionen verglichen werden.

Eine beliebte Anwendung ist die oben erwähnte unterschiedliche Seitennumerierung: Die folgende Anweisungsfolge bewirkt bei ungeraden Seitenzahlen eine rechtsbündige und bei geraden Seitenzahlen eine linksbündige Seitennummer:

```
\footline={\rm\ifodd\pageno\folio\hss
\else\hss\folio\fi}
```

Wechselnde Seitenüberschriften realisiert man am sinnvollsten durch

```
\def\linkertitel{\it\hfill Autor \hfill}
\def\rechtertitel{\it\hfill Titel
\hfill}
\headline={\rm\ifodd\rechtertitel\else
\linkertitel\fi}
```

Auf den Seiten mit ungerader Seitenzahl steht jeweils «Titel» als zentrierte Seitenüberschrift, auf den anderen Seiten entsprechend «Autor».

Ein weiterer Anwendungsfall ist die Prüfung, ob ein Makroparameter leer ist. Dazu wird zunächst ein leeres Makro `\empty` definiert, gegen das geprüft wird. (Die Definition von `\empty` kann auch entfallen, sie ist in plain-TeX vorhanden.) Innerhalb der Definition muss noch ein zweites Vergleichsmakro – hier `\test` – als Gegenstück definiert werden. Anschliessend können diese beiden mit Hilfe von `\ifx` miteinander verglichen werden.

```
\def\empty{}
\def\meinMakro#1#2#3{\def\test{#1}
\ifx\test\empty … \else … \fi}
```

Wird nun z.B. `\meinMakro{}{ABC}{DEF}` aufgerufen, expandiert der Parameter `#1` zum (leeren) Innern von `\def\test{}`. Dies stimmt aber mit dem Definitionsteil von `\empty` überein. Hingegen wird beim Aufruf von `\meinMakro{eins}{zwei}{drei}`

das lokale Prüfmakro zu `\def\test{eins}`; daher sind beim Vergleich `{}` und `{eins}` verschieden.

Es folgen noch einige Befehle, die prüfen in welchem der verschiedenen TeX-Arbeitsmodi man sich gerade befindet. So kann innerhalb eines Makros entschieden werden, ob es innerhalb einer mathematischen Formel oder im Fliesstext aufgerufen wurde. Der Befehl dazu lautet `\ifxmode`, wobei das '`x`' durch '`v`' für *vertical mode*, '`h`' für *horizontal mode*, '`m`' für *mathematical mode* oder '`i`' für *internal mode* zu ersetzen ist. (Die verschiedene Arbeitsmodi werden in der nächsten Folge des TeX-Kurses eingehend behandelt werden.)

Ein Beispiel: Der Befehl `\stars` soll unabhängig von Text- oder Mathematikmodus drei Sterne '***' ausgeben. Dies leistet die folgende Definition:

```
\def\stars\ifmmode{\star}{\star}{\star}
\else${\star}{\star}{\star}$\fi}
```

Eigene if-Befehle

Zur Abfrage und Erzeugung eigener `if`-Befehle sind drei Befehle vorhanden, die diese Konstruktionen unterstützen. Zunächst sind dies die beiden Befehle `\iftrue` und `\iffalse`. Diese liefern beim Aufruf stets den Wert *true* bzw. *false*. Dies scheint für eine Abfrage zunächst überhaupt nicht sinnvoll. Durch eine Kombination mit dem Befehl `\newif` werden diese beiden Konstrukte aber sehr brauchbar.
Beispielsweise werden nach `\newif\ifvorwort` drei neue Makros erzeugt. Diese sind: `\ifvorwort` zur Abfrage, `\vorworttrue` zum Setzen von *true* und `\vorwortfalse` zum Setzen von *false*. Genauer betrachtet wird im Fall der beiden letzten Makros `\ifvorwort` zu einem `\iftrue` bzw. `\iffalse`.

Nach der Deklaration ist der Zustand *false* voreingestellt. Damit kann in der Anwendung eine schöne und vor allem lesbare Abfrage formuliert werden:

```
\ifvorwort … \else … \fi
```

Trick-Makros für Fortgeschrittene

Der `\afterassignment` folgende Befehl wird nicht sofort ausgeführt, sondern gespeichert und erst nach der nächsten Zuweisung ausgeführt.
Als Beispiel soll ein Befehl zum «Aussperren» von Wörtern dargestellt werden, also zur Vergrösserung des Buchstabenzwischenraums innerhalb eines Wortes. Hier liegt das Problem darin, dass unbekannt ist, wie viele Buchstaben dem Befehl folgen. Nach der Befehlsfolge `\sperr{Aussperren}` liefert der Aufruf folgenden Output:

A u s s p e r r e n

```
\def\sperr#1{\SperrRest#1\endlist}
\def\endlist{\endlist}
\def\SperrRest{\afterassignment
\SperrZeichen\let\next= }
\def\SperrZeichen{%
   \ifx\next\endlist
      \let\next\relax\kern-0.25em
   \else
      \next\kern0.25em
      \let\next\SperrRest
   \fi
\next}
```

Ausblick

Diese Folge des TeX-Kurses ist etwas umfangreicher ausgefallen als ich es vorgesehen hatte. Aus diesem Grund wird die für diese Ausgabe geplante Vorstellung der Arbeitsweise von TeX auf die nächste Folge verschoben. Bis dann…

Dominik Moser

Do you speek «C»?

Mal ehrlich: Könnt Ihr mit diesen kryptischen Deklarationen in C etwas anfangen? Sind Euch Ausdrücke wie `(int (*)())x` geläufig, oder könnt Ihr gar Zungenbrecher wie *«cast x into pointer to function(pointer to char) returning int»* in die entsprechende C-Deklaration übersetzen? Wenn ja, dann gehört Ihr zu der beneidenswerten Spezies von fortgeschrittenen C-Programmieren. Wenn nicht, kann Euch jetzt vielleicht geholfen werden.

Wie sag' ich's meinem Compiler

Nun ja, man wird wohl auf keiner Party gross Eindruck erwecken, wenn man fliessend C spricht – es soll allerdings auch *solche* Parties geben. Und zu einer Einladung in Thomas Gottschalk's *Wetten dass…?* wird es wahrscheinlich auch nicht reichen. Das im folgenden beschriebene Prográmmchen gehört in die Kategorie «Funny Tools», zu denen auch so geistreiche :-) Programme wie *Eyecon* oder *Fortune* gehören.

Gefunden habe ich das Kleinod namens *cdecl* bei einer Archie-Session, um die 20.- DM zu sparen, die ein gieriger Verteiler in Deutschland dafür haben wollte. Schwubs heruntergeladen, entzippt und entpackt, offenbarten sich mir die folgenden Files:

```
 9817 Jan  2  1991 cdecl.1
25573 Jun  2  1991 cdecl.c
22746 Jan  2  1991 cdgram.y
 2380 Jan  2  1991 cdlex.l
 1340 Jun  2  1991 makefile
 1038 Jan  2  1991 testset
 1337 Jan  2  1991 testset++
```

Nach den Anwerfen von *make* bekommt man schliesslich zwei ausführbare Programme namens *cdecl* und *c++decl*. In der Folge werden ich mich, nicht zuletzt wegen gewissen Abneigungen gegen gehäuftes Auftreten von Pluszeichen, auf ersteres konzentrieren.

Da ist es also nun in seiner «gestripten» Grösse von gut 49 KByte und verspricht uns einen C-Guru im Westentaschenformat! Ein schüchternes Klopfen an die Pforten der Weisheit verschafft uns Zugang:

```
> cdecl
Type 'help' or '?' for help
cdecl>
```

Tja, mal sehn':

```
cdecl> help
…
  command:
     declare <name> as <english>
     cast <name> into <english>
     explain <gibberish>
     set or set options
     help, ?
     quit or exit
…
```

gibberish?

> **gib•ber•ish** \'jib-(e-)rish, 'gib-\ n
> [prob. fr. gibber]
> (1554)
> :unintelligible or meaningless language:
> a: a technical or esoteric language
> b: pretentious or needlessly obscure language

Danke Webster! Na wenn das so ist:

```
cdecl> explain jabba dabba doo
syntax error
```

Spielverderber! Halt – alles zurück und den Anfang noch einmal lesen… Also es geht um C und um Dinge wie `(int (*)())x`. Jetzt wollen wir doch einmal sehen:

```
cdecl> explain (int (*)())x
cast x into pointer to function
returning int
```

Heureka!

Programmbeschreibung

Cdecl ist ein Programm zur Umwandlung von C (oder C++) Typdeklarationen in pseudoenglischen Text und zurück. Die Defaultsprache von *cdecl* ist C, basierend auf dem X3J11 ANSI Standard. Optional kann die Sprachdefinition gemäss Kernighan & Ritchie's Buch *The C Programming Language* (Option `-p`) oder Ritchie's PDP-11 C-Compiler (Option `-r`) verwendet werden. Durch die Option `-+` (bzw. einen Link mit Namen *c++decl*) kann alternativ dazu die C++ Sprachdefinition gemäss Stroustrup's *The C++ Programming Language* verwendet werden.

Cdecl lässt sich während der Erstellung eines C-Programms aus Editoren wie *vi* und *emacs* direkt aufrufen. Einfach die gewünschte Deklaration in Pseudoenglisch eingeben und *cdecl* als Filter aufrufen.

Kommandosprache

Die Kommandosprache umfasst sechs Anweisungen. Durch *declare* wird eine pseudoenglische Zeichenkette in eine C Typdeklaration umgewandelt. Der Befehl *cast* erzeugt aus der Zeichenkette einen sogenannten cast, also eine implizite Typumwandlung. Mittels *explain* werden Typdeklarationen oder Typumwandlungen in eine Textbeschreibung umgewandelt. Durch *help* (oder *?*) erhält man eine kurze Programmbeschreibung, während *quit* (oder *exit*) zur Beendung des Programms führen. Schliesslich kann man mit *set* die Optionen von *cdecl* interaktiv verändern.

Beispiele

Um einen Array bestehend aus Zeigern auf Funktionen ähnlich *malloc()* zu deklarieren, tippt man

```
declare fptab as array of pointer to
function returning pointer to char
```

und erhält

```
char *(*fptab[])()
```

Falls man in einem fremden Programm die folgende Deklaration findet, kann man sie sich von *cdecl* erklären lassen:

```
explain void (*signal(void))()
```

liefert

```
declare signal as function returning
pointer to function returning void
```

Vielleicht wundert sich der Programmierer, wo er bei einer solchen Funktion die Parameter unterbringen soll

```
declare signal as function (arg1,arg2)
returning pointer to function returning
void
```

ergibt mit der Option `-c`

```
void (*signal(arg1,arg2))() { }
```

Cdecl kann auch bei der richtigen Verwendung von *const* hilfreich sein:

```
declare foo as pointer to const int
```

liefert

```
const int *foo
```

während

```
declare foo as const pointer to int
```

einen (im Sinne von C) völlig anderen Output erzeugt:

```
int * const foo
```

Fazit

Cdecl ist ein witziges kleines Prográmmchen, das dem C-Programmierer gute Dienste bei der fehlerfreien Deklarationen von komplizierten Ausdrücken leisten kann. Leider wird Objective-C zur Zeit noch nicht unterstützt.

(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, Johannes Labusch, George Fankhauser
"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 bzw. Kassier.

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 gewerbsmässige Inserate von Mitgliedern gratis.

NiCE – NeXT User Group

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

PC-Konto: 80-46102-05

Vorstand

PowerKey 4/94 **erscheint Ende Dezember 1994**
Redaktions- und Anzeigenschluss: 10. Dezember 1994