

Dokumentation zum Setup des Domain-Nameservers der Fachhochschule Regensburg

© Hubert Feyrer, Februar 1998

Inhaltsverzeichnis

1	Einleitung	2
2	Systeminstallation.....	2
3	Installation der BIND-Software	2
3.1	Quelle.....	2
3.2	Entpacken und Patch anbringen.....	2
3.3	Konfiguration bind V4.9.3 P11	2
3.4	Übersetzen und installieren.....	4
4	Setup Primary	4
5	Setup Secondary	5
6	Administrivia.....	5
6.1	„Offizielles“ Anmelden des Nameservers beim DFN	5
6.2	Manuelles Updaten des Nameservers	6
6.3	Halbautomatisches Updaten des Nameservers	6
6.4	Vollautomatisches Updaten des Nameservers	6
7	Intern-Dokumentation	6
7.1	Format der internen DNS-Datenbank	6
7.2	Erzeugen der Zonen-Files	7
7.3	Das serial-File	8
7.3.1	Das SOA-File	8
7.3.2	Fine-Tuning des Umwandlungsprozesses.....	8
7.4	Sonstige Files	9
7.5	Benachrichtigen des named-Prozesses	9
7.6	Automatisches Updaten der Datenbank nach FTP-Upload	9
7.7	Automatisches Updaten der Datenbank via WWW	10
A	Verwaltungs-Dateien	11
A.1	Makefile	11
A.2	SOA.....	13
A.3	db.cache	14
A.4	db2db	15
A.5	etc_namedb_Makefile	18
A.6	grabdb	19
A.7	named.boot	20
A.8	update_serial.....	20
A.9	/soft/bind-4.9.3-P1/install/bootscrip	21
A.10	dump-access-db.....	21
A.11	prepare-grabdb-via-access	23
B	Beispiele für Datenbank-Dateien.....	24
B.1	DNS-Datenbank	24
B.2	localhost-Datenbank.....	24
B.3	db.fh-regensburg.de.sed.....	24
B.4	db.fh-regensburg.de.add	24

C Beispiele für erzeugte Dateien	26
C.1 db.127.0.0.....	26
C.2 db.194.95.104.....	26
C.3 db.fh-regensburg.de	26

8

1. Einleitung

Dieses Dokument beschreibt die Einrichtung des Nameservers der FH Regensburg, die Schritte zur Wartung desselbigen sowie die Funktionsweise der dazu eingesetzten Werkzeuge. Dieses Dokument ist keine Einführung zur BIND-Software oder zum Betrieb eines DNS-Servers als solches. Nähere Informationen zu diesem Thema sind dem folgenden Buch zu entnehmen:

Paul Albitz, Cricket Liu: DNS and Bind;
O'Reilly & Associates, Inc.; Sebastopol 1992.
ISBN 1-56592-010-4.

2. Systeminstallation

- Hardware: Sun SparcStation 4, 32 MB RAM
- Betriebssystem: Solaris 2.5

Bei der Installation des Betriebssystems ist darauf zu achten, daß die Partition, auf der /var liegt, nicht zu klein bemessen wird. d.h. / auf 250MB setzen.

3. Installation der BIND-Software

3.1 Quelle

- `ftp.nic.de :/pub/tools/bind/release/4.9.3/bind-4.9.3-REL.tar.gz`
- `ftp.nic.de :/pub/tools/bind/release/4.9.3/Patch1`

3.2 Entpacken und Patch anbringen

- `mkdir bind-4.9.3-P1`
- `cd bind-4.9.3-P1`
- `tar zvxf ../bind-4.9.3-REL.tar.gz`
- `patch <../Patch1`
- `rm -f Makefile.orig`

3.3 Konfiguration bind V4.9.3 P11

- do-configure:


```
# Quelle: ftp.nic.de:/pub/tools/bind/release/4.9.3/bind-4.9.3-REL.tar.gz
# Quelle: ftp.nic.de:/pub/tools/bind/release/4.9.3/Patch1

if [ "$BASE" = "" ]; then
    BASE=/soft
fi
if [ "$PKG" = "" ]; then
    PKG=bind-4.9.3-P1
fi
```

```

chmod +w Makefile

if [ ! -f Makefile.orig ]; then
    cp Makefile Makefile.orig
else
    cp Makefile.orig Makefile
fi

rm -f Makefile.BAK
cp Makefile Makefile.BAK
chmod +w Makefile

sed -e "s|^#\*(CC\).*=.*$|\1 = gcc $(CPPFLAGS) -Xlinker -R$(DESTLIB)|" \
    \
    -e "s|^#\*(CDEBUG\).*=.*$|\1 = -O3|" \
    -e "s|^#\*(DESTBIN\).*=.*$|\1 = $BASE/$PKG/bin|" \
    -e "s|^#\*(DESTSBIN\).*=.*$|\1 = $BASE/$PKG/bin|" \
    -e "s|^#\*(DESTEXEC\).*=.*$|\1 = $BASE/$PKG/bin|" \
    -e "s|^#\*(DESTMAN\).*=.*$|\1 = $BASE/$PKG/man|" \
    -e "s|^#\*(DESTHELP\).*=.*$|\1 = $BASE/$PKG/lib|" \
    -e "s|^#\*(DESTLIB\).*=.*$|\1 = $BASE/$PKG/lib|" \
    -e "s|^#\*(DESTINC\).*=.*$|\1 = $BASE/$PKG/include|" \
    -e "s|^#\*(PIDDIR\).*=.*$|\1 = /etc|" \
    -e "s|^#\*(LEX\).*=.*$|\1 = lex|" \
    -e "s|^#\*(CPPFLAGS\).*=.*$|\1 = -DSVR4 -D_SYS_STREAM_H $(SOLCOM-
    PAT)|" \
    \
    -e "s|^#\*(INDOT\).*=.*$|\1 = in.|" \
    -e "s|^#\*(XFER\).*=.*$|\1 = |" \
    -e "s|^#\*(LIBS\).*=.*$|\1 = -ll -lnsl -lsocket|" \
    -e "s|^#\*(LDS\).*=.*$|\1 = :|" \
    -e "s|^#\*(INSTALL\).*=.*$|\1 = /usr/ucb/install|" \
    -e "s|^#\*(RANLIB\).*=.*$|\1 = :|" \
    -e "s|^#\*(ARPPREF\).*=.*$|\1 = \\'order|" \
    -e "s|^#\*(ARSUFF\).*=.*$|\1 = \\'tsort\'|" \
    -e "s|^#\*(CATEXT\).*=.*$|\1 = \\'$\\$\\$N|" \
    -e "s|^#\*(PS\).*=.*$|\1 = ps -p|" \
    -e "s|^#\*(IOT\).*=.*$|\1 = IOT|" \
    -e "s|^#\*(SHRES\).*=.*$|\1 = shres/solaris|" \
    -e "s|^#\*(SHCC\).*=.*$|\1 = gcc -DSVR4 -D_SYS_STREAM_H|" \
    -e "s|^#\*(PIC\).*=.*$|\1 = -fpic|" \
    -e "s|^#\*(SHLD\).*=.*$|\1 = /usr/ccs/bin/ld -G|" \
    <Makefile.BAK >Makefile

```

- **do-make:**

```
make
```

- **do-install:**

```

mkdir -p $BASE/$PKG/lib
mkdir -p $BASE/$PKG/bin
mkdir -p $BASE/$PKG/man/cat1
mkdir -p $BASE/$PKG/install
mkdir -p $BASE/$PKG/include/arpa

```

```
make -n install | sed -e 's/-o bin//g' -e 's/-g bin//g' | sh -v
```

```

sed 's/^X //' >$BASE/$PKG/install/bootscript <<'EOF'
X #!/bin/sh
X

```

```

X PATH=/usr/bin:/bin
X BASE=/soft/bind-4.9.3-P1
X VERS=`basename $BASE`
X
X case $1 in
X 'start')
X     if [ -f /etc/named.pid ]; then
X         echo "Killing old (Solaris) named."
X         kill `cat /etc/named.pid` >/dev/null 2>&1
X     fi
X     if [ -x $BASE/bin/in.named ]; then
X         $BASE/bin/in.named
X     fi
X     echo "Domain Name Service ($VERS) started."
X     ;;
X 'stop')
X     if [ -f /var/tmp/named.pid ]; then
X         kill `cat /var/tmp/named.pid` >/dev/null 2>&1
X     fi
X     echo "Domain Name Service ($VERS) stopped."
X     ;;
X *)
X     echo "usage: $0 {start|stop}"
X     ;;
X esac
EOF

```

3.4 Übersetzen und installieren

- sh do-configure
- sh do-make
- sh do-install

4. Setup Primary

Primary: *rffis0004* (IP: 194.95.104.1)

Für ein Anlaufen des Nameservers sind von den in Anhang A aufgelisteten Dateien mindestens die folgenden (in `/etc/namedb`) nötig:

- db.cache
- etc_namedb_Makefile (als Makefile)
- named.boot

Außerdem werden (ebenfalls in `/etc/namedb`) die folgenden Dateien benötigt, die mit Hilfe diverser an anderer Stelle (s.u.) dokumentierter Tools erzeugt werden können:

- db.fh-regensburg.de
- db.127.0.0
- db.194.95.104
- db.194.95.105
- db.194.95.106
- db.194.95.107
- db.194.95.108

- db.194.95.109
- db.194.95.132

Zum Starten/Updaten des name-Daemons ist ausserdem der Befehl `/etc/namedb/suid/kill` nötig. Ist all dies vorhanden, dann kann der Nameserver entweder durch einen reboot oder alternativ mit einem der beiden folgenden Befehl gestartet werden:

```
cd /etc/namedb ; make start
```

bzw.

```
sh /etc/init.d/bind-4.9.3-P1 start
```

5. Setup Secondary

- Secondaries:
Momentan keine

Um den Nameserver-Daemon auch auf dem Secondary sofort beim Booten zu starten sind folgende Befehle unter Solaris nötig:

- `ln -s /soft/bind-4.9.3-P1/install/bootscript /etc/init.d/bind-4.9.3-P1`
- `ln /etc/init.d/bind-4.9.3-P1 /etc/rc3.d/S95bind-4.9.3-P1`
- `ln /etc/init.d/bind-4.9.3-P1 /etc/rc3.d/K05bind-4.9.3-P1`

Die Datei `bootscript` ist in Anhang A.9 zu finden. Weiterhin werden die folgenden Dateien benötigt:

- `/etc/namedb/named.boot`:


```
directory /etc/namedb
primary 0.0.127.in-addr.arpa db.127.0.0
secondary fh-regensburg.de 194.95.104.1 db.fh-regensburg.de
secondary 104.95.194.in-addr.arpa 194.95.104.1 db.194.95.104
secondary 105.95.194.in-addr.arpa 194.95.104.1 db.194.95.105
secondary 106.95.194.in-addr.arpa 194.95.104.1 db.194.95.106
secondary 107.95.194.in-addr.arpa 194.95.104.1 db.194.95.107
secondary 108.95.194.in-addr.arpa 194.95.104.1 db.194.95.108
secondary 109.95.194.in-addr.arpa 194.95.104.1 db.194.95.109
secondary 132.95.194.in-addr.arpa 194.95.104.1 db.194.95.132
cache . db.cache
```
- `/etc/namedb/db.127.0.0`: wie Primary
- `/etc/namedb/db.cache`: wie Primary

6. Administrivia

6.1 „Offizielles“ Anmelden des Nameservers beim DFN

Kontakt zum Anmelden beim DFN ist Dr. Jürgen Rauschenbach (Tel.: +49 30 884299-46, EMail: jrau@dfn.de, WWW: <http://www.dfn.de/adressen/rauschenbach.html>). Bekantztugeben sind:

- Name & IP-Nummer des Primarys: *dns.fh-reg.ensburg.de*, 194.95.104.1
- Name & IP-Nummer des Secondarys (soweit vorhanden!)

Weiterhin ist der Nameservers des DFN, *deneb.dfn.de*, als Secondary einzutragen.

6.2 Manuelles Updaten des Nameservers

Updates sind in der Datei „work/DNS-Datenbank“ zu machen, anschließend ist mittels „make install“ die DNS-Konfiguration zu aktualisieren.

6.3 Halbautomatisches Updaten des Nameservers

Das Vorgehen zum Updaten der DNS-Datenbank (Format siehe Kapitel 7.1 bzw. Anhang B) via FTP ist folgendes:

- Einloggen via FTP (Login: *dnsadmin*, Passwort: **geheim**)
- ins incoming-Verzeichnis wechseln: `cd incoming`
- incoming/dnsdb.lock erzeugen: `put somefile dnsdb.lock`
- incoming/dnsdb uploaden: `put somefile dnsdb`
- incoming/dnsdb.lock löschen: `delete dnsdb.lock`
- FTP beenden: `bye`

Das Verriegeln des Uploads mit der *.lock*-Datei soll verhindern, das eine unvollständige Datei in den Nameserver übernommen wird.

Zum Updaten des Nameservers läuft alle paar Stunden ein Cron-Job „grabdb“, der evtl. vorhandene Datenbank-Updates übernimmt, die Dateien für den Nameserver erzeugt und diesen mit den neuen Daten füttert („make install“). Siehe hierzu auch Kapitel 7.6.

6.4 Vollautomatisches Updaten des Nameservers

Die Daten der FH-Rechner werden in einer Access-Datenbank mit Web-Interface gehalten, von wo aus die DNS-Datenbank generiert und abgefragt werden muß, um anschließend via FTP ins entsprechende Verzeichnis abgelegt zu werden, wo sie dann weiterverarbeitet werden. Alle diese Schritte können auf die folgende Weise vereinfacht werden:

```
~/work/prepare-grabdb-via-access
```

7. Intern-Dokumentation

7.1 Format der internen DNS-Datenbank

Die Datenbank besteht aus einer ASCII-Datei (7bit, Zeilenende ohne ^M), die einzelne durch eine Leerzeile getrennte Records enthält. Ein Record besteht wiederum aus mehreren Zeilen, mit einem Feld pro Zeile. Zeilen, die mit '#' beginnen gelten als Kommentarzeilen und werden bis zum Zeilenende überlesen. Die folgenden Felder existieren:

- **Name:** Offizieller (canonical) Name; Kann mehrfach vorkommen, falls ein Host mehrere Ethernet-Interfaces hat; zwingend notwendig.
- **Alias:** Alias, der nur 1x auf dem FH-Campus vorkommt; optional

- IP-Adresse: Eindeutige IP-Adresse in Dotted-Quad-Darstellung; zwingen notwendig.
- Rechnertyp: Rechnertyp bzw. Hersteller, relativ freies Format; empfohlen. Beispiele:
 - PC
 - Sun SparcStation 5
 - SGI IRIS Indigo
 - DEC AlphaStation
- Betriebssystem: Angaben zum Betriebssystem, relativ freies Format; empfohlen. Beispiele:
 - NetWare 3.x
 - NetWare 4.x
 - DOS/Windows
 - Solaris 2.x
 - SunOS 4.x
 - HPjUX 10.x

Siehe Anhang B für ein vollständiges Beispiel.

7.2 Erzeugen der Zonen-Files

Die Datei "localhost-Datenbank" wird zum Erzeugen der Zonen-Datei "db.127.0.0" verwendet, die eigentlich nur dazu da ist, die IP-Adresse 127.0.0.1 in den Namen "*localhost*" aufzulösen.

Die Zonen-Dateien "db.fh-regensburg.de", "db.194.95.104", "db.194.95.105", "db.194.95.106", "db.194.95.107", "db.194.95.108" und "db.194.95.109" werden aus der Datei "DNS-Datenbank" erzeugt, wobei erstere zum Auflösen der Namen in Adressen (resolving), die restlichen zum umgekehrten Ermitteln der Hostnamen aus den IP-Adressen (reverse resolving) verwendet werden.

Welche Zonen-Datei für welche Zone genau verwendet wird steht auch im named.boot-File (siehe Anhang A.7).

Neu erzeugt werden die Zone-Files, wenn eine der folgenden Dateien verändert wurden:

- localhost-Datenbank
- DNS-Datenbank
- SOA

Das Erzeugen der Zonen-Dateien erfolgt durch das Programm db2db und geht wie folgt vor sich:

1. Der für alle Zonen-Dateien gleichen Header (Start Of Authority, SOA) wird aus der Datei SOA eingelesen, und `%serial%` durch den Inhalt des serial-Files (s.u.) ersetzt
2. Das angegebene Datenbank-File (localhost-Datenbank bzw. DNS-Datenbank) wird zeilenweise eingelesen
3. Wurde auf der Befehlszeile (db2db-Aufruf) ein IP-Netz angegeben, so wird eine Datei für Reverse-Mapping erzeugt, ansonsten eine Datei zum Auflösen von Hostnamen in IP-Nummern.
- 3.1. Ist eine Zonen-Datei für "normales" Resolving zu erstellen, so werden die folgenden Felder aus der angegebenen Eingabe-Datenbank wie folgt umgesetzt:
 - aus "IP-Adresse" und "Name" wird der A-Record gebaut

- ist ein Feld "Alias" angegeben, so wird dieses als CNAME verwendet, offizieller (canonical) Name ist, was im "Name"-Feld angegeben ist
- sind "Rechnertyp" und "Betriebssystem" angegeben, so werden diese für den HINFO-Record abgestellt. Sind Leerzeichen in einem der beiden Felder enthalten, so wird das ganze Feld automatisch mit ' 's gequotet.

Ein Beispiel für eine so erzeugte Zonen-Datei gibt's in Anhang C.3.

- 3.2. Ist eine Zonen-Datei für Reverse-Resolving zu erstellen, so wird das auf der Befehlszeile angegeben IP-Netz vom Inhalt des Feldes "IP-Adresse" entfernt, und das resultierende (4.) Host-Byte zusammen mit dem Inhalt des Feldes "Name" in einem PTR-Record abgestellt. Beispiele gibt's in Anhang C.1 und C.2.

7.3 Das serial-File

Das Serial-File (bezeichnender Name: "serial") wird beim Erzeugen der Zonen-Dateien ausgelesen und in das entsprechende Feld im SOA-Record eingesetzt, nachdem es durch das Programm "update_serial" upgedated wurde.

Die Datei enthält eine 8stellige Zahl, die sich aus dem (jeweils zweistelligen) Jahres-, Monats- und Tagesdatum sowie einer laufenden (zweistelligen) Nummer für Updates innerhalb eines Tages zusammensetzt.

Entsprechend wird, falls das Datum, an dem "update_serial" aufgerufen wird, dasselbe wie im serial-File ist, der Counter der letzten Stelle um 1 erhöht. Unterscheiden sich die beiden Daten, so wird das aktuelle Datum gesetzt und der Counter auf 00 gesetzt.

(Auf die Bedeutung des Serial-Wertes beim DNS soll hier nicht näher eingegangen werden und ist in der einschlägigen Literatur nachzulesen)

7.3.1 Das SOA-File

Die SOA-Datei stellt den Start-Of-Authority-Header (SOA) für alle erzeugten Zonen-Dateien bereit. Ist eine Änderung an dieser Stelle nötig, so ist die nur einmal an dieser zentralen Datei zu machen (gefolgt von einem "make install").

7.3.2 Fine-Tuning des Umwandlungsprozesses

Um gewisse Details beim Erzeugen der DNS-Dateien (Nameserver muß immer "dns" heißen und als solches einen A-Record haben, MX-Records, etc.) zu korrigieren, können für jede erzeugte Datei noch zwei weitere Dateien mit den Endungen ".sed" bzw. ".add" abgestellt werden.

Ist nach dem Erzeugen einer Zonen-Datei eine zugehörige .sed-Datei vorhanden, so wird die Zonendatei mit den in der .sed-Datei enthaltenen Anweisungen durch das Unix-Programm sed nachbearbeitet. Auf diese Weise können Zeilen gelöscht oder Werte ersetzt werden. Details siehe sed-Man-Page.

Ist anschließend noch eine .add-Datei vorhanden, so wird deren Inhalt unverändert an die soeben erzeugte Zonen-Datei angehängt. Da hier keine weitere Überprüfung auf mehrfaches Vorkommen von Namen, IP-Nummern, etc. vorgenommen wird ist hier besondere Vorsicht walten zu lassen.

Vor dem Verändern/Erstellen/Löschen bitte ich auf jeden Fall, mich (Hubert Feyrer <hubert.feyrer@rz.uni-regensburg.de>) zu kontaktieren!

7.4 Sonstige Files

Um alle wichtigen Dateien in einem Verzeichnis zu halten liegen auch die beiden Dateien `etc_namedb_Makefile` (siehe A.5) und `named.boot` (siehe A.7) in `~dnsadmin/work`, und werden erst beim `"make install"` nach `/etc/namedb/Makefile` bzw. `.../named.boot` kopiert.

Mithilfe des Makefiles kann der Name-Daemon gestartet (`"make start"`), gestopt (`"make stop"`) und neu gestartet (`"make restart"`) werden. Zusätzlich kann der für den `dnsadmin` zum updaten des Name-Daemons nötige `kill`-Befehl erzeugt werden, siehe 7.5.

Wie schon unter 7.2 erwähnt, ordnet die Datei `named.boot` zu, welche Zonen-Datei für welche Zone zuständig ist. Sie wird ebenfalls nur der Vollständigkeit halber in `~dnsadmin/work` gehalten, damit sie bei den restlichen Config-Dateien liegt und zusammen mit diesen in `/etc/namedb` installiert werden kann.

7.5 Benachrichtigen des named-Prozesses

Die Benachrichtigung des Name-Daemons zum Erneuern seines Datenbestandes etc. geschieht über Signale. Da der `named`-Prozess unter der Kennung des Systemverwalters (`root`) läuft, muß das entsprechende Signal (`SIGHUP`, etc.) auch als `root` abgesetzt werden. Damit jedoch der Update-Prozess auch von einer "normalen" Benutzerkennung aus durchgeführt werden kann, wird ein eigener `"kill"`-Befehl verwendet, der unabhängig von der UID des Aufrufers immer als `root` läuft (`setuid root`).

Dieser `kill`-Befehl steht im normalen Betrieb in `/etc/namedb/suid/kill` und kann auch nur vom DNS-Verwalter ausgeführt werden, da nur dieser die Zugriffsrechte dazu besitzt (das `suid`-Verzeichnis ist Owner `dnsadmin`, Mode `0700`). Erzeugt werden kann der Befehl als `root` mit folgenden Befehlen:

- `cd /etc/namedb`
- `rm -fr suid`
- `make suid/kill`

Da es dieser Befehl erlaubt, jeden beliebigen Prozeß im System empfindlich zu stören, sollte er nur im Rahmen der hier beschriebenen Programme verwendet werden.

Die Möglichkeit, den `suid/kill`-Befehl mittels eines `setuid-root` gesetzten `/bin/kill`-Befehl zu realisieren scheiterte leider, da der `/bin/kill` ein `ksh`-Script ist und den eingebauten `kill`-Befehl der Korn-Shell verwendet. Setzt man dieses Script `setuid-root`, so hat man keinen `setuid-kill`-Befehl, sondern eine Shell, die `setuid-root` ist, was eine weitaus größere Sicherheitslücke darstellen würde. Aus diesen Gründen ist der `suid/kill`-Befehl als `perl`-Script realisiert, was auch der Grund dafür ist, daß das zu sendende Signal immer und ohne führendem `'-'` anzugeben ist.

7.6 Automatisches Updaten der Datenbank nach FTP-Upload

Der zugehörige `crontab`-Eintrag der `dnsadmin`-Kennung sieht wie folgt aus:

```
0 * * * * /usr/home/dnsadmin/work/grabdb
```

d.h. jede volle Stunde läuft der Job "grabdb" los, sieht in ~dnsadmin/incoming nach, ob eine neue dns-Datenbank vorhanden ist, und aktiviert sie. Falls dies geschieht, so wird - wie bei Cron-Jobs üblich - der Output via Mail an den Besitzer des Cron-Jobs gesandt, in diesem Fall dnsadmin. Da dies aber eine reine Admin-Kennung ist, ist in der Datei ~dnsadmin/forward noch die EMail-Adresse der Leute angegeben (durch "," getrennt), die den Output der grabdb-Läufe erhalten sollen.

7.7 Automatisches Updaten der Datenbank via WWW

Da der Weg via den FTP-Upload umständlich und langsam ist wurde noch das Script 'dump-access-db' geschrieben, das die ansonsten manuell vorzunehmende Abfrage der DNS-Datenbank via Access & WWW automatisiert und die aktuelle Version derselben ausgibt. Durch das Script 'prepare-grabdb-via-access' wird damit der gesamte Update-Prozess erheblich verkürzt und vereinfacht. Ein einfacher Aufruf ohne Argumente reicht, um den Nameserver auf den neuesten Stand zu bringen.

A Verwaltungs-Dateien

A.1 Makefile

```

#
# Makefile fuer den Nameserver
#
# 19960416 HFininit
# 19960602 HF.add und .sed-Dateien fuer alle NS-Dateien
#

DB          = DNS-Datenbank
DB_local=  localhost-Datenbank
DBfiles     = db.fh-regensburg.de \
              db.194.95.104 \
              db.194.95.105 \
              db.194.95.106 \
              db.194.95.107 \
              db.194.95.108 \
              db.194.95.109 \
              db.194.95.132 \
              db.127.0.0
DBdir       = /etc/namedb
SERIAL      = serial
SOA         = SOA

#####

all: $(DBfiles)

#####

$(SERIAL): $(DB) $(DB_local) $(SOA)
        ./update_serial $(SERIAL)

###

db.fh-regensburg.de: $(SERIAL) db.fh-regensburg.de.add db.fh-
regensburg.de.sed
        ./db2db fh-regensburg.de $(SOA) $(SERIAL) $(DB)
        if [ -s $@.sed ]; then \
            mv $@ $@.tmp ;\
            sed -f $@.sed $@.tmp >$@ ;\
            rm $@.tmp ;\
        fi
        if [ -s $@.add ]; then \
            echo '' >>$@ ;\
            echo ';; Zusaetzliche Eintraege:' >>$@ ;\
            cat $@.add >>$@ ;\
        fi

db.127.0.0: $(SERIAL)
        ./db2db fh-regensburg.de $(SOA) $(SERIAL) $(DB_local) 127.0.0
        if [ -s $@.sed ]; then \
            mv $@ $@.tmp ;\
            sed -f $@.sed $@.tmp >$@ ;\
            rm $@.tmp ;\
        fi

```

```
if [ -s $@.add ]; then \  
    echo '' >>$@ ;\  
    echo ';; Zusaetzliche Eintraege:' >>$@ ;\  
    cat $@.add >>$@ ;\  
fi  
  
db.194.95.104: $(SERIAL)  
./db2db fh-regensburg.de $(SOA) $(SERIAL) $(DB) 194.95.104  
if [ -s $@.sed ]; then \  
    mv $@ $@.tmp ;\  
    sed -f $@.sed $@.tmp >$@ ;\  
    rm $@.tmp ;\  
fi  
if [ -s $@.add ]; then \  
    echo '' >>$@ ;\  
    echo ';; Zusaetzliche Eintraege:' >>$@ ;\  
    cat $@.add >>$@ ;\  
fi  
  
db.194.95.105: $(SERIAL)  
./db2db fh-regensburg.de $(SOA) $(SERIAL) $(DB) 194.95.105  
if [ -s $@.sed ]; then \  
    mv $@ $@.tmp ;\  
    sed -f $@.sed $@.tmp >$@ ;\  
    rm $@.tmp ;\  
fi  
if [ -s $@.add ]; then \  
    echo '' >>$@ ;\  
    echo ';; Zusaetzliche Eintraege:' >>$@ ;\  
    cat $@.add >>$@ ;\  
fi  
  
db.194.95.106: $(SERIAL)  
./db2db fh-regensburg.de $(SOA) $(SERIAL) $(DB) 194.95.106  
if [ -s $@.sed ]; then \  
    mv $@ $@.tmp ;\  
    sed -f $@.sed $@.tmp >$@ ;\  
    rm $@.tmp ;\  
fi  
if [ -s $@.add ]; then \  
    echo '' >>$@ ;\  
    echo ';; Zusaetzliche Eintraege:' >>$@ ;\  
    cat $@.add >>$@ ;\  
fi  
  
db.194.95.107: $(SERIAL)  
./db2db fh-regensburg.de $(SOA) $(SERIAL) $(DB) 194.95.107  
if [ -s $@.sed ]; then \  
    mv $@ $@.tmp ;\  
    sed -f $@.sed $@.tmp >$@ ;\  
    rm $@.tmp ;\  
fi  
if [ -s $@.add ]; then \  
    echo '' >>$@ ;\  
    echo ';; Zusaetzliche Eintraege:' >>$@ ;\  
    cat $@.add >>$@ ;\  
fi
```

```

db.194.95.108: $(SERIAL) db.194.95.108.sed db.194.95.108.add
./db2db fh-regensburg.de $(SOA) $(SERIAL) $(DB) 194.95.108
if [ -s $@.sed ]; then \
    mv $@ $@.tmp ;\
    sed -f $@.sed $@.tmp >$@ ;\
    rm $@.tmp ;\
fi
if [ -s $@.add ]; then \
    echo ' ' >>$@ ;\
    echo ';; Zusaetzliche Eintraege:' >>$@ ;\
    cat $@.add >>$@ ;\
fi

```

```

db.194.95.109: $(SERIAL)
./db2db fh-regensburg.de $(SOA) $(SERIAL) $(DB) 194.95.109
if [ -s $@.sed ]; then \
    mv $@ $@.tmp ;\
    sed -f $@.sed $@.tmp >$@ ;\
    rm $@.tmp ;\
fi
if [ -s $@.add ]; then \
    echo ' ' >>$@ ;\
    echo ';; Zusaetzliche Eintraege:' >>$@ ;\
    cat $@.add >>$@ ;\
fi

```

```

db.194.95.132: $(SERIAL)
./db2db fh-regensburg.de $(SOA) $(SERIAL) $(DB) 194.95.132
if [ -s $@.sed ]; then \
    mv $@ $@.tmp ;\
    sed -f $@.sed $@.tmp >$@ ;\
    rm $@.tmp ;\
fi
if [ -s $@.add ]; then \
    echo ' ' >>$@ ;\
    echo ';; Zusaetzliche Eintraege:' >>$@ ;\
    cat $@.add >>$@ ;\
fi

```

#####

```

install: $(DBfiles)
cp $(DBfiles) $(DBdir)
cp named.boot $(DBdir)
cp db.cache $(DBdir)
cp etc_namedb_Makefile $(DBdir)/Makefile
( cd /etc/namedb ; make restart )

```

#####

```

clean:
-rm $(DBfiles)
-rm serial

```

A.2 SOA

```

@ IN          SOA  dns.fh-regensburg.de.      feyrer.dns.fh-regensburg.de. (
                                %serial%; serial

```

```

                28800; refresh
                7200; retry
                604800; expire
                172800; default ttl
            )

@ IN      NS    dns.fh-regensburg.de.
@ IN      NS    rrzs2.rz.uni-regensburg.de.
@ IN      NS    deneb.dfn.de.

```

A.3 db.cache

```

;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache . <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC registration services
;       under anonymous FTP as
;       file           /domain/named.root
;       on server      FTP.RS.INTERNIC.NET
;       -OR- under Gopher at  RS.INTERNIC.NET
;       under menu     InterNIC Registration Services (NSI)
;       submenu        InterNIC Registration Archives
;       file           named.root
;
;       last update:   Aug 22, 1997
;       related version of root zone:  1997082200
;
;
; formerly NS.INTERNIC.NET
;
.           3600000   IN   NS       A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.  3600000   A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.           3600000   NS     B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.  3600000   A     128.9.0.107
;
; formerly C.PSI.NET
;
.           3600000   NS     C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.  3600000   A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.           3600000   NS     D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.  3600000   A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.           3600000   NS     E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.  3600000   A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.           3600000   NS     F.ROOT-SERVERS.NET.

```

```

F.ROOT-SERVERS.NET.      3600000      A      192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                          3600000      NS      G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.      3600000      A      192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                          3600000      NS      H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A      128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                          3600000      NS      I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A      192.36.148.17
;
; temporarily housed at NSI (InterNIC)
;
.                          3600000      NS      J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.      3600000      A      198.41.0.10
;
; housed in LINX, operated by RIPE NCC
;
.                          3600000      NS      K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.      3600000      A      193.0.14.129
;
; temporarily housed at ISI (IANA)
;
.                          3600000      NS      L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.      3600000      A      198.32.64.12
;
; housed in Japan, operated by WIDE
;
.                          3600000      NS      M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A      202.12.27.33
; End of File

```

A.4 db2db

```
#!/soft/local/bin/perl5
```

```

#
# History:
# 19960428 HF   Keine MAC-Adressen mehr in Datenbank-File
#

if ( $#ARGV < 3 ){
    print STDERR "Usage: $0 domain SOA-template serial-file DB-file\n";
    print STDERR "or:   $0 domain SOA-template serial-file DB-file IP-Net\n";
}

$domain    = $ARGV[0];      # Domain, e.g. fh-regensburg.de
$SOA_f     = $ARGV[1];      # SOA-file
$serial_f  = $ARGV[2];      # File with serial in it
$DB_f      = $ARGV[3];      # File with Database in it
$ipnet     = $ARGV[4];      # Subnet-Address (IP)

```

```

#print "domain:    $domain\n";
#print "SOA_f:    $SOA_f\n";
#print "serial_f: $serial_f\n";
#print "DB_f:    $DB_f\n";
#print "ipnet:    $ipnet\n";

#
# Read serial# from file
#
open(IN,"<$serial_f")
    || die "Can't read $serial_f: $!\n";
chomp($serial=<IN>);
close(IN);

#
# Open output file
#
if(!$ipnet){
    # Forward Mapping
    open(OUT,">db.$domain")
        || die "Can't write db.$domain: $!\n";
}else{
    # Reverse Mapping
    open(OUT,">db.$ipnet")
        || die "Can't write db.$ipnet: $!\n";
}

#
# SOA
#
open(IN,"<$SOA_f")
    || die "Can't read $SOA_f: $!\n";
while(<IN>){
    s/%serial%/$serial/g;
    print OUT $_;
}
close(IN);
print OUT "\n";

#
# Aliase & Hostnamen einlesen & merken
#
undef %HNAME;
undef %HALIAS;
open(DB,"$DB_f")
    || die "Can't read $DB_f: $!\n";
while(<DB>){
    chomp;
    ($kw,$val)=/^[^:]+\:\s*(.*)\s*$/;
    if($kw eq "Name" and ($val!~/\s+/) and ($val ne "")){
        $HNAME{$val}++;
    }elseif($kw eq "Alias" and ($val!~/\s+/) and ($val ne "")){
        $HALIAS{$val}++;
    }
}

```



```
    }
}
close(DB);

#
# The hosts themselves
#
open(DB,"$DB_f")
    || die "Can't read $DB_f: $!\n";

$loaded=0;
$lineno=0;
undef %H;
while(!eof(DB)){
    while(!eof(DB) and ($line=<DB>!~/^\s*$/)){
        $lineno++;
        chomp($line);
        next if $line=~/^#/;

        ($kw,$val)=($line=~/^([\^:]+):\s*(.*)\s*$/);

        if(    $kw eq "Name"){
            if($val!~/^[a-zA-Z_-][-.a-zA-Z0-9_]*$/){
                die "Bogus Hostname: '$val' near line $lineno.\n";
            }

            # Name eindeutig?
            if($HNAME{$val}+$HALIAS{$val} > 1){
                die "Multiple hostname/alias '$val' near line $lineno.\n";
            }
        }
        elsif($kw eq "Alias"){
            if($val ne "" and $val!~/^[a-zA-Z_-][-.a-zA-Z0-9_]*$/){
                die "Bogus Alias: '$val' near line $lineno.\n";
            }

            # Name eindeutig?
            if($HNAME{$val}+$HALIAS{$val} > 1){
                die "Multiple hostname/alias '$val' near line $lineno.\n";
            }
        }
        elsif($kw eq "IP-Adresse"){
            # Sanity check?
        }
        elsif($kw eq "Rechnertyp"){
            $val="\ "$val" " if $val=~/\s/;
        }
        elsif($kw eq "Betriebssystem"){
            $val="\ "$val" " if $val=~/\s/;
        }
        elsif($kw eq "Standort"){
            # Sanity check?
        }
        elsif($kw eq "Betreuer"){
            # Sanity check?
        }
    }
}
```

```

    }else{
        die "Unknown keyword '$kw' in '$line' near line $lineno.\n";
    }

    $H{$kw}=$val
    if $val !~ /^\\s*$/ and
        $val ne "\\\"";

    $loaded=1;
}

if($loaded){
    if(!$ipnet){
        # Forward-Mapping
        #
        printf OUT "%-20s \\t\\tIN A      $H{'IP-Adresse'}\\n", $H{'Name'}
if $H{'IP-Adresse'} ne "" and $H{'Name'} ne "";
        printf OUT "%-20s \\t\\tIN CNAME  $H{'Name'}\\n", $H{'Alias'}
        if $H{'Alias'} and $H{'Name'} ne "";
        printf OUT "%-20s \\t\\tIN HINFO  $H{'Rechnertyp'}
$H{'Betriebssystem'}\\n",
        $H{'Name'}
        if $H{'Rechnertyp'} and $H{'Betriebssystem'} and $H{'Name'}
ne "";

        print OUT "\\n";
    }else{
        # Reverse Mapping
        #
        printf OUT "%-20d \\t\\tIN PTR    $H{'Name'}.\\$domain.\\n\\n",
(split(/\\.\\./, $H{'IP-Adresse'}))[3]
        if $H{'IP-Adresse'}=~/$ipnet\\.\\./ and $H{'Name'} ne "";
    }
}
$lineno++;

undef %H;
}

close(DB);

close(OUT);

```

A.5 etc_namedb_Makefile

```

#NAMED = /usr/sbin/in.named
NAMED = /soft/bind-4.9.3-P1/bin/in.named
DNSADMIN = xfeyrer

#####

clean:
    -rm db.fh-regensburg.de db.19*

stop: suid/kill
    -suid/kill 15 `ps -ef | grep named | grep -v grep | awk '{print $$2}'`

start:
    ${NAMED}

```

```
restart: suid/kill
        -suid/kill 1 `ps -ef | grep ${NAMED} | grep -v grep | awk '{print $$2}'`

# Das folgende als 'root' starten, um ein suid/kill zu bauen, das
# nur von $DNSADMIN aufgerufen werden kann und (u.a.!) den Nameserver
# updaten kann.
suid/kill:
    rm -fr suid
    mkdir -m 700 suid
    chown ${DNSADMIN} suid
    echo '#!/soft/local/bin/perl5 -U' >>suid/kill
    echo 'kill @ARGV;' >>suid/kill
    chmod 4555 suid/kill
    chown root suid/kill
```

A.6 grabdb

```
#!/bin/sh

PATH=/bin:/usr/bin:/usr/ccs/bin
export PATH

cd $HOME/work

SRC_DB=../incoming/dnsdb
DST_DB=../DNS-Datenbank

if [ -f "$SRC_DB.lock" ]; then
    echo 'DNS-Transfer in Progress\c'

    sleep 10

    if [ -f "$SRC_DB.lock" ]; then
        echo ' - locked?'
        exit 1
    else
        echo ' - done.'
    fi
fi

if [ -f "$SRC_DB" ]; then
    echo 'Unterschiede (< alt, > neu):'
    diff $DST_DB $SRC_DB | sed 's/^/| /'
    echo ''

    rm -f $DST_DB.BAK
    mv $DST_DB $DST_DB.BAK
    mv $SRC_DB $DST_DB
    touch $DST_DB

    cd `dirname $DST_DIR`
    make install
fi
```

A.7 named.boot

```

;
; /etc/named.boot
;
; 19960102 HF
; 19960410 HF    Eintragen der Regensburger Zonen
;

directory /etc/namedb

primary      fh-regensburg.de          db.fh-regensburg.de
primary      0.0.127.in-addr.arpa      db.127.0.0
primary      104.95.194.in-addr.arpa   db.194.95.104
primary      105.95.194.in-addr.arpa   db.194.95.105
primary      106.95.194.in-addr.arpa   db.194.95.106
primary      107.95.194.in-addr.arpa   db.194.95.107
primary      108.95.194.in-addr.arpa   db.194.95.108
primary      109.95.194.in-addr.arpa   db.194.95.109
cache        .                          db.cache

secondaryuni-regensburg.de132.199.1.2   urrz.zone
secondary199.132.in-addr.arpa132.199.1.2 urrz.revzone

```

A.8 update_serial

```

#!/soft/local/bin/perl5

if($#ARGV<0){
    die "Usage: $0 serial-file\n";
}

$serial_f = $ARGV[0];

#
# Read serial# from file
#
if(open(IN,"<$serial_f")){
    chomp($serial=<IN>);
    close(IN);
}else{
    print STDERR "Warning: Can't read $serial_f: $!\n";
}

# Lock $serial_f or die

($day,$mon,$year)=((localtime(time))[3,4,5]);
$nser=sprintf "%02d%02d%02d%02d",
    ($year>95)?"19":"20",
    $year,$mon+1,$day;

($oser,$ocnt)=($serial=~/(.....)(..)/);

if($nser eq $oser){
    $ocnt;
    $serial=sprintf("%8s%02d",$oser,$ocnt+1);
}

```

```

}else{
    $serial=sprintf("%8s%02d",$nser,00);
}

if(!open(OUT,">$serial_f")){
    # Unlock $serial_f
    die "Can't update $serial_f: $!\n";
}
print OUT "$serial\n";
close(OUT);

# Unlock $serial_f

```

A.9 /soft/bind-4.9.3-P1/install/bootscrip

```

#!/bin/sh

PATH=/usr/bin:/bin
BASE=/soft/bind-4.9.3-P1
VERS=`basename $BASE`
PIDF=/etc/named.pid

case $1 in
'start')
    if [ -f /etc/named.pid ]; then
        echo "Killing old (Solaris) named."
        kill `cat $PIDF` >/dev/null 2>&1
    fi
    if [ -x $BASE/bin/in.named ]; then
        $BASE/bin/in.named
    fi
    echo "Domain Name Service ($VERS) started."
    ;;
'stop')
    if [ -f /etc/named.pid ]; then
        kill `cat $PIDF` >/dev/null 2>&1
    fi
    rm -f $PIDF
    echo "Domain Name Service ($VERS) stopped."
    ;;
*)
    echo "usage: $0 {start|stop}"
    ;;
esac

```

A.10 dump-access-db

```

#!/soft/local/bin/perl5

$debug=0;
$server="rfhnt8001";
$port=80;
$timeout=60;

use Socket;

#####

```

```

sub doTimeout
{
    print <<EOF;
    Sorry, server currently not responding. Please try
    again later! :-)<P>
    EOF

    exit(0);
}

#####
#####
##
## M A I N
##
#####
#####

{
    $sockaddr = 'S n a4 x8';
    chop($hostname = `hostname`);

    ($name, $aliases, $proto) = getprotobyname('tcp');
    ($name, $aliases, $port) = getservbyname($port, 'tcp')
        unless $port =~ /^d+$/;
    ($name, $aliases, $type, $len, $localaddr)
        = gethostbyname($hostname);
    ($name, $aliases, $type, $len, $remoteaddr)
        = gethostbyname($server);

    $local = pack($sockaddr, &AF_INET, 0, $localaddr);
    $remote = pack($sockaddr, &AF_INET, $port, $remoteaddr);

    socket(S, &PF_INET, &SOCK_STREAM, $proto) || die "socket: $!";
    bind(S, $local) || die "bind: $!";
    connect(S, $remote) || die "connect: $!";

    select(S); $| = 1; select(stdout);

    vec($sockWaitVec,fileno(S),1) = 1;

    #SIG{'ALRM'}=\doTimeout;
    #alarm($timeout);
}

print S <<EOF;
POST /db/namesrv.IDC HTTP/1.0
Content-type: application/x-www-form-urlencoded
Content-length: 12

neu=+NAMESRV
EOF

$inpre=0;
line:
while($l=<S>){
    $l=~s/[r\n]//g;

```

```
$l=~s/\s+$/g;
if($l=~/^\.PRE.$/){
    $inpre=1;
    next line;
}
print "$l\n" if $inpre;
}

alarm 0;

exit 0;
```

A.11 prepare-grabdb-via-access

```
#!/bin/sh

PATH=/bin:/usr/bin:/usr/ccs/bin
export PATH

cd $HOME/work

SRC_DB=../incoming/dnsdb

touch $SRC_DB.lock

./dump-access-db >/tmp/db.$$
cp /tmp/db.$$ $SRC_DB
rm /tmp/db.$$

rm $SRC_DB.lock
```



```
extern                IN MX      10 rfhs0002
*.mgate              IN MX      10 rfhs0002
*.pgate              IN MX      10 rfhs0002
@                    IN MX      50 rfhs0001
mailhub              IN MX      50 rfhs0001
rz                   IN MX      50 rfhs0001
informatik           IN MX      50 rfhs0001
mathematik           IN MX      50 rfhs0001
e-technik            IN MX      50 rfhs0001
mikro                IN MX      50 rfhs0001
bau                  IN MX      50 rfhs0001
maschinenbau         IN MX      50 rfhs0001
bwl                  IN MX      50 rfhs0001
soz                  IN MX      50 rfhs0001
architektur          IN MX      50 rfhs0001
bib                  IN MX      50 rfhs0001
verwaltung           IN MX      50 rfhs0001
stud                 IN MX      50 rfhs0001
extern               IN MX      50 rfhs0001
*.mgate              IN MX      50 rfhs0001
*.pgate              IN MX      50 rfhs0001

;; Well known services
whois                IN CNAMErfhs0003
ftp                  IN CNAMErfhs0002
www                  IN CNAMErfhs0002
www-stud             IN CNAMErfhs0005
homepages            IN CNAMErfhs0005

;; Intranet-Server - Diplomarbeit Christoph Metten
INTERN               IN CNAMErfhs0001

;; Stuff - finger weg!!! :- ) - HF
smaug                IN A194.95.108.11
rfhs8030             IN CNAMEsmaug
smaug                IN HINFO"SPARCSTATION 1" NetBSD
grizu                IN A194.95.108.8
rfhu0001             IN CNAMEgrizu
grizu                IN HINFO"Amiga 3000 UX" NetBSD
jabberwocky          IN A194.95.108.22
rfhu8002             IN CNAMEjabberwocky
jabberwock           IN CNAMEjabberwocky
jabberwocky          IN HINFO"IBM 6x86-P150+" NetBSD
```

C Beispiele für erzeugte Dateien

C.1 db.127.0.0

```
@      IN      SOA  dns.fh-regensburg.de.      feyrer.dns.fh-regensburg.de. (
                                1997090302; serial
                                28800; refresh
                                7200; retry
                                604800; expire
                                172800; default ttl
                                )

@      IN      NS   dns.fh-regensburg.de.
@      IN      NS   rrzs2.rz.uni-regensburg.de.
@      IN      NS   deneb.dfn.de.

1      IN      PTR   localhost.fh-regensburg.de.
```

C.2 db.194.95.104

```
@      IN      SOA  dns.fh-regensburg.de.      feyrer.dns.fh-regensburg.de. (
                                1996060600; serial
                                28800; refresh
                                7200; retry
                                604800; expire
                                172800; default ttl
                                )

@      IN      NS   dns.fh-regensburg.de.
@      IN      NS   rrzs2.rz.uni-regensburg.de.
@      IN      NS   deneb.dfn.de.

1      IN      PTR   dns.fh-regensburg.de.
11     IN      PTR   smaug.fh-regensburg.de.
```

C.3 db.fh-regensburg.de

```
@      IN      SOA  dns.fh-regensburg.de.      feyrer.dns.fh-regensburg.de. (
                                1996060600; serial
                                28800; refresh
                                7200; retry
                                604800; expire
                                172800; default ttl
                                )

@      IN      NS   dns.fh-regensburg.de.
@      IN      NS   rrzs2.rz.uni-regensburg.de.
@      IN      NS   deneb.dfn.de.

dns      IN      A      194.95.104.1
rfhs0004 IN      CNAME   dns
dns      IN      HINFO  "Sun SparcStation 4" "Solaris 2.5"

smaug    IN      A      194.95.104.11
rfhs1030 IN      CNAME   smaug
smaug    IN      HINFO  "Sun SparcStation 1+" "NetBSD 1.1"
```

```
grizu          IN A      132.199.208.131
grizu          IN HINFO  "Amiga 3000UX" "NetBSD/amiga V1.1"

rfhs0003      IN A      132.199.208.200
delphi        IN CNAME  rfhs0003
rfhs0003      IN HINFO  "Sun SparcStation 5" "Solaris 2.4"

;; Zusaetzliche Eintraege:
;; rz          IN MX     10 rfhs4004.fh.uni-regensburg.de.
rz            IN MX     10 rfhs0002.fh.uni-regensburg.de.
```