

NICK

Netzwerkweites Informationssystem auf dem Campus der Universität Karlsruhe

Technische Anleitung VS - NfD

Jochen Topf
topf@rz.uni.karlsruhe.de

5. August 1997

Zusammenfassung

In diesem Artikel werden die für den Administrator und den Softwareentwickler wichtigsten Konzepte des NICK-Systems erklärt. Der mit UNIX vertraute Leser sollte nach der Lektüre in der Lage sein, die Grundlagen des NICK-Systems zu verstehen, den täglichen Betrieb zu gewährleisten und Software-Updates durchzuführen. Dabei kann und will dieses Dokument natürlich nicht alles erklären. Zur weiteren Lektüre sei der Leser insbesondere auf die Manpages und die einschlägige UNIX- (und insbesondere LINUX)-Literatur verwiesen.

Inhaltsverzeichnis

0	Einführung	3
0.1	Geschichte	3
0.2	Sicherheit	3
1	Hardware	4
1.1	Spezifikation	4
1.2	Sicherheit	4
2	BIOS	4
3	Betriebssystem	5
3.1	Kernel	5
3.2	Partitionierung	5
3.3	LILO	6
3.4	Bootvorgang	6
3.5	Firewalling	6
3.6	Directory-Hierarchie und Files	8
4	Elemente des NICK-Systems	9
4.1	Ablaufsteuerung	9
4.2	Das Menu	9
4.2.1	Format des <i>.mnu</i> -Files	9
4.2.2	Beispiel für ein <i>.mnu</i> -File	9
4.3	WorldWideWeb-Zugang	10
4.4	Bibliotheks-Zugang	10
4.5	rzstud-Zugang	10
4.6	Sonstiges Elemente der Oberfläche	10
4.7	Anschluß eines ASCII-Terminals	10
4.8	Anschluß eines Laufschriftmoduls	11
5	Das Entwicklungssystem	12
5.1	Partitionierung	12
5.2	Directory-Hierarchie	12
6	Der Administrationsrechner	14
6.1	Directories und Files in <i>/home/nickadm</i>	14
6.1.1	Das Masterconfigfile	15
6.2	Directories und Files in <i>/home/nick</i>	17
7	Die Bootdisketten	18
7.1	Inhalt der Rootdiskette	18
7.2	Vorgang bei der Erstinstallation	18
8	Der Logging-Mechanismus	19
8.1	Syslog	19
8.2	Logrec	19
9	Administrationszugang	20
10	Verteilung von Software	21
10.1	Verteilung <i>per Hand</i>	21
10.2	Automatischer Update-Mechanismus	21
10.2.1	Das Skript <i>newversion</i>	21

A	Programme und Skripte auf den NICKs	22
A.1	Neu geschriebene Programme	22
A.1.1	crexec	22
A.1.2	display	22
A.1.3	forw	22
A.1.4	logrec	22
A.1.5	menu	22
A.1.6	timeout	22
A.1.7	ticker	22
A.1.8	preloadfonts	23
A.2	Sonstige Programme	23
A.2.1	fvwm	23
A.2.2	mosaic	23
A.2.3	mpeg_play	23
A.2.4	mount	23
A.2.5	rclock	23
A.2.6	rxvt	23
A.2.7	ghostscript	23
A.2.8	ghostview	24
A.3	Shell-Skripts	24
A.3.1	<i>/usr/sbin/nickreboot</i>	24
A.3.2	<i>/sbin/mountro</i> und <i>/sbin/mountrw</i>	24
A.3.3	<i>/usr/nick/alive</i>	24
A.3.4	<i>/usr/nick/xinitrc</i>	24

0 Einführung

0.1 Geschichte

Die NICKs sind eine Reihe von Rechnern, die seit 1994 (?) auf dem Campus der Universität Karlsruhe aufgestellt wurden, um allen Studenten einen einfachen Zugriff auf Informationen über die Universität zu geben. Das **Netzwerkweite Informationssystem auf dem Campus der Universität Karlsruhe** (NICK) sollte aber auch eine Schnittstelle zu anderen Informationssystemen in der Welt haben. Als Grundlage wurde das damals gerade in der Entstehung befindliche World Wide Web (WWW) gewählt.

Außerdem sollten die NICKs einen einfachen Zugang zur Universitätsbibliothek und zu den beiden von den meisten Studenten verwendeten Rechnern (*IBM3090* und *rzstud*) ermöglichen. Als Betriebssystem wurde LINUX, eine frei erhältliche UNIX-Variante für PCs, verwendet. Die ursprüngliche Konzeption hat sich sehr bewährt und wurde in zwei Jahren kaum abgewandelt, einige Details wurden verbessert.

Im Frühjahr 1996 wurde eine neue Software-Generation in Betrieb genommen, die auf einem neuen Kernel basiert und einige Erweiterungen beinhaltet. Insbesondere wurde ein automatischer Update-Mechanismus entwickelt und implementiert, der es erlaubt, auf einfache Weise neue oder erweiterte Software auf allen NICKs einzusetzen. Zur Zeit sind knapp 20 NICKs im Einsatz, die überall auf dem Campus verteilt stehen (siehe Anhang B).

0.2 Sicherheit

Bei einem Rechner, der an einem nicht überwachten Ort auf dem Campus steht und außerdem eine Verbindung zum Internet hat, sind einige Sicherheitsmaßnahmen notwendig. Dabei sind zwei grundsätzlich verschiedene Sicherheitsaspekte zu berücksichtigen: Einerseits muß der Rechner in der Lage sein, sich selber so zu schützen, daß er jederzeit betriebsbereit ist, andererseits muß der Rechner verhindern, daß er mißbräuchlich benutzt wird (z.B. um Mails zu verschicken oder andere Rechner zu hacken), sei es von dem Benutzer, der am Gerät steht, sei es von einem "Hacker", der sich über das Internet Zugang verschafft. Diese Vorgaben wurden insbesondere durch das Firewalling (siehe Kapitel 3.5, S. 6), das Read-Only-Filesystem (siehe Kapitel 3.2, S. 5) und die Modifizierung von unzähligen Programmen (wie z.B. *Mosaic*) erreicht.

1 Hardware

1.1 Spezifikation

Prozessor: 486/40 oder 486/66

Grafikkarten: Spea V7 Mirage VLB, Miro 20SD

RAM: 16 MB

Platte: mindestens 170 MB

Tastatur: Cherry

Maus: Logitech, 3-Tasten

Monitor: Spea Videoseven, Philips

Netzwerkkarte: SMC Ultra

1.2 Sicherheit

- Soweit möglich Unterbringungen des Rechners in einem abgeschlossenen Raum. Tastatur, Maus und Monitor lassen sich so natürlich nicht schützen. Monitor und Tastatur lassen sich durch Metallstifte, Schlösser etc. sichern.
- Um ein Booten des Rechners mit einer Floppy zu verhindern wurde das Floppy-Laufwerk ausgebaut.
- Durch Verwendung eines Read-Only-Filesystems (siehe Kapitel 3.2, S. 5) ist es im Gegensatz zu einem gewöhnlichen UNIX-System kein Problem, wenn man den Rechner im Betrieb ausschaltet oder resettet.

2 BIOS

- BIOS ist passwortgesichert.
- Floppy-Zugriff im BIOS ist abgeschaltet. Bootreihenfolge auf C: A: gesetzt.

3 Betriebssystem

3.1 Kernel

Als Betriebssystem wurde Linux ausgewählt. Zur Zeit wird die Version 1.2.13 benutzt. In dieser Version ist IP-Firewalling auf Kernel-Ebene bereits eingebaut.

3.2 Partitionierung

Partitionierung eines Standard-NICK-Systems. Weil in den verschiedenen NICKs verschieden große Platten verwendet werden, stimmt die Partitionierung nicht überall genau überein.

```
Disk /dev/hda: 10 heads, 34 sectors, 984 cylinders
Units = cylinders of 340 * 512 bytes
```

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/hda1		1	1	338	57443	83	Linux native
/dev/hda2		339	339	676	57460	83	Linux native
/dev/hda3		677	677	773	16490	82	Linux swap
/dev/hda4		774	774	984	35870	5	Extended
/dev/hda5		774	774	882	18513	83	Linux native
/dev/hda6		883	883	984	17323	83	Linux native

Funktion der Partitionen:

/dev/hda1 Root-Partition 1

/dev/hda2 Root-Partition 2

Diese beiden Partitionen werden für das Root-Filesystem verwendet. Natürlich wird davon normalerweise nur eine benutzt. Die zweite ist nur für den automatischen Update-Mechanismus (siehe Kapitel 10.2, S. 21) notwendig. Dabei wird nämlich die neue NICK-Version auf die andere – zur Zeit nicht benutzte – Root-Partition eingespielt und dann mit LILO auf diese Partition umgeschaltet. Die zur Zeit benutzte Root-Partition steht immer im File */etc/PARTITION*. Die Root Partition wird **immer** Read-Only gemountet. Damit wird sichergestellt, daß nach einem versehentlichen oder absichtlichen Ausschalten des Rechners oder dergl. kein Filesystemcheck notwendig ist. Die Konsistenz der Daten ist immer gewährleistet. Files auf der Root-Partition, die geschrieben werden müssen, werden durch einen symbolischen Link auf ein entsprechendes File auf der */tmp*-Partition ersetzt.

/dev/hda3 Swap-Partition

16 MB Swap Partition.

/dev/hda5 */tmp*-Partition

Diese Partition wird unter */tmp* gemountet. Bei jedem Booten wird auf dieser Partition ein neues Filesystem angelegt. Damit erspart man sich einen langwierigen und fehleranfälligen Filesystemcheck.

/dev/hda6 */sec/tmp*-Partition

Diese Partition wird unter */sec/tmp* gemountet. Vor dem Anzeigen des Menus wird auf dieser Partition ein neues Filesystem angelegt. Damit erspart man sich einen langwierigen und fehleranfälligen Filesystemcheck. Das Filesystem wird für temporäre Files von Programmen, die in einer chroot-Umgebung

laufen (z.B. Mosaic), verwendet. Die Größe dieses Filesystems ist so zu wählen, daß einerseits der mkfs (der ja recht häufig ausgeführt wird) nicht zu lange dauert, andererseits aber alle Files, die Mosaic lädt, darauf passen.

3.3 LILO

Der Linux-Bootloader LILO ist mit einem Passwort abgesichert, damit der Benutzer nicht irgendeine Boot-Option angeben kann (z.B. im Single-User-Mode booten). Da das Passwort im Klartext im File */etc/lilo.conf* steht, ist besonders auf die Sicherheit dieses Files auf dem Entwicklungssystem etc. zu achten.

Zum Booten stehen im LILO drei Möglichkeiten zur Wahl:

default Normale NICK-Umgebung wird gestartet. Wird auch ausgewählt, wenn man am Bootprompt nichts angibt.

level2 Der NICK wird in den Runlevel 2 gebootet. Login auf der Konsole ist möglich. Netzwerk ist aktiviert. Diese Einstellung sollte man im Normalfall zum Debugging verwenden. Auswahl nur mit Passwort möglich.

debug Single-User-Mode. Kein Netzwerk. Es wird direkt eine Shell gestartet. Englische Tastatur. Auswahl nur mit Passwort möglich. (Nur im Notfall verwenden, wenn level2 nicht mehr geht).

3.4 Bootvorgang

Single User Mode (*/etc/rc.d/rc.S*)
/proc und */tmp* sind gemountet.

level2 (*/etc/rc.d/rc.M*)
 Netzwerk ist initialisiert. *Syslogd*, *klogd*, *inetd*, *sshd* laufen. Deutsche Tastatur.

level3 (*/etc/rc.d/rc.3*)
 X wird gestartet und die NICK-Oberfläche angezeigt.

3.5 Firewalling

Der Linux-Kernel hat ein Firewalling eingebaut, mit dem es ohne Probleme möglich ist, schon im Kernel den Zugriff auf den NICK und von den NICKs einzuschränken. Dabei ist es möglich, IP-Pakete aufgrund der Kombination der enthaltenen Source- und Destinationadresse, des Protokolls (TCP, UDP, ICMP) und der Portadresse (bei UDP und TCP) durchzulassen oder zu blockieren. Auf den NICKs wird das vor allem verwendet, um den Zugriff auf die für die Administration notwendigen Dienste (*rlogin*, *ssh*, etc.) zu beschränken. Das Firewalling wird in dem Skript */etc/rc.d/rc.fw* eingestellt. Im einzelnen sind folgende Regeln implementiert:

```
/sbin/ipfwadm -F -p deny
Kein IP-Packet-Forwarding.
```

```
/sbin/ipfwadm -B -p deny
Per Default werden alle Pakete blockiert.
```

```
/sbin/ipfwadm -B -a accept -S 0.0.0.0/0 -D $NICK_IP -P icmp -b
ICMP (z.B. ping) geht von überall.
```

```
for i in $NICK_FW_ADMIN; do
  /sbin/ipfwadm -B -a accept -S $i 1:1023 -D $NICK_IP login -P tcp -b
  /sbin/ipfwadm -B -a accept -S $i -D $NICK_IP ssh -P tcp -b
done
```

Alle Rechner in der Variablen FW_ADMIN dürfen per *rlogin* und per *ssh* zugreifen.

```
if [ "$NICK_TTYS1" = "DISPLAY" -o "$NICK_TTYS2" = "DISPLAY" -o
"$NICK_TTYS3" = "DISPLAY" ]; then
  for i in $NICK_FW_DISPLAY; do
    /sbin/ipfwadm -B -a accept -S $i -D $NICK_IP display -P tcp -b
  done
fi
```

Wenn eine der Variablen TTYS1, TTYS2 oder TTYS3 auf DISPLAY gesetzt ist, wird für alle Rechner in der Variablen FW_DISPLAY der Zugriff auf den Display-Daemon freigegeben.

```
/sbin/ipfwadm -B -a accept -S 129.13.0.0/16 www nntp ftp 20 -D $NICK_IP -P tcp -b
```

FTP, WWW und News geht nur zu Rechnern der Uni Karlsruhe. Da ja immer der Proxy (für http, gopher, ftp) verwendet wird, ist das keine Einschränkung.

```
/sbin/ipfwadm -B -a accept -S 129.13.95.0/29 login -D $NICK_IP 1:1023 -P tcp -b
```

Vom Nick ausgehender *rlogin* zu den *rzstuds* ist erlaubt.

```
/sbin/ipfwadm -B -a accept -S 129.13.0.0/16 -D $NICK_IP ident -P tcp -b
```

Ident-Abfrage geht von überall in der Uni.

```
/sbin/ipfwadm -B -a accept -S 0.0.0.0/0 -D $NICK_IP finger -P tcp -b
```

Finger geht von überall in der Welt. Da kommt nämlich so eine nette Message, was der NICK überhaupt ist.

```
if [ "$NICK_TTYS1" = "TERMINAL" -o "$NICK_TTYS2" = "TERMINAL" -o
"$NICK_TTYS3" = "TERMINAL" ]; then
  /sbin/ipfwadm -B -a accept -S 129.13.96.2 telnet -D $NICK_IP -P tcp -b
fi
```

Telnet auf die *nz20* geht genau dann, wenn ein ASCII-Terminal angeschlossen ist (siehe Kapitel 4.7, S. 10).

```
for i in $NICK_NAMESERVER; do
  /sbin/ipfwadm -B -a accept -S $i domain -D $NICK_IP -P udp -b
  /sbin/ipfwadm -B -a accept -S $i domain -D $NICK_IP -P tcp -b
done
```

DNS geht nur mit den Nameservern des RZ, die in der Variable NAMESERVER stehen.

```
/sbin/ipfwadm -B -a accept -S $NICK_FW_LOG syslog 10012 -D $NICK_IP -P udp -b
```

Syslog und logrec nur zum dem eingetragenen Logging-Rechner.

```
for i in $NICK_NETDATE; do
  /sbin/ipfwadm -B -a accept -S $i time -D $NICK_IP -P udp -b
done
```

Nur zu den eingetragenen Timeservern gehts auf dem Time Port.

3.6 Directory-Hierarchie und Files

Die Directory-Hierarchie und die Lage der Files orientiert sich an den üblichen LINUX-Systemen und dem LINUX Filesystemstandard (FSSTND). Änderungen gegenüber dem Standard ergeben sich aus den besonderen Anforderungen des NICK-Systems. (Im wesentlichen sind das die chroot-Umgebung und das Read-Only-Filesystem). Im folgenden sind nur die Directories und Files erwähnt, die es nur im NICK-System gibt, oder die von der normalen Verwendung abweichen und deren Funktion sich nicht sofort aus dem Zusammenhang erschließt.

/etc

/etc/NICKCONFIG File, das aus dem Masterconfigfile (siehe Kapitel 6.1.1, S. 15) erzeugt wird und alle Definitionen enthält, die der NICK braucht, um sich richtig zu konfigurieren.

/etc/finger.out Meldung, die bei einer Finger-Anfrage zurückgegeben werden soll.

/etc/nissue Meldung, die vor dem Login-Prompt auf einem angeschlossenen ASCII-Terminal ausgegeben wird (siehe Kapitel 4.7, S. 10).

/home Homeverzeichnis der Admins. Auch diese sind natürlich Read-Only. Dient im wesentlichen nur dazu, daß sich jeder seine Umgebung so einrichten kann, wie er es mag. Außerdem wird es für den Zugang über *ssh* benötigt.

/root Homeverzeichnis von root.

/sec Root-Verzeichnis der chroot-Umgebung.

/sec/bin Programme, die vom Menu in der chroot-Umgebung aufgerufen werden (Mosaic und dergl.).

/sec/data Grafiken für das Timeout-Programm

/sec/dev Enthält nur ein *pty/tty*-Paar für das *rxvt*, das für den rzstud-Zugang verwendet wird.

/sec/etc Minimalversion des */etc*-Verzeichnisses.

/sec/sbin Binaries, die als Hilfsprogramme von Mosaic ausgeführt werden.

/sec/tmp Mountpoint für das tmp-Filesystem für die chroot-Umgebung.

/tmp Mountpoint für das tmp-Filesystem

/usr/nick Enthält NICK-spezifische Skripts, Programme und Daten für das Menu und einige andere Kleinigkeiten.

/usr/nick/xpm Enthält Grafiken, die vom *menu*-Programm verwendet werden.

4 Elemente des NICK-Systems

4.1 Ablaufsteuerung

Das Skript `/usr/nick/xinitrc` wird von dem Bootskript `/etc/rc.d/rc.3` gestartet. Es sorgt für den Aufruf des `menu` Programms (siehe nächsten Abschnitt) und ruft danach das Programm auf, das der Benutzer sich gewünscht hat. Die Standardausgabe dieses Skripts befindet sich im File `/tmp/x.out`, die Standardfehlerausgabe in `/tmp/x.err`. Dieses Skript sorgt auch dafür, daß mittels `logrec` ein Logfileeintrag geschrieben wird, wenn der Benutzer ein Programm auswählt.

4.2 Das Menu

Das Menu wird durch das Programm `menu` (wer hätte das gedacht :-)) realisiert. Es liest ein `.mnu` File ein, das eine Beschreibung des darzustellenden Menus enthält. Dann werden die benötigten Grafiken geladen und das Menu angezeigt. Wenn der Benutzer nun ein Icon auswählt, gibt das Menu-Programm auf der Standardausgabe den Namen dieses Icons aus und beendet sich. Der Name wird dann von einem Skript ausgewertet, das die diversen Programme (Mosaic etc.) startet.

4.2.1 Format des `.mnu`-Files

Das Konfigurationsfile für das Menu besteht aus mehreren Zeilen, die wiederum mehrere Felder enthalten, die jeweils in " eingeschlossen und durch Leerzeichen getrennt sind. Es folgt eine Liste aller Zeilen mit den Feldern:

1. "Hintergrundbild" "Mittenbild" "(nicht benutzt)" "X-Verschiebung des Mittenbildes" "Y-Verschiebung des Mittenbildes"
2. "Erste Zeile des Textes im Mittenbild" "Zweite Zeile des Textes im Mittenbild" "Bildschirmauflösung in X-Richtung" "Bildschirmauflösung in Y-Richtung"
3. "Font für alle Texte"
4. "Screensaver-Icon 1" "Screensaver-Icon 2" "Screensaver-Icon 3" "Timeout für Screensaver in Sekunden"

Für jedes Icon, das ein Programm aufrufen soll, kommt dann eine Zeile mit dem folgende Aufbau:

- "Grafikfile für das Icon" "Name des Icons" "X-Koordinate des Icons" "Y-Koordinate des Icons" "Erste Zeile des Textes im Mittenbild" "Zweite Zeile des Textes im Mittenbild" "Programm"

Das letzte Feld in einer Icon-Zeile ("Programm") gibt an, was passiert, wenn dieses Icon angeklickt wird. Beim Anklicken eines Icons wird das Programm, das hier angegeben ist, gestartet und dann eine Sekunde gewartet, dann beendet sich das Menu-Programm und gibt den Namen des angeklickten Icons aus. Wenn das Feld leer ist, wird kein Programm gestartet und auch keine Sekunde gewartet. Dieses Feld dient dazu Bildschirmeffekte, wie z.B. ein "Zerlaufen" des Menubildes zu realisieren :-).

4.2.2 Beispiel für ein `.mnu`-File

```
"/usr/nick/xpm/back.pm" "/usr/nick/xpm/rzbox.pm" "xyz" "0" "0"
"Bitte auf eines der" "vier Symbole klicken!" "1024" "768"
```

```

"-adobe-times-bold-r-normal--24-240-75-75-p-132-iso8859-1"
"/usr/nick/xpm/rz_sb.pm" "/usr/nick/xpm/nick_s.pm" "/usr/nick/xpm/uni_s.pm" "120"
"/usr/nick/xpm/bibliothek.pm" "bib" "825" "33" "Zur Universitaetsbibliothek" "(Ausleihe und Recherche)" ""
"/usr/nick/xpm/nick3.pm" "nick" "109" "103" "Zum Informationssystem" "auf dem Campus (NICK)" ""
"/usr/nick/xpm/rzstud.pm" "rzstud" "385" "535" "Zur Studentenworkstation" "des Rechenzentrums" ""

```

4.3 WorldWideWeb-Zugang

Für den WWW-Zugang wird das Programm Mosaic in einer stark modifizierten Version verwendet. Als Startseite wird <http://www.rz.uni-karlsruhe.de/Nick/> verwendet.

4.4 Bibliotheks-Zugang

Der Zugang zur Bibliothek entspricht dem WWW-Zugang mit dem einzigen Unterschied, daß eine andere WWW-Startseite (<http://www.ubka.uni-karlsruhe.de/hylib/suchmaske.html>) benutzt wird.

4.5 rzstud-Zugang

Für den rzstud-Zugang wird das Terminalemulationsprogramm *rxvt* und das Programm *forw* für den Remote Login verwendet. Das Programm *forw* sorgt dabei dafür, daß der User immer auf die "richtige" Maschine eingeloggt wird (je nach Username *rzstud1* oder *rzstud2*).

4.6 Sonstiges Elemente der Oberfläche

Alle Funktionen, die vom Menu aufgerufen werden, haben ein paar Elemente gemeinsam:

Exit: Links unten wird ein EXIT-Button angezeigt, der einen ins Menu zurück führt.

Zähler: Links oben ist ein Zähler zu sehen, der in Stunden und Minuten anzeigt, wie lange man sich schon in dieser Funktion befindet. Außerdem ändert dieses Fenster mit der Zeit seine Farbe von grün über gelb zu rot. Damit soll deutlich gemacht werden, daß man sich nicht zu lange vor dem Rechner aufhalten soll und anderen Leuten auch mal eine Chance geben, den NICK zu benutzen.

Timeout: Es existiert ein Timeout, der dafür sorgt, das nach einer Weile ohne Tastatureingaben und Mausbewegungen ins Menu zurückgeschaltet wird.

Uhr: Rechts oben ist einen Analog-Uhr zu sehen.

(Die ersten drei Funktionen werden durch das Programm *timeout* realisiert (siehe Anhang A.1.6, S. 22), die Uhr durch das Programm *rclock*.)

4.7 Anschluß eines ASCII-Terminals

Es besteht die Möglichkeit ASCII-Terminals an die seriellen Schnittstellen eines NICK anzuschließen und sich darüber in die rzstud oder andere Rechner einzuloggen. Normalerweise steht dafür natürlich nur die zweite serielle Schnittstelle zur Verfügung, weil die erste ja von der Maus belegt ist.

Um diese Funktion zu aktivieren, sind im Masterconfigfile die Variablen TTYS1, TTYS2 und/oder TTYS3 auf TERMINAL zu setzen. Es wird dann auf der jeweiligen seriellen Schnittstelle ein modifiziertes *agetty* namens *ngetty* gestartet (inittab). Dies gibt das File */etc/nissue* aus und wartet auf einen Login. Ist ein Login-Name eingegeben worden, wird das Skript */sbin/nicktermmenu* gestartet. Dieses unterscheidet je nach dem eingegebenen Usernamen, ob entweder ein Telnet auf die *nz20* (Beim Usernamen "m") oder ob ein Rlogin auf die *rzstud* erfolgen soll (alle anderen Usernamen).

4.8 Anschluß eines Laufschriftmoduls

In einem Schaukasten an der Einfahrt in den Mensahof (neben der Uni-Bibliothek) ist ein Laufschriftmodul installiert. Für die Programmierung wurde ein Rechner mit Netzanschluß benötigt, der in der Nähe des Moduls steht. Und da der NICK in der Mensa der nächste Rechner ist, wurde beschlossen das Display an den Mensa-NICK per serieller Schnittstelle anzuschließen. Es wurden einige Programme entwickelt, die es auf einfache Weise ermöglichen, von einigen Rechnern im Netz einen Daemon auf dem NICK anzusprechen, der dann die Programmierung des Display vornimmt. Für die Konfiguration des NICK-Systems gibt mehrere Variablen im Zusammenhang mit dem Display: Ein der Variablen TTYS1, TTYS2 und TTYS3 kann auf DISPLAY gestellt werden, wenn ein Display an dem entsprechenden Port angeschlossen ist. Die Variable FW_DISPLAY gibt an, von welchen Systemen aus auf den Daemon zugegriffen werden kann. Jeder User auf einem dieser Systemen kann das Display nach seinen Wünschen programmieren!

5 Das Entwicklungssystem

Der Rechner *nick00.rz.uni-karlsruhe.de* ist das Entwicklungssystem für die NICK-Rechner. Auf der Platte sind mehrere Partitionen, die einem erlauben, den Rechner als normales Linux-System zu benutzen oder ihn als NICK zu booten. Das normale Linux-System dient zum Entwickeln der Software, das NICK-System zum Test. Außerdem dient die NICK-Partition als Master-Kopie, aus der die tar-Files erzeugt werden, die dann auf die anderen NICK-Systeme verteilt werden.

5.1 Partitionierung

Disk /dev/hda: 64 heads, 63 sectors, 789 cylinders
Units = cylinders of 4032 * 512 bytes

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/hda1	*	1	1	28	56416+	83	Linux native
/dev/hda2	*	29	29	56	56448	83	Linux native
/dev/hda3	*	57	57	76	40320	6	DOS 16-bit >=32M
/dev/hda4		77	77	789	1437408	5	Extended
/dev/hda5		77	77	86	20128+	83	Linux native
/dev/hda6		87	87	96	20128+	83	Linux native
/dev/hda7		97	97	104	16096+	82	Linux swap
/dev/hda8	*	105	105	358	512032+	83	Linux native
/dev/hda9		359	359	359	1984+	83	Linux native
/dev/hda10		360	360	360	1984+	83	Linux native
/dev/hda11		361	361	789	864832+	83	Linux native

Partition	Mountpoint	Beschreibung
/dev/hda1	/nick1	NICK-Root-Partition 1
/dev/hda2	/nick2	NICK-Root-Partition 2
/dev/hda3	/dos	DOS-Partition oder für Erweiterungen
/dev/hda5	/tmp1	NICK-/tmp-Partition
/dev/hda6	/tmp2	NICK-/sec/tmp-Partition
/dev/hda7		Swap-Partition
/dev/hda8	/	Linux Root-Partition (NICK-Entwicklungssystem)
/dev/hda9	/disk1	Zum Diskettenerstellen
/dev/hda10	/disk2	Zum Diskettenerstellen
/dev/hda11	/rest	Für Erweiterungen

(Die Partitionen 1, 2, 5 und 6 entsprechen den Partitionen eines "richtigen" NICK-Systems (siehe Kapitel 3.2, S. 5).)

(Die Partitionen 9 und 10 haben etwa die Größe einer Floppy Disk. Wenn man des öfteren mit langsamen Disketten arbeiten muß, kann man die per dd hier draufkopieren und dann mounten. Wenn man mit der Arbeit fertig ist, unmounted man die Platte wieder und kopiert die Daten auf die Floppy zurück.)

5.2 Directory-Hierarchie

/nick/ Alle Files, die speziell zum Nick-Entwicklungssystem gehören.

/nick/src/ Sourcecode für NICK-spezifische Programme (siehe Anhang A).

/nick/doc/ Dokumentation zum NICK-System.

/nick/disk/ Enthält Files für NICK-Bootdisketten (siehe Kapitel 7, S. 18).

/nick/tars/ Enthält Tarfiles von NICK-Distributionen.

6 Der Administrationsrechner

Es gibt einen Rechner, auf dem alle Daten und Programme liegen, die zur Verwaltung der NICK-Rechner notwendig sind. Zur Zeit ist das die *nz14* (FTP-Server). Hier gibt es zwei User *nick* und *nickadm* mit den entsprechenden Homeverzeichnissen */home/nick* und */home/nickadm*. Der User *nickadm* ist für die Verwaltung etc. der NICKs gedacht. Unter der Userid des Users *nick* läuft ein Daemon zur Speicherung der Logfiles.

6.1 Directories und Files in */home/nickadm*

- ~**nickadm/adm/** Files für die NICK-Administration
- ~**nickadm/adm/masterconfig** Hauptkonfigurationsfile. Alle NICKs müssen hier eingetragen werden. Aus diesem File werden alle anderen Config Files automatisch erzeugt (siehe Kapitel 6.1.1, S. 15).
- ~**nickadm/adm/lilopassword** Dieses File enthält das Passwort für LILO im Klartext!
- ~**nickadm/adm/NICKLIST** Liste aller Nicknummern (wird aus dem Masterconfigfile erzeugt).
- ~**nickadm/adm/TARFILES** Liste aller Tarfiles (wird beim Erzeugen der Tarfiles angelegt).
- ~**nickadm/adm/Makefile** Makefile mit dessen Hilfe man leicht neue config-Files und tar-Files erzeugen kann.
- ~**nickadm/adm/notify** eMail-Adresse desjenigen, der informiert werden soll, wenn es Probleme mit dem automatischen Update gibt oder wenn das Update fertig gelaufen ist.
- ~**nickadm/adm/src/** Template-Files, aus denen die auf den verschiedenen NICKs unterschiedlichen Files erzeugt werden.
- ~**nickadm/bin/** Binaries und Skripts für die NICK-Administration.
- ~**nickadm/cfg/** Config-Files für die einzelnen NICKs. Die Files in diesem Verzeichnis werden automatisch aus dem *masterconfig* File erzeugt.
- ~**nickadm/cfg/nick??conf** Config File für einen NICK.
- ~**nickadm/cfg/nick??/** Dieses Directory enthält die Files, die für jeden NICK verschieden sind. Es ist so aufgebaut, daß man davon ein tar-File machen und es auf dem jeweiligen NICK auspacken kann.
- ~**nickadm/disk/** Files für die Bootdisketten.
- ~**nickadm/dist/** Files hier werden von dem NICK-Verteilmechanismus erzeugt und benutzt.
- ~**nickadm/doc/** Dokumentation zum NICK.
- ~**nickadm/src/** Source für diverse NICK-Programme.
- ~**nickadm/tars/** Tarfiles zum Verteilen auf die NICKs. (Link auf *nick/tars*.)
- ~**nickadm/tars/nick.tar** Tarfile mit einem kompletten NICK-Image ohne die Files, die auf den NICKs unterschiedlich sind. Das erzeugt man auf dem Entwicklungssystem und kopiert es hierher.

`~nickadm/tars/nick??tar` Tarfiles mit den Files, die auf allen NICKs verschieden sind. Wird aus dem *masterconfig* File erzeugt.

`~nickadm/tars/passwd`

`~nickadm/tars/group` Diese beiden Files enthalten die Passwort- und die Group-Datei des NICK (allerdings ohne Passwörter!). Das wird benötigt um das Tarfile auf den NICKs mit den richtigen Userinformationen auspacken zu können. Dazu werden diese Files auf den NICK kopiert bevor man das Tarfile auspackt.

6.1.1 Das Masterconfigfile

Die meisten Files sind auf allen NICKs gleich. Natürlich gibt es aber auch Unterschiede, z.B. den Rechnernamen, die Grafikkarte ist verschieden, usw. Um alle diese Verschiedenheiten an zentraler Stelle verwalten zu können, gibt es das File */home/nickadm/adm/masterconfig*. Es enthält Definitionen von Variablen, die von verschiedenen Komponenten des NICK-Systems verwendet werden.

Format des Masterconfigfiles: Zeilen, die mit *#* anfangen sind Kommentarzeilen, Leerzeilen werden ignoriert. Zeilen der Form "VARIABLE=Inhalt" definieren eine Variable. Der Inhalt erstreckt sich dabei bis zum Zeilenende inclusive aller Leerzeichen. Keine Anführungsstriche verwenden! Zeilen der Form "*zweistellige Nummer*:" leiten die Variablendefinitionen für den NICK mit dieser Nummer ein. Die Zeile "default:", die immer am Anfang des Files stehen muß, leitet die Default-Definitionen ein. Wird für einen NICK eine Variable nicht definiert, wird die Definition aus dem Default-Bereich benutzt.

Variablen im Masterconfigfile: Die folgenden Variablen können im Masterconfigfile eingestellt werden. Auf dem NICK finden sich diese Variablen in dem File */etc/NICKCONFIG* wieder. Allerdings bekommen die Variablen dort ein *NICK_* vorangestellt, damit es keine Kollisionen mit anderen Environment-Variablen gibt. Aus *HOST* wird also z.B. *NICK_HOST*. Dieses File kann von Shell-Skripten mit ". */etc/NICKCONFIG*" eingelesen werden. In der folgenden Tabelle sind alle Variablen, die zur Zeit benutzt werden aufgelistet. Es ist ohne Probleme möglich weitere Variablen zu definieren.

Variable	Beispiel	Beschreibung
HOST	nick00	Der Name dieses Rechners ohne die Domain.
ONLINE	true	Rechner, die hier nicht auf <i>true</i> stehen, werden beim automatischen Software-Update übergangen.
DOMAIN	rz.uni-karls...	Die Domain, in der dieser Rechner ist.
IP	129.13.98.85	Die IP-Nummer dieses Rechners.
NETWORK	129.13.0.0	Das Netzwerk, in dem dieser Rechner ist.
BROADCAST	129.13.255.255	Broadcastadresse.
NETMASK	255.255.0.0	Netzmaske für diesen Rechner.
GATEWAY	129.13.98.254	IP-Nummer des Gatewayrechners.
NAMESERVER	129.13.98.225	Durch Leerzeichen getrennte Liste von Nameservern.
MONITOR	Philips	Monitor-Identifizier für XF86Config.
DEVICE	Miro	Identifizier der Grafikkarte für XF86Config.
TEST	true	<i>true</i> für einen Testrechner, <i>false</i> für normale NICKs. Bei Testrechnern wird der Bildschirm beim booten nicht blau eingefärbt.
NETDATE	129.13.64.5	Liste mit Rechnern, deren Timeserver abgefragt werden soll.
PI	false	Mit pi-Button. Dies ist für einen Aprilscherz gedacht und im normalen Betrieb immer auf <i>false</i> zu setzen.
SYSLOG	@nz14.rz...	Wohin Syslog der seine Meldungen schreiben soll. Das kann entweder <i>@rechnername</i> sein oder ein Pfad.
FW_LOG	129.13.96.4	IP-Nummer des Rechners, auf dem der syslogd und der logrecd läuft. Für das Firewaling.
FW_ADMIN	129.13.96.4	Liste von IP-Nummern der Rechner, die Admin-Berechtigung haben.
FW_DISPLAY	129.13.96.4	Liste von IP-Nummern der Rechner, von denen der Display-Daemon bedient werden darf.
GEB	20.21	Gebäude in dem der NICK steht (Nummer).
ORT	Raum 318	Beschreibung des Ortes, wo der NICK steht.
TIMEOUT	180	Timeout für Mosaic, rzstud-Login, etc. (in Sekunden)
BOOTDISK	/dev/hda	Bootdisk
ROOT1PART	/dev/hda1	1. Rootpartition
ROOT2PART	/dev/hda2	2. Rootpartition
SWAPPART	/dev/hda3	Swap-Partition
TMPPART	/dev/hda5	Tmp-Partition
SECTMPPART	/dev/hda6	Sec/Tmp-Partition
RES	1024x768	Bildschirmauflösung
TTYs1	NONE	Gerät, daß an ttyS1 hängt. Es sind folgende Einträge möglich: NONE, kein Gerät; TERMINAL, ein ASCII-Terminal für den rzstud-Zugang ist angeschlossen; DISPLAY, das Moving Message Display ist angeschlossen; LOGIN, ein ASCII-Terminal ist angeschlossen von dem man sich lokal einloggen kann (nur für Debugging).
TTYs2	NONE	Gerät, daß an ttyS1 hängt. Möglichkeiten siehe oben.
TTYs3	NONE	Gerät, daß an ttyS1 hängt. Möglichkeiten siehe oben.
INVPC		Inventarnummer des PCs
INVMON		Inventarnummer des Monitors
SERPC		Seriennnummer des PCs
SERMON		Seriennnummer des Monitors

Was wird aus dem Masterconfig: Mit dem *Makefile* in */home/nickadm/adm* kann man aus dem *masterconfig* File alle vom NICK oder dem Administrationsystem benötigten Files erzeugen. Das *Makefile* kennt folgende Targets:

all Entspricht *config* und *tar*.

config Das Shell-Skript *mknick* wird gestartet. Es erzeugt ein paar im folgenden benötigte Directories und startet für jeden NICK die Programme *mknickcfg* und *mknickfiles*, die wiederum alle für einen NICK benötigten Files erzeugen und in einem Directory-Baum unterhalb */home/nickadm/cfg/nick??/* ablegen. Als Nebeneffekt produziert das Skript eine Liste aller NICKs in */home/nickadm/adm/NICKLIST*.

tar Das Perl-Skript *mknicktar* wird gestartet. Es packt die Files in */home/nickadm/cfg/nick??/* in ein tar-File für jeden NICK zusammen. Diese werden in */home/nickadm/tars/* abgelegt, wo der automatische Verteilmechanismus sie findet. Als Nebeneffekt produziert das Skript eine Liste aller NICKs in */home/nickadm/adm/TARFILES*.

6.2 Directories und Files in */home/nick*

~nick/bin/ Skripts und Programme zur Logfileverwaltung (siehe Kapitel 8, S. 19).

~nick/log/ Logfiles, die mit *logrec* erzeugt wurden.

~nick/syslog/ Logfiles, die vom *syslog* erzeugt wurden.

~nick/tar/ Tarfiles zum Verteilen auf die NICKs.

~nick/tmp/ Verzeichnis für temporäre Files.

7 Die Bootdisketten

Zum NICK-System gehören auch zwei Disketten, mit denen ein neuer NICK eingerichtet werden kann. Die erste ("Bootdiskette") enthält den Kernel. Mit ihr wird der Rechner gebootet. Auf Anforderung wird dann die zweite Diskette ("Rootdiskette") eingelegt. Sie enthält alle notwendigen Daten, Programme und Skripts um einen neuen NICK einzurichten.

7.1 Inhalt der Rootdiskette

Die Rootdiskette enthält ein normales Linux-System. Außerdem findet man dort folgende Verzeichnisse:

/root Home-Verzeichnis des root-Users. Enthält eine Reihe von Skripts zum Initialisieren des NICK-Systems.

/nick Enthält eine Reihe von Konfigurationsdateien für alle NICKs. Um einen NICK einrichten zu können **muß** die Konfigurationsdatei für diesen NICK vorhanden sein!

7.2 Vorgang bei der Erstinstallation

1. Config-Files aus der *masterconfig* erstellen (siehe 6.1.1, S. 15).
2. Das File `~nickadm/cfg/nickXX.conf` für den neuen Nick auf die Root-Diskette kopieren: `cp /home/nickadm/cfg/*.conf /floppy/nick/`
3. Rechner mit der Boot- und Root-Diskette booten. Auf die Frage nach der Nicknummer diese eingeben. Einloggen als *root* (kein Passwort).
4. Mit `ping 129.13.96.4` überprüfen, ob das Netzwerk richtig konfiguriert ist. Wenn nein, von Hand überprüfen und korrigieren.
5. `/root/runfirst` aufrufen und Platte partitionieren. (siehe Kapitel 3.2, S. 5).
6. Nick rebooten.
7. Starten von `/root/runsecond`. Beobachten, ob alles gut geht! Wenn alles geklappt hat wird der NICK automatisch gebootet und müßte als NICK hochkommen. Wenn es Probleme gibt kann man versuchen beim LILO-Prompt "default vga=0" anzugeben. Dann wird der Bildschirm beim Booten nicht weggeschaltet und man sieht die Bootmeldungen.
8. Überprüfen, ob einloggen von der *nz14* aus möglich ist.

8 Der Logging-Mechanismus

Meldungen der NICKs werden auf zwei verschiedene Arten geloggt: Über den normalen Syslog-Mechanismus und einen speziell für den NICK geschriebenen weiteren Service.

8.1 Syslog

Alle Meldungen die an den Syslog-Daemon geschickt werden, werden an den Adminrechner weitergeleitet und dort im normalen Syslog geloggt. Dieser speichert die Meldungen in */var/adm/syslog* bzw. */var/adm/messages*. Mit dem Skript */home/nick/bin/greplogs* kann man aus diesen Logfiles die Zeilen, die von einem NICK kommen extrahieren. Sie werden durch das Skript automatisch in die Files */home/nick/syslog/syslog* und */home/nick/syslog/messages* abgelegt. Die Logfiles sollten regelmäßig überprüft werden, damit Fehlfunktionen und Hackversuche erkannt werden.

[Das sollte irgendwie noch automatisiert werden]

8.2 Logrec

Für spezielle NICK-Meldungen wird das Programm *logrec* verwendet, das über UDP kurze Meldungen an den *logrecd* auf der *nz14* weiterleitet. Die Meldungen dienen zur statistischen Auswertung und der Funktionsüberprüfung der NICKs. Diese werden in */home/nick/log/nicklog* gespeichert. Das File hat folgendes Format:

<IP-Nummer> <Datum/Uhrzeit> <Text>

Die IP-Nummer ist als 32-Bit Hexstring formatiert, ebenso die Zeitangabe. Der Text kann einer der folgenden sein:

boot	Der Rechner bootet. Wird bei jedem Bootvorgang geschickt.
reboot	Der Rechner führt einen angeforderten Reboot aus. D.h. der Administrator hat mit dem <i>nickreboot</i> -Skript (siehe Anhang A.3.1, S. 24) den Reboot ausgelöst.
alive	Der Rechner lebt noch. Wird alle 15 Minuten geschickt. Eine automatische Auswertung, die den Admin benachrichtigt, wenn ein NICK eine Weile nicht läuft, wäre praktisch, ist aber noch nicht implementiert.
menu	Das Menu wird angezeigt.
rzstud	Zugang zur rzstud wurde ausgewählt.
nick	Zugang zum NICK (Mosaic) wurde ausgewählt.
bib	Zugang zur Bibliothek wurde ausgewählt.
ibm	Zugang zur IBM3090 wurde ausgewählt. (Obsolete)

XXX Statistische Auswertung

XXX Aufräummechanismus.

9 Administrationszugang

Für die NICK-Administration existiert ein Remote-Zugang auf die NICKs. Der Zugang wird nur von bestimmten Rechnern aus gewährt. Die IP-Nummern dieser Rechner müssen im Masterconfigfile (siehe Kapitel 6.1.1, S. 15) in der Variable FW_ADMIN eingetragen sein.

- *rlogin*, wie üblich mit Username und Passwort.
- Ein *ftp* auf die NICKs ist **nicht** möglich.
- *ssh* (secure remote shell) ist ein Ersatz für *rlogin*, *rsh* und *rcp*. Verbindungen werden verschlüsselt und authentifiziert. Der Zugriff ist deutlich sicherer und auch einfacher, als mit *rlogin*. Allerdings erfordert *ssh* einige Vorbereitungen. So müssen auf jedem beteiligten Rechner Host-Keys erzeugt werden und für jeden User User-Keys. Erklärungen dazu finden sich in den Manpages zu *ssh*.

10 Verteilung von Software

10.1 Verteilung *per Hand*

Softwareverteilung *per Hand* sollte man normalerweise nicht machen, weil das schnell zu einem Durcheinander führt: Jeder NICK hat eine andere Softwareausstattung. Wenn man es dennoch macht, benutzt man sinnvollerweise *ssh* und *scp*. Dabei ist natürlich zu beachten, daß das Root-Filesystem Read-Only gemountet ist. Am besten kopiert man die neuen Files erst auf die tmp-Platte und loggt sich kurz ein, remountet die Root-Platte Read-Write, kopiert die Software oder packt das tar-File aus und remountet die Root-Platte wieder Read-Only. Damit ist sichergestellt, daß der Zeitraum, in dem die Platte gefährdet ist, möglichst klein ist. Falls ein Reboot nötig ist, benutze man das Programm *nickreboot* (siehe Anhang A.3.1, S. 24).

10.2 Automatischer Update-Mechanismus

Der automatische Updatemechanismus sorgt dafür, daß alle NICKs, die im *master-config* als **ONLINE** markiert sind, aktualisiert werden.

[Der Updatemechanismus ist noch nicht ganz fertig, einstweilen muss man noch das Programm *newversion* von Hand aufrufen].

10.2.1 Das Skript *newversion*

Den eigentlichen Update-Vorgang führt das Skript *newversion* durch. Es kann nur auf der *nz14* gestartet werden und muß als User *nickadm* laufen. Als einziger Parameter auf der Kommandozeile wird die Nummer des NICKs erwartet, der aktualisiert werden soll. Das Skript kontaktiert dann den NICK und führt den Update-Vorgang durch. Die Ausgabe des Skripts sollte beobachtet werden, um etwaige Fehler festzustellen.

Der Updatevorgang

1. Feststellung, auf welcher Partition das NICK-System zur Zeit läuft.
2. Auf der anderen Root-Partition wird ein Filesystem erzeugt und unter */mnt* gemountet.
3. Das File *~nick/tars/nick.tar* wird übertragen und ausgepackt. Es enthält den Großteil der NICK-Distribution.
4. Das File *~nick/tars/nick??tar* wird übertragen und ausgepackt. Dabei steht ?? für die Nummer des NICKs. Das Tarfile enthält alle Files, die sich von NICK zu NICK unterscheiden.
5. Die Files, die je nach Root-Partition verschieden sind, werden erzeugt.
6. Die für *ssh* benötigten Files werden erzeugt.
7. Der Befehl *sync* wird dreimal aufgerufen, um alle Änderungen sicher auf der Platte zu haben.
8. LILO wird gestartet, um die neue Partition zu aktivieren.
9. Die neue Platte wird ungemountet.

Damit ist der Update-Vorgang abgeschlossen. Wenn keine Fehler aufgetreten sind, ist die neue Software nach dem nächsten Booten aktiv. Wenn man will, kann man den Rechner mit dem *nickreboot*-Skript neu starten.

A Programme und Skripte auf den NICKs

A.1 Neu geschriebene Programme

A.1.1 `crexec`

Ausführen von Programmen in einer chroot-Umgebung. Das Programm erwartet als Parameter den Usernamen, unter dem die Programme gestartet werden sollen, den Namen des Directories, das das neue Root-Verzeichnis wird, und eine durch Klammeraffen "@" getrennte Liste von Programmen mit Parametern, die ausgeführt werden sollen.

Ein Beispiel: `crexec WWWuser /sec /bin/Mosaic @ /bin/fvwm`

A.1.2 `display`

Moving-Message-Display. XXX

A.1.3 `forw`

Ersatz für `rlogin`. Hier verwendet für den `rzstud`-Zugang, es kann aber auch für andere Rechner verwendet werden. Das Programm fragt nach dem Usernamen und entscheidet dann, welche `rzstud` für einen zuständig ist, und baut eine `rlogin`-Verbindung dahin auf.

Wenn ein ASCII-Terminal (siehe Kapitel 4.7, S. 10) an den NICK angeschlossen ist, wird das `forw` auch dort für den `rzstud`-Zugang und außerdem für den Menü-Zugang auf der `nz20` benutzt.

A.1.4 `logrec`

Dient zum Loggen von kurzen NICK-Meldungen (siehe Kapitel 8.2, S. 19). Das erste Argument wird per UDP an den Adminrechner geschickt, wo es durch das Programm `logrecd` zusammen mit der IP-Nummer des Rechners und der Zeit geloggt wird.

A.1.5 `menu`

Das Menu-Programm. Siehe XXX.

A.1.6 `timeout`

Das Programm `timeout` wird von `crexec` gleichzeitig mit `Mosaic` und dergl. gestartet. Es erfüllt mehrere Funktionen:

- Idle-Timeout. Die Anzahl der Sekunden wird als erstes Argument übergeben.
- Anzeige eines Hintergrundbildes. Das Filename des Bildes findet sich als zweites Argument.
- Exit-Button in der linken unteren Ecke des Bildschirms. Der Name des Icon-Files wird als drittes Argument erwartet.
- Anzeige des Timers links oben am Bildschirm. XXX.

A.1.7 `ticker`

Dieses Programm zeigt am unteren Bildschirmrand einen Lauf-Text ähnlich einem Nachrichten-Ticker an. Als Argumente erwartet es den Font, den es benutzen soll und einen Filenamen. Es läßt dann in einer Endlosschleife den Inhalt des Files über den Bildschirm laufen. Newlines in dem File werden dabei durch Leerzeichen ersetzt.

A.1.8 preloadfonts

Mosaic braucht eine ganze Reihe von Fonts. Diese werden normalerweise beim Starten von Mosaic in den X-Server geladen. Um die Ladezeit von Mosaic abzukürzen, werden die Fonts mit dem Programm *preloadfonts* schon direkt nach dem Anzeigen des Menus geladen. Das erspart ein paar Sekunden Ladezeit. Als einziges Argument erwartet das Programm den Filenamen der Fontliste.

A.2 Sonstige Programme

Die folgenden Programme werden auf den NICKs verwendet. Viele sind speziell für den Gebrauch auf den NICKs modifiziert. Die folgenden Kommentare erwähnen nur die wichtigsten Dinge zu den einzelnen Programmen. Weitere Informationen gibt es zu den meisten Programmen auf dem NICK-Entwicklungssystem unter */nick/src/Programmname/README.NICK*.

A.2.1 fvwm

Der Window-Manager *fvwm* wird für den WWW-, den rzstud-Zugang verwendet. XXX

A.2.2 mosaic

Mosaic ist der benutzte Web-Browser. Er wurde **stark** modifiziert, weil das Standard-Mosaic grauenhaft unsicher ist. Außerdem mußten die Menus und Tastenbelegungen an die eingeschränkte Funktionalität angepaßt werden. Alle Änderungen werden durch `#ifdef NICK` oder `#ifndef NICK` eingeleitet.

Einige Änderungen im einzelnen:

- Alle Aufrufe von *system()* wurden in eine eigene Routine umgeleitet, die *fork* und *exec* benutzt.
- URLs vom Typ "telnet", "rlogin", "mailto" und "file" wurden deaktiviert.
- XXX

A.2.3 mpeg_play

Dieses Programm dient zum Anzeigen von Filmen im mpeg-Format. Es wird als "helper application" von Mosaic aufgerufen, wenn es einen Film abspielen möchte.

A.2.4 mount

Mount-Befehl, der die Option "-n" unterstützt. Das ist wichtig, weil das Root-Filesystem ja Read-Only ist und deshalb die Datei */etc/mtab* nicht geschrieben werden kann.

A.2.5 rclock

Analog-Uhr.

A.2.6 rxvt

Terminalemulator. Wird vom rzstud-Zugang verwendet.

A.2.7 ghostscript

Interpreter für Postscript-Dateien. Wird von Ghostview verwendet.

A.2.8 ghostview

Programm zum Anzeigen von Postscript-Dateien. Änderungen im C Source und im Application-Ressource File. Verschiedene Menus und Tastenbelegungen wurden ausgebaut, damit man keine Files speichern kann und dergl.

A.3 Shell-Skripts

Auf den NICKs sind eine Reihe von kleinen Programmen und Skripten installiert um dem Administrator das Leben leichter zu machen:

A.3.1 */usr/sbin/nickreboot*

Dieses Skript muß als *root* gestartet werden. Es dient zum Rebooten des NICK-Rechners (z.B. nach einem Software-Update). Wenn auf der Kommandozeile das Wort "now" als Parameter übergeben wird, wird der Rechner sofort rebootet. Andernfalls wird er nur dann rebootet, wenn der NICK z.Z. im Menu ist, d.h. wenn ihn niemand benutzt. Wenn der NICK gerade benutzt wird, wird der Reboot verzögert, bis der Benutzer den Rechner verläßt (genauer: bis zu dem Zeitpunkt kurz bevor das Menu wieder angezeigt wird).

A.3.2 */sbin/mountro* und */sbin/mountrw*

Mit diesen Befehlen kann man eine Platte Read-Only (bzw. Read-Write) remounten. Als einziges Argument wird die Partition bzw. der Mount-Point verlangt. (In Wirklichkeit ist das nur ein kleines Skript jeweils, damit man sich die Optionen von *mount* nicht merken muß.)

A.3.3 */usr/nick/alive*

Dieses Skript wird vor dem ersten Anzeigen des Menus gestartet und läuft die ganze Zeit. Es sorgt dafür, das alle 15 Minuten eine "alive" Meldung an den Log-Host geschickt wird und holt sich außerdem alle Stunde die Uhrzeit von einem Timeserver.

A.3.4 */usr/nick/xinitrc*

Dieses Skript zeigt das Menu an und wertet die Rückgabe des Menu-Programms aus und ruft dann den gewünschten Menüpunkt auf.