

FAI - ein Tool zur automatischen Installation von Debian GNU/Linux

Thomas Lange

lange@informatik.uni-koeln.de

Institut für Informatik, Universität zu Köln

3. Braunschweiger Linux-Tage, 4-6. Mai 2001

1

FAI - Fully Automatic Installation

3. Braunschweiger Linxutage

Was ist FAI ?

- FAI macht alles, was ihr Systemadministrator zu tun hat, bevor Sie das erste Mal auf einem neuen Rechner arbeiten können
- Es ist eine Sammlung von Skripten zur vollautomatischen Installation von Debian GNU/Linux
- Es kann das ganze Betriebssystem und die Anwendungsprogramme installieren und konfigurieren

Motivation

- Manuelle Installation dauert einige Stunden
- Viele Fragen sind zwischendurch zu beantworten
- Gleiche Daten müssen bei jedem Rechner erneut eingegeben werden
- Wiederholende Arbeit führt zu Fehler
- Automatische Installation dauert nur wenige Minuten
- Es kann sehr viel Zeit gespart werden !
- Nach Hardwaredefekt ist der Rechner in kürzester Zeit wieder mit identischer Konfiguration einsatzbereit
- Cluster, Serverfarmen, Pools wegen gleicher Hardware ideal

Voraussetzungen

- Rechner mit Netzwerkkarte
- Server mit BOOTP oder DHCP, NFS und TFTP Diensten (muß kein Linux sein)
- Lokaler Spiegel von Debian (NFS, FTP oder HTTP)
- Plattenplatz auf dem Server:

FAI Paket	10 MB	Kernel, Skripte, Konfiguration
NFS-Root	100 MB	base2_2.tgz
Debian Spiegel	2.5 GB	Debian 2.2 (nur i386)
- Plattenplatz konstant (auch bei beliebig vielen Clients), da nur Lesezugriff und Client unabhängig

Ablauf einer Installation

- Rechner startet ein Linux mit NFS-Root, ohne lokale Platte zu benutzen
- Klassen und Variablen definieren
- Kernel Module laden
- Festplatten partitionieren
- Dateisysteme erzeugen und mounten
- Software Pakete installieren
- Betriebssystem und Anwendungen konfigurieren
- Protokolldateien auf Server speichern
- Neu installiertes System booten

- Installationszeit (ohne booten) auf Dual Pentium II 400 MHz, 128 MB RAM, 10Mbit LAN
 - 90 MB Software : 2 Minuten
 - 520 MB Software : 11 Minuten
 - 4 GB auf defekte Blöcke testen: ca. 6 Minuten
- Die Installationszeit bleibt bei einem Client und 100Mbit LAN gleich.
- **Die Installationszeit wird durch die Menge der Software bestimmt.**

Booten

- Via Netzwerkkarte mit TFTP und BOOTP oder DHCP
- `make-fai-bootfloppy` erstellt universelle Boot-Diskette
- Kernelparameter sind möglich
- FAI Parameter via BOOTP oder DHCP:
 - `$FAI_LOCATION (T170)`: Konfigurationsverzeichnis
 - `$FAI_ACTION (T171)`: `sysinfo`, `install`
 - `$FAI_FLAGS (T172)`: `verbose`, `debug`, `sshd`, `reboot`
- remote Zugriff während der Installation via `ssh`
- zwei weitere Konsolen, keine initial Ramdisk notwendig, 8MB RAM reichen, 386 CPU kompatibel

```

/etc/bootptab:

.faiglobal:\
:ms=1024:hd=/boot/fai:hn:bs=auto:rp=/usr/lib/fai/nfsroot:

.failocal:\
:tc=.faiglobal:sa=kueppers:ts=rubens:\
:T170="kueppers:/usr/local/share/fai":T171="sysinfo":\
:sm=255.255.255.0:gw=134.95.9.254:\
:dn=informatik.uni-koeln.de:\
:ds=134.95.9.136,134.95.100.209,134.95.100.208:\
:ys=rubens:yd=informatik4711.YP:\
:nt=time.rrz.uni-koeln.de,time2.rrz.uni-koeln.de:

faiclient99:ha=0x02608c7b40d6:bf=faiclient99:\
:tc=.failocal:T171="install":T172="sshd verbose"

```

Die Konfiguration

- Wie heißt der Rechner ? Welche IP-Adresse ?
- Wie sollen die Platten partitioniert werden ?
- Wo und wie werden die Partitionen gemountet ?
- Welche Software soll wie installiert werden ?
- Welche lokalen Anpassungen sind nötig ?
 - Root Paßwort, Accounts, Zeitzone, Tastatur, Kernel
 - NTP, NIS, inetd, ftp, ssh, lpr, autofs, X11, lilo
 - /etc/fstab, /etc/exports, /etc/hosts.allow
- Summe aus: Klassen, Variablen, Skripte, Prototypen
- **Ist für all dies ein Konzept vorhanden ?**

Klassen

- Ein Rechner gehört zu mehreren Klassen
- Liste aller Klassen eines Rechners in `$classes`. Reihenfolge !
- Klassennamen bestehen aus: `[0-9A-Z_]` oder `hostname`
- Klassen werden wie folgt definiert:
 - Vordefinierte Klassen:** DEFAULT, `hostname` und LAST
 - In einer Datei:** alle Klassennamen in einer Datei werden genutzt
 - Dynamisch über Skripte:** ein Skript gibt z.B. abhängig von der Hardware Klassennamen aus

```

# two SCSI disks 2-5 GB
($numdisks == 2) and
    disksize(sda,2000,5000) and
    disksize(sdb,2000,5000) and
    class("SD_2_5GB");

# one disk 1-4 GB, IDE or SCSI
($numdisks == 1) and
    testsize($sum_disk_size,1000,4000) and
    class("4GB");

-----

#! /bin/sh
if [ -n "$YPDOMAIN" ];then
    echo NIS
    echo $YPDOMAIN | tr '.a-z-' '_A-Z_'
else
    echo NONIS
fi

```

Variablen

- Variablen werden abhängig von Klassen definiert
- Konfigurations-Skripte sind durch Variablen generischer

```

.../class/DEFAULT.var:
FAI_CONSOLEFONT=
FAI_KEYMAP=us-latin1
UTC=yes
time_zone=Europe/Berlin
rootpw='a3hxVqR5t1t9L'
liloappend='append="mem=320M"'
kernelimage=kernel-image-2.2.19-idepci
printers="kyocera optra hp4si hp hpcolor juenger"

.../class/ATA33.var:
hdparm='hdparm -c1 -d1 -m16 -X66 /dev/hda'

```

Festplattenkonfiguration

- Im Verzeichnis `/fai/disk_config/` gibt es beispielsweise die Datei `4GB`:

```
# <type> <mountpoint> <size> [mount opt] [;extra opt]
disk_config hda
primary /          50          rw,errors=remount-ro ;-c
logical swap      100-200    rw
logical /var      150-200    rw
logical /usr      1500       rw
logical /tmp      100-300    ;-m 1
logical /home     700-       rw,nosuid  ;-m 0
logical /scratch  0-         rw,nosuid  ;-m 0 -i 50000
#logical /scratch preserve9 rw,nosuid  ;-m 0 -i 50000
```

Merkmale der Plattenkonfiguration

- Eine Konfigurationsdatei für alle lokalen Platten
- Einfache Angabe von Größe, Mountpoint und Optionen
- Partitionsgröße fest oder variabel
- Daten auf einzelnen Partitionen können bewahrt werden (preserveN)
- Optionen für mkfs und mount, z.B. nosuid, ro
- Windows Partitionen können in `/etc/fstab` mit eingetragen werden
- Automatische Erzeugung von `/etc/fstab`

Software Konfiguration

Beispiel für die Klasse COMPILER:

```
# COMPILER: packages for developing software
PACKAGES install
task-c-dev task-c++-dev
bin86 binutils m4
libc6-dev libg++2.8.1.3 libstdc++2.10-dev
libstdc++2.10 g77 byacc cvs
```

- Aktionen: install, remove
- Abhängigkeiten werden durch `apt-get` aufgelöst
- auch Format von `dpkg --get-selections` möglich

Klassenkonzept

- *Nutze die Dateien, deren Name als Klasse definiert ist*
- Konfiguration wird aus Bausteinen (Klassen) zusammengesetzt
- Jeder Rechner ist in mehreren Klassen enthalten
- Jede definierte Klasse bestimmt einen Teil der Konfiguration
- Einfaches Hinzufügen einer Klasse erweitert die Konfiguration für einen Rechner
- Klassenbeispiele: COMPILER, FALBOOTPART, NIS, DATALESS, NOTEBOOK, MBR, DEBIAN_DEVEL, 4GB, KERNEL_SOFT, BOOTP_SERVER

Beispiele für Konfigurationskripte

```

.../scripts/DEFAULT:
#! /bin/sh
chmod 1777 $target/tmp
chown root:root $target/tmp

# create NIS/NONIS config
fcopy /etc/nsswitch.conf /etc/host.conf
ifclass NONIS && rm -f $target/etc/defaultdomain
if ifclass NIS; then
    echo $YPDOMAIN > $target/etc/defaultdomain
    rm -f $target/etc/yp.conf
    for s in $YPSRV; do
        echo "ypserver $s" >> $target/etc/yp.conf
    done
fi

```

```

# copy default dotfiles for root account
fcopy /root/.bash_profile /root/.bashrc /root/.cshrc

ln -s /boot/vmlinuz-nolink $target/vmlinuz
if [ -f $files/packages/$kernelimage ]; then
    yes 'n'|dpkg --root=$target -i $files/packages/$kernelimage
else
    yes 'n'|chroot $target apt-get install $kernelimage
fi

-----

.../scripts/NOTEBOOK
#! /bin/sh
maxcount=101
partitions='mount |grep /tmp/target|grep "type ext2"| \
            perl -ane 'print "$F[0] "'
for part in $partitions; do
    tune2fs -c $maxcount $part
done

```

```

#! /usr/bin/cfengine
control:  actionsequence = ( editfiles )
editfiles:
any::
  { ${target}/etc/passwd
    LocateLineMatching "^root:.*"
    InsertLine    "roott::0:0:root:/root:/usr/bin/tcsh"
    ReplaceAll    "^root::" With "root:${rootpw}:"
    ReplaceAll    "^roott::" With "roott:${rootpw}:"
  }
  { ${target}/etc/default/rcS
    ReplaceAll    "^UTC=.*" With "UTC=${UTC}"
  }
HOME_CLIENT::
  { ${target}/etc/fstab
    HashCommentLinesContaining "/home"
    AppendIfNoSuchLine "${hserver}:/home /home nfs"
  }

```

Prototypen installieren

- Für viele Konfigurationsdateien gibt es vorbereitete Prototypen
- Jede Installation braucht eigenen Prototyp
- Kopieren dieser Dateien klassenabhängig mit `fcopy`
- Extrahieren von Archiven klassenabhängig mit `ftar`

```

/usr/local/share/fai/files
|-- etc
|  |-- nsswitch.conf
|  |  |-- NIS
|  |  '-- NONIS

```

```

DEFAULT: fcopy /etc/nsswitch.conf /etc/host.conf

```

```

X11: fcopy /etc/X11/XF86Config /etc/X11/Xserver

```

Informationen über FAI

- Homepage: `www.informatik.uni-koeln.de/fai`
- Maillingliste, CVS repository
- Zusätzlich während Installation verfügbar sind: lvm, raidtools, dump, restore, ext2resize, hdparm, parted
- Zugriff auf Pakete via NFS, FTP oder HTTP
- FAI mit Aktion `sysinfo` kann als Rescue System benutzt werden. Mountet alle lokalen Partitionen automatisch, findet `/etc/fstab`
- Tests mit Linux auf SUN SPARC erfolgreich
- Neues Release FAI 2.0 verfügbar im Mai 2001

Eigenschaften von FAI

- Voll automatische Installation ! Keine Interaktion
- Garantiert identische, konsistente Installationen
- Einfaches Erstellen der Konfiguration durch Klassen
- Konstanter Plattenplatz auf dem Server
- Alle Konfigurations- und Protokolldateien zentral auf dem Server
- **FAI ist eine skalierende Methode zum Installieren von Debian GNU/Linux**