

imond unter 'normalem' Linux

HowTo-Version 1.3a (13. Juni 2004)

1. Einleitung

Worum geht es hier überhaupt ? Und was ist "imond" ?

Ich werde zunächst die zweite Frage beantworten. "imond" ist ein Server-Daemon aus dem fli4l-Linux-Disketten-Router (www.fli4l.de). Das Programm ist für die Kontrolle und Statusausgabe der Internet- Verbindung(en) des Routers zuständig. Also im Großen und Ganzen eine sehr nützliche Sache.

In diesem kleinen How-To soll anschaulich erklärt werden, wie man den Server-Daemon "imond" unter einem 'normalen' Linux-System zum laufen bringt und was man dabei beachten muß. Mein Versuchssystem läuft unter SuSE Linux 8.2 professional (www.suse.de).

Sinn und Zweck des Ganzen

Ich benutze meinen Linux-Rechner als Router für eine Internet- Verbindung. Damit ermö gliche es allen Computern in meinem Netzwerk (LAN) die Internet- Verbindung gemeinsam zu nutzen. Da ich keine Flatrate habe, habe ich Abstand davon genommen, die Internet- Verbindung bei Bedarf automatisch aufbauen zu lassen. Es muß also jedesmal der Wä hlvorgang per Hand ausgelö st werden. Das sah bis jetzt so aus, daß sich der Benutzer über Telnet auf dem Server / Router einlogt und ein Script aufruft, welches dann die Verbindung aufbaut bzw. abbaut. Nun sind aber nicht alle Mitglieder meiner Familie computerbegeistert und tun sich mit diesem Vorgehen doch etwas schwer. Hier kommt "imond" ins Spiel. Der Server-Daemon kann nämlich mit Client-Programmen für verschiedene Betriebssysteme (zur Zeit: Windows und Linux) über das Netzwerk gesteuert werden. Somit ist es nun mö glich, die Internet- Verbindung mit einem einzigen Mausclick zu starten und mit einem weiteren zu beenden. Zusä tzlich zeigt das Client- Programm auf jedem Rechner den Verbindungsstatus an.

Einschränkungen

Alle Angaben in diesem How-To beziehen sich nur auf SuSE Linux 8.2 professional (außer anders angegeben) und auch nur auf ISDN-Verbindungen. Der Server-Daemon "imond" ist auch in der Lage, Modem- und xDSL-Verbindungen zu kontrollieren. Des weiteren gehe ich davon aus, daß die Internet- Verbindung über eine ISDN-Karte bereits eingerichtet ist und funktioniert. Ich zeige in diesem How-To nur die Mö glichkeit auf, die ISDN-Schnittstelle direkt mit den Tools aus dem Paket 'isdn4linux' (unter SuSE 'i4l-base'), hier ganz besonders "isdnctrl", zu steuern. Dies kann sicherlich auch ohne großen Aufwand anders gestaltet werden. Mehr zu diesem Thema weiter unten an gegebener Stelle.

Ich beziehe mich in diesem How-To eigentlich auch nur auf zwei Funktionen des "imond". Das sind das Starten bzw. Beenden einer Internet- Verbindung und die Statusanzeige auf dem Client-Computer. Alle anderen Funktionen wie auf der fli4l-Homepage beschrieben bleiben zunächst unbeachtet.

Anforderungen an das System / Netzwerk, den Anwender

Wie unter *Einschränkungen* beschrieben, behandle ich in diesem How-To (vorerst) nur ISDN-Verbindungen. Ich gehe in diesem How-To nicht auf die Einrichtung der ISDN-Hardware oder der zum Betrieb notwendigen Software ein. Dies sollte zu Beginn schon funktionieren. Ich gehe auch davon aus, daß das lokale Netzwerk (LAN) funktioniert und es wenigstens, neben dem Server / Router, einen weiteren Computer im Netzwerk gibt, der als Client fungiert. Dieser Computer muß unter Windows oder Linux laufen. Ich gehe auch davon aus, daß die Kommunikation zwischen Client und Server über das TCP/IP-Protokoll bereits funktioniert.

Die Installation des Servers erfolgt komplett unter Linux und kann auch nicht über Windows vorgenommen werden. Alle hier gezeigten Befehle müssen als User ROOT direkt in der Console eingegeben werden. Grundlegendes Linux-Wissen aus folgenden Bereichen ist hilfreich: Umgang mit der Text-Console, Skriptsprache, Boot-Scripte und deren Funktion. Ich versuche aber all dieses anschaulich darzustellen.

2. benötigte Software

Es wird natürlich der Server-Daemon "imond" benötigt. Dazu kommt noch ein (oder auch mehrere) Client-Programme, wahlweise für Windows oder Linux. Alle benötigten Programme (Server & Clients) befinden sich samt Quellcode und Dokumentation in dem Basis- Paket der fli4l-Distribution. Dieses kann auf der fli4l-Homepage (www.fli4l.de) in der Rubrik 'Download' bezogen werden. Dieses How-To bezieht sich auf die Version 2.0.8 der fli4l- Distribution. Es sollte aber auch mit jeder neueren Version funktionieren. Zur Zeit der Entstehung dieses How-Tos war auch die Entwickler-Version 2.1.3 verfügbar. Mit dieser geht es nicht, da der Windows- Client "imonc.exe" einen Bug hat. Wer auf Windows verzichten kann, kann es auch mit dieser Version probieren. Diese Möglichkeit steht mir aber nicht zu, da in meinem Netzwerk hauptsächlich Windows-Clients laufen.

3. "imond" kompilieren

Bevor es überhaupt losgehen kann, muß der Server-Daemon neu kompiliert (aus dem Quellcode erstellt) werden. Dazu wird das Archiv zunächst in ein neues Verzeichnis entpackt:

```
~ > md unpack
```

Nun muß das Archiv in das neu angelegt Verzeichnis kopiert werden: z.B.

```
~ > cp fli4l-2.0.8.tar.gz unpack/
```

In das neue Verzeichnis wechseln und das Archiv entpacken:

```
~ > cd unpack  
~/unpack > tar -xzf fli4l-2.0.8.tar.gz
```

Aus dem Archiv wurde ein weiteres Verzeichnis ausgepackt. Wir wechseln gleich in das Verzeichnis, in dem sich der Quellcode für "imond" befindet und kompilieren das Programm:

```
~/unpack > cd fli4l-2.0.8/src/base  
~/unpack/fli4l-2.0.8/src/base > make all-glibc
```

Das Kompilieren kann, je nach Hardwareausstattung, einen Augenblick dauern. Dabei kann es vorkommen, daß verschiedene Fehlermeldungen ausgegeben werden. Wichtig ist für dieses Projekt nur, daß das Executable imond im gleichen Verzeichnis erzeugt wird.

Sollte das Make-Script nicht funktionieren, kann man den Kompiler auf direkt aufrufen. Auf meinem System ist standardmäßig gcc installiert.

```
~/unpack/fli4l-2.0.8/src/base > gcc -o imond -s imond.c -l crypt
```

Es sollten aber keine Probleme auftreten. Wenn doch, bitte ich um Mitteilung, um die Probleme und Lösungen in dieses How-To aufzunehmen.

4. Installation

Die Installation ist denkbar einfach. Etwas komplizierter wird es erst bei der Konfiguration des Servers und der Skripte. Also hier die Installation (diese Schritte unbedingt als ROOT ausführen !):

```
~/unpack/fli4l-2.0.8/src/base > cp imond /usr/local/bin
```

Eventuell existiert das Verzeichnis /usr/local/bin gar nicht. Dann muß es angelegt werden. Prinzipiell kann "imond" in jedes beliebige Verzeichnis installiert werden. Ich weiß nur nicht, was für Auswirkungen das auf die Funktion hat. Im fli4l ist "imond" jedenfalls in /usr/local/bin installiert.

Nun brauchen wir noch das Script, daß von "imond" aufgerufen wird, um die Verbindung zu steuern. Dieses Script muß auf jeden Fall an das bestehende System angepaßt werden !

```
~/unpack/fli4l-2.0.8/src/base > cd ../../opt/files/usr/local/bin  
~/unpack/fli4l-2.0.8/opt/files/usr/local/bin > cp fli4lctrl /usr/local/bin
```

In dem Verzeichnis ~/unpack/fli4l-2.0.8/opt/files/usr/local/bin befindet sich eine schon fertig kompilierte Version des "imond". Diese funktioniert aber auf einem 'normalen' Linux System nicht, da die vorkompilierte Version gegen andere Bibliotheken gelinkt ist. "imond" muß also zur Verwendung unter einem 'normalen' Linux System auf jeden Fall neu kompiliert werden.

5. Konfiguration

In diesem Abschnitt geht es um die Konfiguration des Server-Daemon "imond". Es geht einfach los und wird dann, leider, etwas schwieriger. Durch das Übertragen aus einer vollkommen anderen Umgebung kommt es manchmal zu Verwirrungen. Ich hoffe, ich kann hier weitestgehend Klarheit schaffen.

5.1 imond.env

In dieser Datei werden einige grundsätzliche Einstellungen für "imond" vorgenommen. Die Datei sollte nach Möglichkeit im Verzeichnis /etc abgelegt werden. Hier ist die Datei 'imond.env' von meinem System: (Diese Datei ist auch in dem Archiv '[imond4linux.tar.gz](#)' enthalten)

```

# environment configuration file for daemon 'imond'
# taken from the fli4l-distribution 2.1.3
#
# created by Ingwer Heidbrede, August 23, 2003
#
#-----
# environment for 'imond'
#-----
START_IMOND='yes' # start imond (yes/no)
IMOND_PORT='5000' # TCP-Port, see also
FIREWALL_DENY_PORT_x!
IMOND_PASS='' # imond-password, may be empty
IMOND_ADMIN_PASS='admin' # imond-admin-password, may be
empty
IMOND_LED='ttyS0' # tty for led: ttyS0, ttyS1, etc or empty
IMOND_BEEP='no' # beep if connection going up/down (yes/no)
IMOND_LOG='yes' # log /var/log/imond.log (yes/no)
IMOND_LOGDIR='/var/log' # log-directory, e.g. /var/log
IMOND_SYSLOG='yes' # also log all messages to syslog (yes/no);
# requires a running syslog daemon;
# useful on a 'standard' linux system

IMOND_ENABLE='no' # accept "enable/disable" commands
IMOND_DIAL='yes' # accept "dial/hangup" commands
IMOND_ROUTE='no' # accept "route" command
IMOND_REBOOT='no' # accept "reboot" command (yes/no)

#-----
# export variables to the environment
#
# ***** Do NOT edit below this line !!! *****
#
#-----

declare -x IMOND_PASS
declare -x IMOND_ADMIN_PASS
declare -x IMOND_ENABLE
declare -x IMOND_DIAL
declare -x IMOND_ROUTE
declare -x IMOND_REBOOT
declare -x DIALMODE

# eof imond.env

```

Der Aufbau der Datei ist recht simpel. Die Syntax folgt den Regeln eines Shell-Scriptes (Genau genommen ist diese Datei ein Shell-Script). Mit '#' wird ein Kommentar eingeleitet. Alles was rechts von diesem Zeichen steht wird ignoriert. Ich werde die Optionen jetzt kurz durchgehen und ihre Funktion erklären. Ein Teil dieser Angaben wird vom Boot-Script (kommt weiter unten noch) ausgewertet, der andere Teil wird als Umgebungsvariable exportiert.

Mit `START_IMOND` wird festgelegt, ob der "imond"-Daemon während des Systemstartes überhaupt gestartet werden soll. Die Möglichkeiten sind 'yes' und 'no'. Wird hier 'no' eingestellt, werden alle weiteren Einstellungen überflüssig. Es macht also nur Sinn, hier 'yes' einzutragen.

`IMOND_PORT` gibt an, auf welchem TCP/IP-Port "imond" auf Anfragen der Client-Programme lauscht. Der Standardport ist 5000. Es kann aber auch jeder beliebige andere Port gewählt werden. Wird kein Wert angegeben, wird automatisch der Standardport verwendet. Sollte gleichzeitig eine Firewall auf dem System laufen, ist zu beachten, daß der hier gewählte Port auch über das lokale Netzwerk zu erreichen ist. Ein Zugriff von Außen auf diesen Port sollte möglichst unterdrückt werden.

Wenn vor der Verbindung zum "imond"-Server ein Kennwort abgefragt werden soll, muß dieses bei `IMOND_PASS` eingetragen werden. Hier ist eine beliebige Eintragung möglich. Keine Eintragung bedeutet, die Client-Programme können ohne Kennwort Verbindung zum Server aufnehmen und Befehle senden.

Bei `IMOND_ADMIN_PASS` sollte auf jeden Fall ein Kennwort eingetragen werden. Warum hier zwei getrennte Kennwörter benutzt werden, soll später noch genauer dargelegt werden. An dieser Stelle nur soviel: Die Client-Programme können in zwei verschiedenen Modi betrieben werden: dem Admin- und dem User-Modus. Im Admin-Modus stehen mehr Befehle zur Verfügung, die unter Umständen Einstellungen am ISDN-System verändern können die wiederum zu Funktionsbeeinträchtigungen führen können. Der User-Modus reicht für den normalen Betrieb vollkommen aus.

Die Angabe `IMOND_LED` bezieht sich auf die serielle Schnittstelle, an der die LED-Status-Anzeige angeschlossen ist. Mehr zu diesem Thema weiter unten. Um diese Funktion nicht zu nutzen, wird einfach kein Wert eingetragen. Ansonsten die serielle Schnittstelle, z.B. `ttyS0` für `COM1` usw. in der Linux-üblichen Schreibweise.

Wird `IMOND_BEEP` auf 'yes' gesetzt, wird bei jedem Verbindungsaufbau / -abbau ein Zweiklangton über den PC-Lautsprecher ausgegeben. Will man seine Ruhe haben, bleibt die Einstellung bei 'no'.

Bei `IMOND_LOG` kann festgelegt werden, ob die Verbindungen protokolliert werden sollen. Hier sind die Angaben 'yes' und 'no' möglich.

Hat man bei `IMOND_LOG='yes'` gewählt, kann man bei `IMOND_LOGDIR` das Verzeichnis einstellen, in dem die Protokolldatei abgelegt wird. Sinnvollerweise ist das `/var/log`. Es kann aber auch jedes andere Verzeichnis gewählt werden. Dann ist allerdings darauf zu achten, das die Zugriffsrechte auch stimmen ! Der Dateiname der Protokolldatei ist immer 'imond.log'.

Gibt man bei `IMOND_SYSLOG 'yes'` an, werden alle Meldungen auch über den `syslog`-Daemon protokolliert. Diese Funktion ist auf einem 'normalen' Linux System sinnvoll. Unter SuSE Linux erscheinen die Meldungen dann zusätzlich auf der 10. Textconsole (`ALT + F10`). Hier sind die Angaben 'yes' und 'no' möglich.

Mit der Einstellung `IMOND_ENABLE` kann man angeben, ob im User-Modus der Dialmode verändert werden darf. Mögliche Arten des Dialmode sind `manual`, `auto` und `off`. Wenn diese Einstellung auf 'no' gesetzt ist, kann im User-Modus der Dialmode nicht verändert werden. Ich halte dies für eine sinnvolle Einstellung auf einem 'normalen' Linux System. Mit der Angabe von 'yes' kann man auch im User-Modus den Dialmode auf dem Server ändern. (Hinweis: Mit dem weiter unten vorgestellten, umgeschriebenen Script 'fli4lctrl' ist es nicht

mö glich, den Dialmode zu ä ndern.)

Die Einstellung `IMOND_DIAL` bestimmt, ob im User-Modus der Wä hlvorgang ausgelö st werden darf. Sinnvollerweise setzt man diese Einstellung aus 'yes'. Mö chte man verhindern das im User-Modus eine Verbindung aufgebaut werden kann, setzt man hier 'no' ein. Dann kann "imond" im User-Modus nur als Statusanzeige benutzt werden.

Die Option `IMOND_ROUTE` bezieht sich darauf, ob im User-Modus der Provider ausgewä hlt werden darf oder ob nur ein vorgegebener Provider benutzt werden darf. Diese Option hat nichts mit der Funktion des Routens im Linux-Sinne zu tun. Will man es im User-Modus erlauben, trä gt man 'yes' ein, ansonsten 'no'.

Die Option `IMOND_REBOOT` bestimmt, ob man den Server über das Client-Programm im User-Modus neu starten oder herunterfahren darf. Bei einem Netzwerk-Server mit einem 'normalen' Linux System ist diese Funktion vielleicht nicht so sinnvoll. Deshalb empfehle ich `IMOND_REBOOT='no'`. Will man es dennoch erlauben, setzt man die Option einfach auf 'yes'.

Zu guter letzt werden einige wichtige Variablen in die Shell-Instanz in der "imond" lä uft exportiert, damit sie stä ndig zur Verfügung stehen. Mit diesen hier vorgestellten Einstellungen funktioniert es bei mir wunderbar.

5.2 imond.conf

Die Datei 'imond.conf' muß im Verzeichnis /etc abgelegt werden. Sie ist für die Funktion von "imond" zwingend erforderlich. Die in dieser Datei abgelegten Informationen geben "imond" die Mö glichkeit, die Telefonnummer des Providers in einen Namen umzusetzen und die Verbindungsentgelte zu berechnen. Hier ist die Datei von meinem System: (Diese Datei ist auch in dem Archiv '[imond4linux.tar.gz](#)' enthalten.)

```
# Least-Cost-Routing -- configuration file for imond
#
# for complete documentation see docs of fli4l !
#
# created by Ingwer Heidbrede, August 23, 2003
#
# hup: hangup-Timeout in seconds
# lcroute:Least cost routing circuit (yes/no)
#
# day hour device bandwith hup lcroute default phone name charge
charge
# route intervall
Mo-Su 00-24 ippp0 -1:-1 180 yes yes 0019237120 FreeNetPower 0.0099
60
# eof
```

Hinweis: Im Umfeld von *fli4l* kö nnen in dieser Datei auch mehrere Datensä tze für Provider

angelegt werden. "imond" kann dann tageszeitabhängig zwischen den günstigsten Providern umschalten. Diese Funktion habe ich bis jetzt nicht getestet.

Diese Datei bedarf wohl einiger Erklärungen. Ich werde im folgenden jede Spalte kurz erläutern und wenn nötig kleine Beispiele geben.

In der ersten Spalte wird der Tagesbereich angegeben, in dem dieser Provider gültig ist. Es werden jeweils die ersten beiden Buchstaben der englischen Wochentage benutzt. Für Montag bis Sonntag ist das dementsprechend 'Mo-Su'.

In der zweiten Spalte wird der Stundenbereich angegeben. Es müssen immer zwei Ziffern angegeben werden. Zum Beispiel 00-24 für einen ganzen Tag oder 08-18 für die Zeit von acht Uhr morgens bis sechs Uhr abends.

Die dritte Spalte enthält den Namen des zu verwendenden Gerätes, in der Regel ipp0 für das erste ISDN-Gerät. Es kann auch Kanalbündelung benutzt werden. In diesem Fall müssen beide Geräte angegeben werden; z.B. 'ipp0/ipp1' für ipp0 und ipp1.

In der vierten Spalte werden die Schwellenwerte für die automatische Kanalbündelung notiert. Die Schwellenwerte werden in folgender Schreibweise notiert: '6144:30'. Das bedeutet, daß der zweite ISDN-Kanal zugeschaltet wird, wenn die Transferrate 6kB/s für mindestens 30 Sekunden überschreitet. Der erste Wert ist die Transferrate in Kilobit pro Sekunde (kbs), im Beispiel 6114 kbs (das entspricht rund 6 kB/s (6114/1024)). Der zweite Wert ist die Dauer in Sekunden, für die der erste Wert überschritten werden muß, damit sich der zweite ISDN-Kanal zuschaltet. Das Wertepaar '-1:-1' bedeutet, daß die Kanalbündelung nicht benutzt wird.

In der fünften Spalte steht die Hang up time in Sekunden. Das ist die Zeit, die gewartet wird bevor die Verbindung automatisch beendet wird, wenn keine Daten mehr fließen.

In der sechsten Spalte wird angegeben, ob es sich bei diesem Provider um einen *Least cost routing circuit* handelt oder nicht. Da dieses hier erst mal nicht wichtig ist, erspare ich mir nähere Informationen. Wer genaueres darüber wissen möchte, der muß in die Dokumentation von *fli4l* schauen. Hier ist nur wichtig, daß bei einem Provider 'yes' steht, da die Client-Programme ansonsten nicht korrekt funktionieren.

Die siebte Spalte teilt dem Programm mit, welches der Standard-Provider ist, der bevorzugt benutzt werden soll. Da hier nur einer eingetragen ist, bekommt dieser natürlich den Eintrag 'yes'.

Die Einwahlnummer des Provider wird in Spalte 8 angegeben. Wer, so wie ich, an einer Telefonanlage hängt und keine automatische Amtsleitungsbelegung benutzt, muß in der Regel der eigentlichen Telefonnummer noch eine '0' voranstellen.

In der neunten Spalte wird ein Name für den Provider festgelegt. Dieser steht dann anstatt der Rufnummer in den Log-Dateien.

Die letzte und die vorletzte Spalte haben mit der Kostenberechnung zu tun. "imond" ist in der Lage, die Kosten für die Online-Verbindungen zu berechnen. Damit dieses richtig funktioniert, müssen in der zehnten Spalte die Kosten für eine Gebühreneinheit eingetragen werden. In der elften Spalte muß eingetragen werden, wie lange eine Gebühreneinheit dauert (in Sekunden).

5.3 Boot-Script

Das Boot-Script dient dazu, den "imond"-Daemon bei jedem Neustart auszuführen. Ich stelle an dieser Stelle mein Boot-Script vor, und ich werde ein ganz einfaches Script zeigen, daß auf jeder Distribution funktionieren sollte. Hier ersteinmal mein Script, daß ich einfach aus einem SuSE-Basisscript erstellt habe: (Auch diese Datei ist in dem Archiv ['imond4linux.tar.gz'](#) enthalten.)

```

#!/bin/sh
#
# Author: Ingwer Heidbrede, August 1, 2003
#
# /etc/init.d/imond
#
# system startup script for 'imond'

### BEGIN INIT INFO
# Provides:          imond
# Required-Start:    $ALL
# X-UnitedLinux-Should-Start:
# Required-Stop:
# X-UnitedLinux-Should-Stop:
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Short-Description: imond - controlls mainly the internet status LED
# Description:        The imond daemon has been copied from the fli4l floppy
#                     linux distribution. It is able to controll the
#                     internet connection. It can be controlled via
#                     network from various OSes.
### END INIT INFO

# Shell functions sourced from /etc/rc.status:
#   rc_check          check and set local and overall rc status
#   rc_status         check and set local and overall rc status
#   rc_status -v      be verbose in local rc status and clear it afterwards
#   rc_status -v -r  ditto and clear both the local and overall rc status
#   rc_status -s      display "skipped" and exit with status 3
#   rc_status -u      display "unused" and exit with status 3
#   rc_failed         set local and overall rc status to failed
#   rc_failed set    set local and overall rc status to
#   rc_reset          clear both the local and overall rc status
#   rc_exit           exit appropriate to overall rc status
#   rc_active         checks whether a service is activated by symlinks
#   rc_splash arg     sets the boot splash screen to arg (if active)
. /etc/rc.status

# Reset status of this service
rc_reset

# Return values acc. to LSB for all commands but status:
# 0      - success
# 1      - generic or unspecified error
# 2      - invalid or excess argument(s)
# 3      - unimplemented feature (e.g. "reload")
# 4      - user had insufficient privileges
# 5      - program is not installed
# 6      - program is not configured
# 7      - program is not running
# 8--199 - reserved (8--99 LSB, 100--149 distrib, 150--199 appl)
#
# Note that starting an already running service, stopping
# or restarting a not-running service as well as the restart
# with force-reload (in case signaling is not supported) are
# considered a success.

# Check for missing binaries (stale symlinks should not happen)
IMOND_BIN=/usr/local/bin/imond
test -x $IMOND_BIN || exit 5

# Check for existence of needed config files
IMOND_CONFIG=/etc/imond.conf
test -r $IMOND_CONFIG || exit 6
IMOND_ENV=/etc/imond.env
test -r $IMOND_ENV || exit 6

# load environment variables for imond
. $IMOND_ENV

case "$1" in
    start)

```

```

echo -n "Starting imond ..."

# analyse START_IMOND
if [ "$START_IMOND" = "no" ]; then
    exit 7
fi

# check if var IMOND_PORT is set
if [ -z "IMOND_PORT" ]; then
    IMOND_PORT='5000' # if var does not exist, set default value
fi

# building commandline
unset IMOND_COMMANDLINE

# use syslog ?
if [ "$IMOND_SYSLOG" = "yes" ]; then
    IMOND_COMMANDLINE=$IMOND_COMMANDLINE' -log-to-syslog'
fi

# beep on building/terminating connection ?
if [ "IMOND_BEEP" = "yes" ]; then
    IMOND_COMMANDLINE=$IMOND_COMMANDLINE' -beep'
fi

# check if the led function is used
if [ "$IMOND_LED" ]; then
    IMOND_COMMANDLINE=$IMOND_COMMANDLINE' -led /dev/'$IMOND_LED
fi

# analyse IMOND_LOG and IMOND_LOGDIR
if [ "$IMOND_LOG" = "yes" -a "$IMOND_LOGDIR" ]; then
    IMOND_COMMANDLINE=$IMOND_COMMANDLINE' '$IMOND_LOGDIR
fi

# running imond
echo command line: $IMOND_COMMANDLINE
$IMOND_BIN -port $IMOND_PORT $IMOND_COMMANDLINE > /dev/null

# Remember status and be verbose
rc_status -v
;;

stop)
echo -n "Shutting down imond ..."

# analyse START_IMOND
if [ "$START_IMOND" = "no" ]; then
    exit 7
fi

# get the PID of the running imond process
IMOND_PID=$( ps ax | grep -m 1 "imond" | awk '{ print $1 }' )
kill $IMOND_PID

# Remember status and be verbose
rc_status -v
;;

restart)
## Stop the service and regardless of whether it was
## running or not, start it again.
$0 stop
$0 start

# Remember status and be quiet
rc_status
;;

status)
echo -n "Checking for imond ..."
# get the PID of the running imond process
IMOND_PID=$( ps ax | grep -m 1 "imond -log" | awk '{ print $1 }' )

```

```

echo
if [ "$IMOND_PID" ]; then
    echo "imond is running with PID: $IMOND_PID"
elif [ -z "$IMOND_PID" ]; then
    echo "imond is not running. No PID found."
else
    echo "ERROR !"
fi
;;

*)
    echo "Usage: $0 {start|stop|restart|status}"
    exit 1
;;

esac
rc_exit

```

Dieses Script ist doch schon relativ komplex und wird vermutlich nur auf einem SuSE Linux unverändert funktionieren. Ein paar kurze Erklärungen: Der Block 'INIT INFO' enthält Informationen für YaST2 mit Hilfe dessen das Script in den Boot-Prozess eingebunden wird. Die beiden folgenden Blöcke binden Funktionen aus dem SuSE System ein. Diese dienen ausschließlich dazu, den Rückgabewert des Scriptes zu definieren. Danach wird geschaut, ob alle benötigten Dateien auch an den dafür vorgesehenen Orten vorhanden sind. Ist dies nicht der Fall, wird mit einem Fehler abgebrochen. Die Datei 'imond.env' wird eingelesen (Die Definitionen der Variablen werden übernommen.). Anschließend verzweigt das Script in verschiedene Richtungen, abhängig davon, mit welchem Parameter das Script aufgerufen wurde. Interessant ist hier eigentlich nur der 'start'-Zweig. Alle anderen sind doch recht überschaubar.

"imond" wird eigentlich über Kommandozeilen-Parameter gesteuert und liest selbst nur die Datei *imond.conf* mit den Providerinformationen ein. Im 'start'-Zweig wird aus den Angaben aus der Datei *imond.env* durch einfach Bedingungsstest eine Kommandozeile zusammengesetzt. Dieser Umweg über die Konfigurationsdatei *imond.env* hat den Vorteil, daß man am Boot-Script keine Änderungen vornehmen muß um die Funktion von "imond" anzupassen.

Das Boot-Script wird (unter SuSE-Linux) im Verzeichnis */etc/init.d/* abgelegt. Es muß mit dem *chmod*-Befehl ausführbar gemacht werden (*chmod u+x imond*). Unter SuSE-Linux kann nun über den Runlevel-Editor eingestellt werden, in welchem Runlevel der "imond"-Daemon gestartet werden soll. Sinnvoll sind hier nur Runlevel, in denen Netzwerkfunktionen gestartet werden. Unter SuSE *_linux v8.2* sind dies Runlevel 3 und 5.

An dieser Stelle möchte ich ein wesentlich einfacheres Boot-Script vorstellen, daß vermutlich auf jedem Linux-System läuft. Auch dieses Boot-Script greift wieder auf die Konfigurationsdatei *imond.env* zu. Im Anschluß stelle ich die einzelnen Kommandozeilenparameter von "imond" vor. Dieses einfache Boot-Script bietet nur die Funktionen 'start' und 'stop'. Alle anderen sind für die grundlegende Funktion nicht wichtig. Deshalb lasse ich sie an dieser Stelle weg. Ich gehe auch davon aus, daß alle Programme und Dateien an den angegebenen Stellen vorhanden sind.

```

#!/bin/sh
#
# Author: Ingwer Heidbrede, August 1, 2003
#
# /etc/init.d/imond
#
# system startup script for 'imond'

# load environment variables for imond
. /etc/imond.conf

case "$1" in
start)
    echo -n "Starting imond ..."

    # analyse START_IMOND
    if [ "$START_IMOND" = "no" ]; then
        exit 7
    fi

    # check if var IMOND_PORT is set
    if [ -z "IMOND_PORT" ]; then
        IMOND_PORT='5000' # if var does not exist, set default value
    fi

    # building commandline
    unset IMOND_COMMANDLINE

    # use syslog ?
    if [ "$IMOND_SYSLOG" = "yes" ]; then
        IMOND_COMMANDLINE=$IMOND_COMMANDLINE' -log-to-syslog'
    fi

    # beep on building/terminating connection ?
    if [ "IMOND_BEEP" = "yes" ]; then
        IMOND_COMMANDLINE=$IMOND_COMMANDLINE' -beep'
    fi

    # check if the led function is used
    if [ "$IMOND_LED" ]; then
        IMOND_COMMANDLINE=$IMOND_COMMANDLINE' -led /dev/'$IMOND_LED
    fi

    # analyse IMOND_LOG and IMOND_LOGDIR
    if [ "$IMOND_LOG" = "yes" -a "$IMOND_LOGDIR" ]; then
        IMOND_COMMANDLINE=$IMOND_COMMANDLINE' '$IMOND_LOGDIR
    fi

    # running imond
    echo command line: $IMOND_COMMANDLINE
    /usr/local/bin/imond -port $IMOND_PORT $IMOND_COMMANDLINE > /dev/null

    ;;

stop)
    echo -n "Shutting down imond ..."

    # analyse START_IMOND
    if [ "$START_IMOND" = "no" ]; then
        exit 7
    fi

    # get the PID of the running imond process
    IMOND_PID=$( ps ax | grep -m 1 "imond" | awk '{ print $1 }' )
    kill $IMOND_PID

    ;;

esac

```

Dieses Script sollte auch für einen normalen Linux-Benutzer kein Problem darstellen. Im 'start'-Abschnitt wird aus den Variablen der Konfigurationsdatei *imond.env* die Kommandozeile zum Aufruf von "imond" zusammengestellt. Im 'stop'-Abschnitt wird die PID des "imond"-Prozesses ermittelt und der Prozess anschließend beendet. Jeder Abschnitt beginnt mit einer Überprüfung, ob "imond" überhaupt gestartet wird / wurde.

Dieses Boot-Script sollte auch an der zentralen Stelle für Boot-Scripte der jeweiligen Distribution abgelegt werden. Dann müssen noch symbolische Links in den Runlevel-Verzeichnissen auf das "imond"-Boot-Script verweisen. Das Script muß mit dem `chmod`-Befehl ausführbar gemacht werden(`chmod u+x imond`).

Hier sind nun die einzelnen Kommandozeilenparameter:

allgemeine Syntax:

```
imond [-port portno] [-log-to-syslog] [-beep] [-led /dev/ttySx] <imond-log-dir>  
<telmond-log-dir>
```

`[-port portno]` "imond" lauscht standardmäßig auf TCP-Port 5000. Mit dieser Option kann ein abweichender Port eingestellt werden. Dieser muß dann auch im Client-Programm extra angegeben werden.

`[-log-to-syslog]` Dieses Flag gibt an, ob Meldungen über syslog ausgegeben werden sollen. Auf einem 'normalen' Linux-System ist dies auf jeden Fall sinnvoll.

`[-beep]` Dieses Flag bewirkt, wenn es angegeben wird, daß "imond" bei jedem Verbindungsaufbau / -abbau einen Zweiklangton ausgibt.

`[-led /dev/ttySx]` Eine Leuchtdiode kann an die serielle Schnittstelle angeschlossen werden, um den Verbindungsstatus anzuzeigen. Hier wird in diesem Falle die benutzte Schnittstelle angegeben. Es wird die Linux-Notation für Schnittstellen verwendet, zum Beispiel `/dev/ttyS0` für die erste serielle Schnittstelle.

`<imond-log-dir>` An dieser Stelle wird das Verzeichnis angegeben, in dem "imond" seine Protokolldatei ablegt. In der Protokolldatei werden alle ausgehenden Verbindungen aufgezeichnet. Die Datei hat immer den Namen 'imond.log'.

5.4 fli4lctrl

Bei der Datei *fli4lctrl* handelt es sich wieder um eine ausführbare Script-Datei. Sie wird von "imond" aufgerufen um die Verbindungskontrolle auszuführen. Die Datei *fli4lctrl* enthält die Linux-Befehle zum Steuern des Verbindungsauf- und -abbaus. Ohne sie funktioniert in diesem Punkte gar nichts.

Hier zeige ich erst einmal meine Script-Datei. Da ich es nicht für sinnvoll erachte, das Rad sprichwörtlich neu zu erfinden, habe ich diese Datei aus dem fli4l-Linux übernommen und an meine Bedingungen angepaßt. (Auch diese Datei ist in dem Archiv '[imond4linux.tar.gz](#)' enthalten.)

```

#!/bin/sh
#-----
#   fli4lctrl - extend isdnctrl program for dsl connections

#*****
# rewritten for use on a 'full' linux system (SuSE Linux 8.2) ( ISDN only )
# the ISDN commands are in a script called 'net'
# by Ingwer Heidbrede, August 23, 2003
#*****

cmd=$1

pgm_name=`basename $0 `

case "$cmd" in
    hangup)
        isdnctrl hangup ippp0
        ;;

    dial)
        isdnctrl dial ippp0
        ;;

    *)
        echo "$pgm_name: invalid arguments" >&2
        exit 2
        ;;
esac
exit 0
# eof

```

Ich habe dieses Script auf das absolut notwendige reduziert. Unter fli4l werden hier wesentlich mehr Funktionen gesteuert die jetzt nicht zur Verfügung stehen. Vielleicht werde ich weitere Funktionen in späteren Versionen dieses HowTos betrachten. Wer wissen möchte, welche Funktionen noch möglich wären, der muß in die Datei *fli4lctrl* aus der fli4l-Distribution schauen.

5.5 IP-Anzeige

Wie all jene schon gemerkt haben, die schon einmal einen Blick auf die Clients geworfen haben, kann die Client-Software auch die aktuelle IP-Adresse anzeigen, die durch den Provider zugeordnet wurde und die weltweit gültig ist - also die externe IP-Adresse. Damit dieses auch wirklich funktioniert, müssen die Schritte in diesem Abschnitt abgearbeitet werden.

Wem dies zu viel wird oder meint er braucht es nicht, kann diesen Abschnitt problemlos überspringen. Die Anzeige der IP-Adresse im Client hat auf die Funktion des selben keinen Einfluß.

Wie funktioniert das Ganze ?

Das Prinzip ist ganz einfach: Beim Aufbauen der InterNet-Verbindung wird die IP-Adresse in einer temporären Datei abgespeichert. Der Server-Daemon "imond" liest dann diese Datei aus und übermittelt die IP-Adresse an den Client.

Jetzt stellt sich natürlich die Frage, wie ich es erreiche, daß die IP-Adresse beim Starten der Verbindung in diese Datei geschrieben wird. Bei diesem Problem hilft uns Linux schon von Hause aus, genauer der Daemon 'ippd'. Der Daemon 'ippd' ist für die gesamte Steuerung der ISDN-Geräte zuständig. Dieser Daemon führt automatisch nach dem erfolgreichen Verbindungsaufbau ein Script aus, das sogenannte ip-up - Script. Unter SuSE Linux sieht es jetzt so aus, daß es vom System benötigte ip-up - Scripte gibt, um zum Beispiel den DNS-Server dynamisch anzupassen. Daneben gibt es ein Verzeichnis (/etc/ppp/ip-up.d/) in dem benutzereigene Ergänzungen dazu abgelegt werden können. Die werden dann automatisch bei jedem Verbindungsaufbau ausgeführt. Hier ist das Script, das ich für diesen Zweck entwickelt habe und welches im Verzeichnis /etc/ppp/ip-up.d abgelegt wird. (Es ist natürlich auch in dem Archiv '[imond4linux.tar.gz](#)' enthalten.)

```
#!/bin/bash
#
# IP-UP-Script: /etc/ppp/ip-up.d/imond-ip.ip-up
#
# Script zum Speichern der aktuellen IP-Adresse der InterNet- Verbindung
# für die weitere Verwendung mit IMOND ( aus fli4l )
#
# Autor: Ingwer Heidbrede, 22. Februar 2004
#

BASENAME=${0##*/}
INTERFACE=$1
DEVICE=$2
SPEED=$3
LOCALIP=$4
REMOTEIP=$5
IPPARAM=$6

# überprüfen, ob die Datei schon existiert
#
if [ -e /var/run/$INTERFACE.ip ]; then
    rm /var/run/$INTERFACE.ip
fi

# die IP-Adresse der aktuellen Verbindung in die Datei schreiben
#
echo $LOCALIP > /var/run/$INTERFACE.ip

# eof
```

Dem Script werden beim Aufruf diverse Parameter übergeben, denen im ersten Teil des Scripts wiedererkennbare Namen zugeordnet werden.

Der Rest ist eigentlich selbsterklärend. Die IP-Adresse wird in der Datei /var/run/interfacename.ip abgespeichert, zum Beispiel /var/run/ipp0.ip für das erste ISDN-Gerät.

Nun bleibt noch die Frage, was beim Beenden der InterNet-Verbindung passiert. Auch dafür hat der Daemon 'ippd' eine passende Lösung parat. Analog zu dem Script ip-up beziehungsweise dem Verzeichnis ip-up.d gibt es ein Script ip-down beziehungsweise ein Verzeichnis ip-down.d (unter /etc/ppp/), welches nach dem Beenden der InterNet-Verbindung ausgeführt wird. Hier ist das entsprechende Script, welches im Verzeichnis /etc/ppp/ip-down.d abgelegt werden sollte. (Ist selbstverständlich auch im Archiv '[imond4linux.tar.gz](#)' enthalten.)


```

#!/bin/bash
#
# IP-DOWN-Script: /etc/ppp/ip-down.d/imond-ip.ip-down
#
# Script zum Speichern der aktuellen IP-Adresse der InterNet- Verbindung
# für die weitere Verwendung mit IMOND ( aus fli4l )
#
# Autor: Ingwer Heidbrede, 22. Februar 2004
#

BASENAME=${0##*/}
INTERFACE=$1
DEVICE=$2
SPEED=$3
LOCALIP=$4
REMOTEIP=$5
IPPARAM=$6

# überprüfen, ob die Datei existiert
#
if [ -e /var/run/$INTERFACE.ip ]; then
    rm /var/run/$INTERFACE.ip
fi

# eof

```

Dieses Script hat eigentlich nur eine Aufgabe: Es löscht die Datei mit der IP-Adresse. Wer aufmerksam gelesen hat, hat sicherlich festgestellt, daß das ip-up - Script bevor es die Datei schreibt, überprüft, ob eine Datei bereits existiert und diese gegebenenfalls löscht. Dieses zweite Script ist eigentlich nur eine kleine Verschönerung des Ganzen. Funktionieren würde es ganz sicher auch nur mit dem ip-up - Script.

Für beide Scripte gilt, daß sie ausführbar sein müssen. Das heißt es müssen die Dateirechte entsprechend angepaßt werden (`chmod u+x imond-ip.ip-up` beziehungsweise `chmod u+x imond-ip.ip-down`).

6 Konfiguration des Linux-Client-Programmes

Im Basispaket der fli4l-Distribution sind gleich zwei Client-Programme für Linux enthalten: ein textbasiertes Konsolen-Programm und ein Client mit graphischer Oberfläche (GTK basiert). Der Quellcode beider Programme befindet sich im Verzeichnis `unix/`.

6.1 imonc

Zunächst möchte ich kurz den textbasierten Client 'imonc' vorstellen. Im ersten Schritt muß der Quellcode übersetzt werden:

```
~/unpack > cd fli4l-2.0.8/unix  
~/unpack/fli4l-2.0.8/unix > make imonc
```

Nun wird das neu erstellte Programm an einen allgemeinen Ort für ausführbare Programme kopiert, wie zum Beispiel /usr/local/bin.

```
~/unpack/fli4l-2.0.8/unix > cp imonc /usr/local/bin
```

Es ist sinnvoll, dieses Programm für jeden Benutzer des Systems ausführbar zu machen. Ansonsten kann nur 'root' das Programm benutzen und die Internet-Verbindung kontrollieren.

```
~/unpack/fli4l-2.0.8/unix > cd /usr/local/bin  
/usr/local/bin > chmod a+rx imonc
```

Nun kann das Programm von allen Benutzern des Rechners aufgerufen werden. Am einfachsten geht dies, wenn der Pfad zu imonc in den Umgebungsvariablen gespeichert ist. /usr/local/bin gehört in der Regel schon dazu. Nun kann 'imonc' von jedem Verzeichnis aus ausgeführt werden. Als Parameter muß 'imonc' der Rechnername / die IP-Adresse des Rechners übergeben werden, auf dem "imond" ausgeführt wird. Für den Fall, daß "imond" und 'imonc' auf dem selben Rechner ausgeführt werden, sieht der Programmaufruf dann so aus:

```
~ > imonc localhost
```

Als zweiten Parameter kann auch noch die Port-Nummer angegeben werden. Dies ist aber nur notwendig, wenn die Standardeinstellung geändert wurde. 'imonc' ist eigentlich selbsterklärend. Die wichtigsten Funktionen sind wohl 'dial' (3), 'hangup' (4) und 'quit' (0). Wenn alle Einstellungen aus diesem HowTo unverändert übernommen wurden, stehen nicht alle Funktionen zur Verfügung.

6.2 gtk-imonc

Der graphische Client 'gtk-imonc' ist wesentlich komfortabler zu bedienen und stellt auch mehr Funktionen zur Verfügung. Die elementaren Funktionen sind aber die Selben. Das Programm 'gtk-imonc' liegt als gepackter Quelltext vor. Als Erstes wird das Archiv in das /unpack- Verzeichnis kopiert und entpackt:

```
~/unpack/fli4l-2.0.8/unix > cp gtk-imonc.tar.gz ~/unpack  
~/unpack/fli4l-2.0.8/unix > cd ../..  
~/unpack/ > tar -xzf gtk-imonc.tar.gz
```

Beim Entpacken wurde der Inhalt des Archives *gtk-imonc.tar.gz* in einem neuen Verzeichnis mit dem Namen *gtk-imonc-0.4.1/* abgelegt. Nun wird aus dem Quelltext ein ausführbares Programm erstellt:

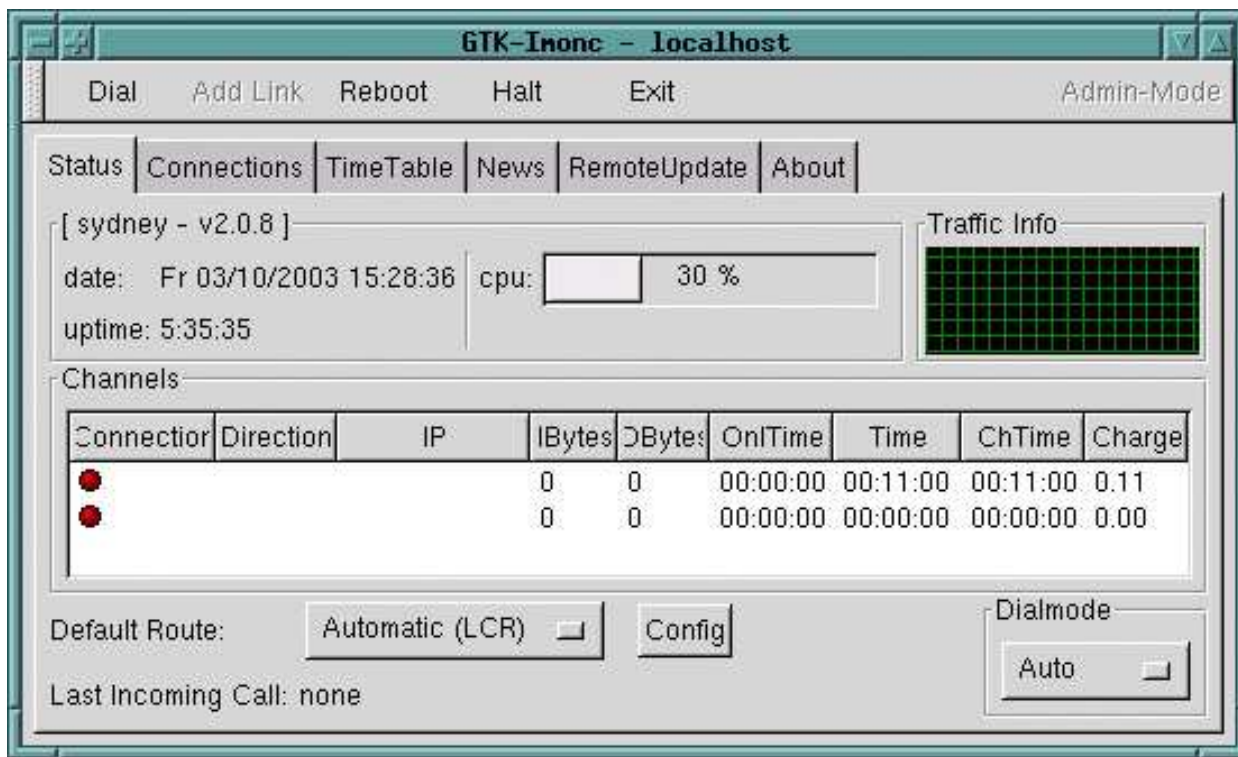
```
~/unpack/ > cd gtk-imonc-0.4.1  
~/unpack/gtk-imonc-0.4.1 > ./configure  
~/unpack/gtk-imonc-0.4.1 > make
```

Nun haben wir ein ausführbares Programm. Weitere Informationen zum Kompilieren von 'gtk-imonc' befinden sich in der Datei *LIESMICH* und *INSTALL* im Verzeichnis *~/unpack/gtk-imonc-0.4.1/*. Nun wird das Programm noch jedem Benutzer des Systems zugänglich gemacht:

```
~/unpack/gtk-imonc-0.4.1 > make install
```

Das Programm wird ebenfalls nach */usr/local/bin* installiert. Nach dem Programmaufruf erscheint das unten abgebildete Fenster. Beim Programmaufruf muß wieder der Rechnername / die IP-Adresse des Rechners auf dem "imond" ausgeführt wird übergeben werden.

```
~ > gtk-imonc localhost
```



Hier schnell ein paar Erklärungen zu den einzelnen Punkten. Zunächst einmal die Menüleiste. Mit dem Punkt 'Dial' wird eine (die Standard-) Verbindung zum Internet aufgebaut. Dieser Punkt ändert sich dann in 'Hangup' und beendet die Verbindung wieder. Über den Punkt 'Exit' wird gtk-imonc beendet. Die Punkte 'Halt' und 'Reboot' stehen nur im Admin-Modus zur Verfügung und sollten auf einem normalen Linux-System nicht benutzt werden.

Nun zu den einzelnen Registerkarten. Die erste Registerkarte trägt die Bezeichnung 'Status'. Hier werden ein paar statistische Systeminformationen angezeigt. Entscheidend ist hier der Bereich 'Channels' da dort Angaben zur Internetverbindung stehen, wenn eine existiert. Im Feld 'Traffic Info' wird die Auslastung der Leitung graphisch dargestellt. Die im Bild sichtbaren Buttons sind nur im Admin-Modus verfügbar.

Hinter der Registerkarte 'Connections' verbirgt sich eine Aufzeichnung aller ausgehenden Verbindungen mit Datums-, Zeit- und Zielangaben. Damit ist eine ausführliche Kontrolle der Verbindungen möglich. Zudem werden die Kosten pro Verbindung und die Gesamtkosten angezeigt.

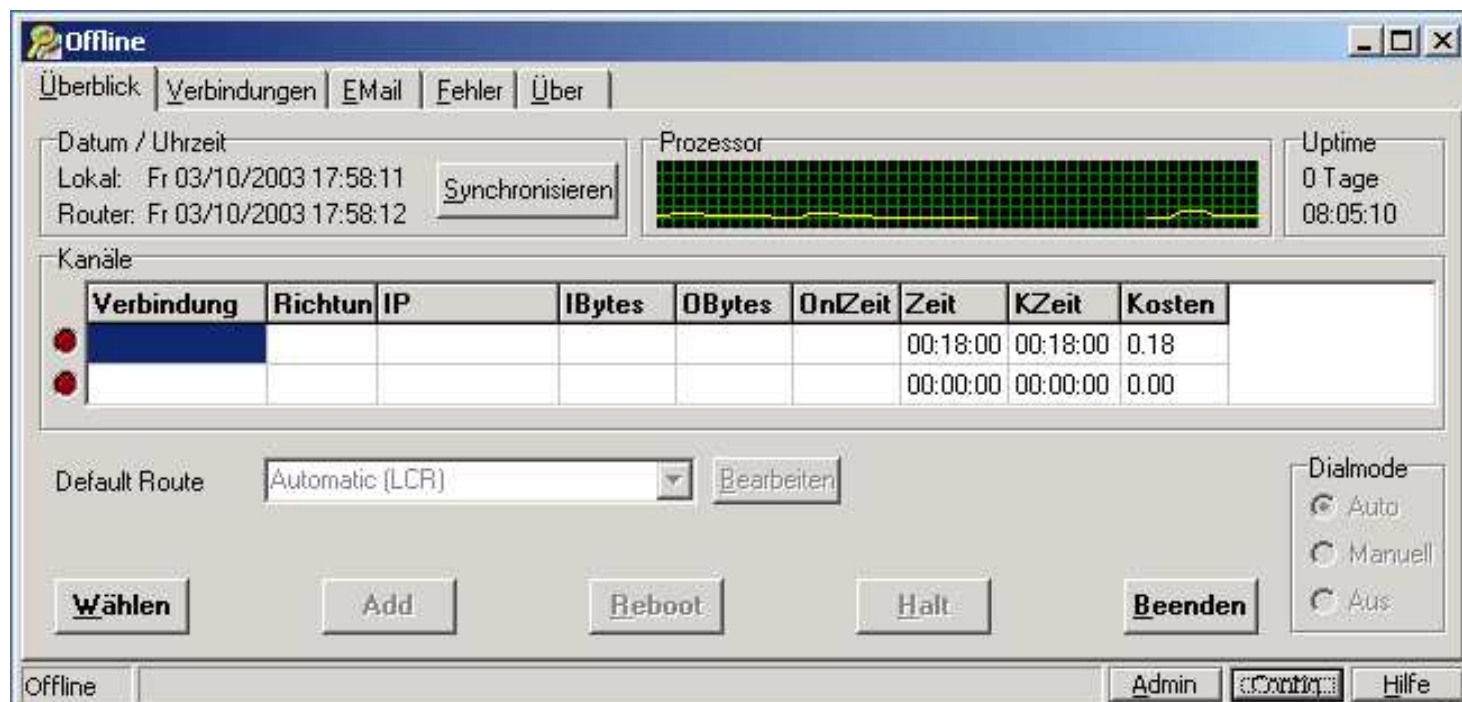
Die Registerkarte 'TimeTable' zeigt an, für welchen Tageszeitbereich welcher Provider zuständig ist. Alle anderen Registerkarten enthalten nur passive Funktionen. Der Abschnitt 'RemoteUpdate' ist nur im Admin-Modus verfügbar.

Weitere Informationen und Programm-Updates gibt es auf der [Homepage von GTK-imonc](#).

7 Konfiguration des Windows-Client-Programmes

Das Windows-Client-Programm 'imonc' ist im Basispaket der fli4l-Distribution bereits enthalten. Es befindet sich im Verzeichnis windows/imonc. Es ist der wohl umfangreichste

Client den es zur Zeit gibt. Alle Dateien aus dem Verzeichnis windows/imonc sollten jeweils in ein Verzeichnis auf der Festplatte des Clientrechners kopiert werden.



Nach dem Programmstart sollte ein solches Fenster zu sehen sein. Die Dokumentation zu 'imonc' für Windows im Rahmen der fli4l-Dokumentation ist sehr gut, weshalb ich an dieser Stelle darauf verzichten möchte. Die fli4l-Dokumentation ist im Basispaket enthalten und enthält alle wichtigen Informationen zu 'imonc'.

Nur einen kleinen Hinweis möchte ich an dieser Stelle doch noch geben: Der 'imonc' versucht beim Programmstart automatisch eine Verbindung zu einem Rechner mit dem DNS-Namen "fli4l" aufzubauen. Da ein richtiger Server aber wahrscheinlich einen anderen Namen hat, muß beim Aufruf von 'imonc' der DNS-Name oder die IP-Adresse des Server-PCs mit dem "/s:" - Parameter übergeben werden. Beispiel für einen Server-PC mit der IP-Adresse '192.168.0.1':

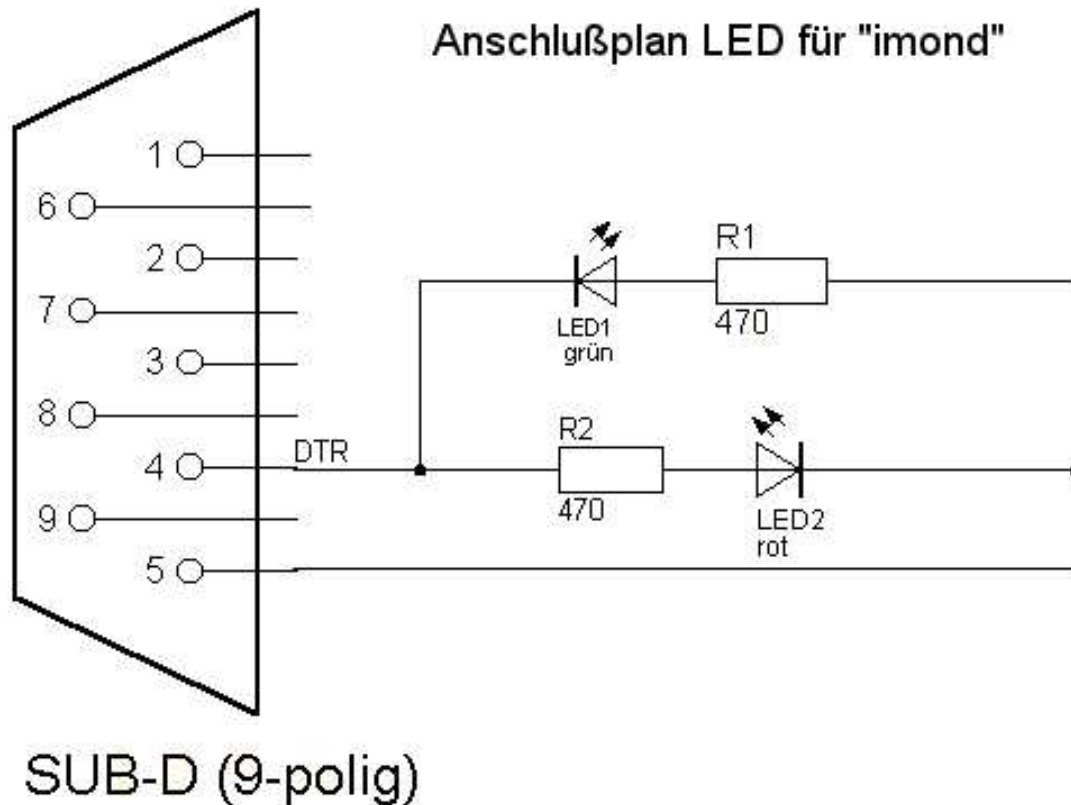
imonc /s: 192.168.0.1

8 Verwendung der LED-Anzeige zum Anzeigen des Online-Status

Wie schon weiter oben mal erwähnt, kann "imond" eine an die serielle Schnittstelle angeschlossene Leuchtdiode dazu benutzen, den Online-Status des Server-Systems anzuzeigen. Ich finde diese Funktion sehr sinnvoll. Der Status ist sofort sichtbar, auch wenn man nicht gerade am PC auf dem "imond" läuft arbeitet. Diese Funktion hat mich ursprünglich dazu veranlaßt, das Programm "imond" auf einem normalen Linux-System zu benutzen.

An der Konfiguration des Server-Daemons "imond" ändert sich nur ein Punkt, wenn man die LED-Anzeige nutzen möchte. In der Datei `/etc/imond.env` muß in der Variablen `IMOND_LED` die verwendete serielle Schnittstelle angegeben werden, zum Beispiel: `IMOND_LED='ttyS0'` für die erste serielle Schnittstelle (COM1).

Nun muß man natürlich noch die Leuchtdiode anschließen. Hier zunächst einmal der schematische Schaltplan:



Eine kurze Erklärung zur Funktion der Schaltung: Die serielle Schnittstelle in PCs (RS232) verwendet zwei Spannungen: -12 V und +12 V. "imond" macht nichts anderes, als diese Spannung entsprechend des Online-Status umzustellen. Die Leuchtdiode hat nun den Vorteil, daß sie wie alle Dioden den Strom nur in einer Richtung passieren läßt. Das hat zur Folge, daß die Leuchtdiode entweder nur bei -12 V oder nur bei +12 V leuchtet. Wie aus dem Schaltplan ersichtlich ist, verwende ich zwei Leuchtdioden und nutze dabei genau diesen Effekt aus. Die Leuchtdioden sind mit gegensätzlicher Polarität eingebaut. Die beiden Widerstände dienen nur als Vorwiderstände für die Leuchtdioden die ansonsten durch die Hohe Spannung zerstört werden würden.

Die Bauteile für diese Schaltung sind sehr günstig. Die Widerstände und Leuchtdioden kosten nur je ein paar Cent. Als Anschlußkabel habe ich das Kabel einer alten seriellen Maus verwendet. Wer so ein Relikt nicht mehr hat, kann so ein Kabel aber auch kaufen oder aus einem Stecker (9pol. SUB-D) schnell selber bauen. Aufgebaut habe ich die ganze Schaltung auf einer Streifenraster-Platine. Das reicht vollkommen aus. Es würde auch als fliegende Schaltung funktionieren.

9 Anhang

Im Anhang stelle ich weiteres Material zum Thema bereit.

9.1 Liste verwendeter Dateien und ihr Speicherort

In dieser Tabelle möchte ich einen Überblick über die Dateien geben, die im Laufe dieses HowTos angelegt wurden und solche, die für die Ausführung von "imond" benötigt werden.

Datei und Pfad	Funktion
/etc/imond.conf	Konfigurationsdatei, die Angaben zu dem/den Internet-Providern enthält
/etc/imond.env	Konfigurationsdatei, die das Verhalten von "imond" steuert (enthält auch Passwörter und Berechtigungen)
/usr/local/bin/imond	ausführbares Programm "imond"
/usr/local/bin/fli4lctrl	ausführbares Script mit den Steuerbefehlen für den Verbindungsaufbau und -abbau
/etc/init.d/imond	Boot-Script (kann auch an anderer Stelle abgelegt werden (distributionsabhängig)
/etc/ppp/ip-up.d/imond-ip.ip-up	Script zum Abspeichern der IP-Adresse beim Verbindungsaufbau
/etc/ppp/ip-up.d/imond-ip.ip-down	Script zum Löschen der Datei mit der IP-Adresse beim Verbindungsabbau
/var/run/ <i>interfacename</i> .ip	Datei mit der aktuellen InterNet-IP-Adresse (existiert nur bei bestehender InterNet-Verbindung und konfigurierter IP-Anzeige)

9.2 Download

Hier biete ich oben erwähnte Archive zum Download an. Hier sind nur von mir erstellte Archive aufgelistet. Besonders das fli4l-Paket muß von der [fli4l-Homepage](#) heruntergeladen werden.

Archiv	Inhalt
---------------	---------------

imond4linux.tar.gz

Archiv mit allen oben erwähnten Beispiel-Dateien (Script- und Konfigurationsdateien) und diesem HowTo im HTML-Format

Ich habe mir beim erstellen der Archive größte Mühe gegeben alle Dateien in der neusten Version einzupacken. Leider ist irren menschlich. Bitte benachrichtigen Sie mich, wenn mit den Archiven etwas nicht in Ordnung ist.

9.3 Dank

An dieser Stelle möchte ich all den Leuten danken, die mich mit hilfreichen Tipps und Lösungsvorschlägen bei der Erstellung dieses How-Tos unterstützt haben. Zugleich möchte ich auch denen danken, die mich auf Fehler in diesem How-To aufmerksam gemacht haben. Und das wären im einzelnen ...

Holger Reichert, info@reichertcomputer.de
Thomas Frye, t.frye@ping.de
Mario Kleinboelting, mario@kleinboelting.de

9.4 Kontakt

Nach dem Sie all das hier überstanden haben, würde ich mich über Ihre Meinung sehr freuen. Ich habe viele Stunden an diesem HowTo gearbeitet und bin schon auf das Feedback gespannt. Ich bin für Lob genauso dankbar wie für konstruktive Kritik!

Und so erreichen Sie mich: Am besten per e-Mail unter heidbrede@freenet.de oder manchmal auch im ICQ unter der UID 123543704.

Ich freue mich auf Ihre e-Mails !

9.5 History

Eine neue Version dieses How-Tos wird unter www.heidbrede-bs.de oder www.thetinker.de.vu veröffentlicht !

13. Juni 2004 keine inhaltlichen Veränderungen; das Archiv mit dem How-To und den Dateien wurde korrigiert; das How-To ist nun ebenfalls als PDF- Dokument verfügbar
15. Mai 2004 Umzug auf eine neue Adresse (von heidbrede.de nach heidbrede-**bs**.de
kleine Ergänzung im Punkt 7 - Windows Client (Hinweis zum Aufruf hinzugefügt)
3. April 2004 Korrektur einiger kleiner Fehler
29. Februar 2004 Umstellung der Formatierung auf CSS-Stylesheets
Konfiguration der Anzeige der IP-Adresse im Client hinzugefügt
Abschnitt "Dank" hinzugefügt
3. Januar 2004 Fehlerkorrektur im Abschnitt "kompilieren"
8. Oktober 2003 erste Veröffentlichung im InterNet unter
people.freenet.de/heidbrede/imond/

Impressum

Ich bin privater Linux-Anwender und stelle mein Wissen an dieser Stelle kostenlos allen Interessenten zur Verfügung. Alle Angaben ohne Gewähr. Eine Haftung meinerseits ist ausgeschlossen. Ich übernehme keine Verantwortung für den Inhalt der von mir gelinkten Seiten oder für die Funktion der hier erwähnten Software.

Dieses How-To basiert auf dem "[Konfiguration von telmond und imond unter Linux - HowTo](#)" von [Frank Meyer](#).

Dieses Dokument darf kostenlos gerne weitergegeben werden, solange daran keine Änderungen vorgenommen wurden. Sollte jemand auf die Idee kommen, dieses Dokument auf einer CD, für die er Geld nimmt, zu verbreiten, braucht dieser jemand eine schriftliche Einwilligung des Autors.

Ich möchte mit diesem Dokument die Linux-Gemeinde fördern und verfolge damit keine egoistischen Absichten. Ich möchte auch nicht irgendwelche Urheberrecht oder ähnliches Verletzten.

© 2003-2004 by Ingwer Heidbrede

