

FAI - a tool for the fully automatic installation of Debian GNU/Linux

Thomas Lange

lange@informatik.uni-koeln.de

Institut für Informatik, Universität zu Köln

Debian One, Bordeaux, july 2001

What is FAI ?

- It's **not**: Fournisseur d'Accès Internet
- FAI does everything a system administrator (you) have to do, before a user can log in a brand new computer for the first time
- It's a collection of scripts for the fully automatic installation of Debian GNU/Linux
- It can install and configure the whole operating system and the applications

Motivation

- Manual installation takes long time (several hours)
- You have to answers many questions
- Equal configuration data must be entered for every new computer again and again
- Repeating work leads to errors
- An automatic installation only lasts several minutes
- You can save much time !
- After the replacement of defect hardware, the computer is ready for action in a few minutes using the same configuration as before

Requisites

- Computer with network interface card called install client
- A server with BOOTP or DHCP, NFS and TFTP daemon (linux not needed) called install server
- Access to a local Debian mirror via NFS, FTP or HTTP
- Disk space on the install server:

Debian package	10 MB	kernel, scripts, configuration
nfsroot	100 MB	base2_2.tgz and additional software
Debian mirror	2.5 GB	Debian 2.2 (i386 only)
- Constant disk space (even for many install clients), due to read only access

Sequence of an installation

- install client boots linux using the nfsroot filesystem, but does not use local hard disks
- define classes and variables
- load kernel modules
- partition local hard disk (using setup_harddisks and sfdisk)
- create file systems (mke2fs, mkswap)
- install software (apt-get)
- configure operating systems and applications
- save log files to install server
- boot newly installed system

- Time for a installation (without booting) on
Dual Pentium II 400 MHz, 128 MB RAM, 10Mbit LAN
 - 90 MB software : 2 minutes
 - 520 MB software : 11 minutes
 - test 4 GB for bad blocks: about 6 minutes
- The time for installation remains unchanged for one client
and 100Mbit LAN.
- **The amount of software determinates the installation
time.**

Booting

- Via bootable network card using TFTP and BOOTP or DHCP
- `make-fai-bootfloppy` creates an universal boot floppy
- kernel parameters can be specified
- FAI parameters are passed via BOOTP or DHCP:
 - `$FAI_LOCATION` (T170) : location of the configuration
 - `$FAI_ACTION` (T171) : sysinfo, install
 - `$FAI_FLAGS` (T172) : verbose, debug, sshd, reboot
- remote access during the installation via ssh
- two more virtual consoles, no initial ramdisk needed, 8MB RAM suffice, 386 CPU compatible

/etc/bootptab:

```
.faiglobal:\n:ms=1024:hd=/boot/fai:hn:bs=auto:rp=/usr/lib/fai/nfsroot:\n\n.failocal:\n:tc=.faiglobal:sa=kueppers:ts=rubens:\\n:T170="kueppers:/usr/local/share/fai":T171="sysinfo":\\n:sm=255.255.255.0:gw=134.95.9.254:\\n:dn=informatik.uni-koeln.de:\\n:ds=134.95.9.136,134.95.100.209,134.95.100.208:\\n:ys=rubens:yd=informatik4711.YP:\\n:nt=time.rrz.uni-koeln.de,time2.rrz.uni-koeln.de:\\n\nfaiclient99:ha=0x02608c7b40d6:bf=faiclient99:\\n:tc=.failocal:T171="install":T172="sshd verbose"
```

The Configuration

- hostname, IP-address, hard disk partition layout, file systems, mount points, software to install, local configuration of operating system and applications
- local customization
 - root password, accounts, timezone, keyboard layout, special kernel, NTP,NIS, inetd, ftp, ssh, lpr, autofs, X11, lilo
 - `/etc/fstab`, `/etc(exports`, `/etc/hosts.allow`
- Sum of: classes, variables, scripts, prototypes of config files
- Do you have a concept for all this ?
- **Plan your installation, and FAI installs your plans !**

Classes

- An install client belongs to many classes
- List of all classes are stored in `$classes`. Order matters !
- Classes are composed of: [0-9A-Z_] or the hostname
- Classes can be defined as follows:
 - predefined classes:** DEFAULT, hostname and LAST
 - listed in a file:** all classes are listed in a file
 - dynamically generated:** a script can define classes depending on hardware found and write them to stdout

Class concept

- Slogan: *Use each file, which file name is defined as a class*
- The configuration is composed of classes
- Each class defines a part of the configuration
- Adding a class easily expands the configuration for a host
- Examples of classes: COMPILER, FAI_BOOTPART, NIS, DATALESS, NOTEBOOK, MBR, DEBIAN_DEVEL, 4GB, KERNEL_SOFT, BOOTP_SERVER

```
#! /usr/bin/perl
# 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 $YPDOMAIN" | tr '.a-z-' '_A-Z_'
else
    echo NONIS
fi
```

Variables

```
.../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'
```

Hard disk configuration

- In `/fai/disk_config/` the file `4GB` exists:

```
# <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
```

Features of the hard disk configuration

- One config file for all local disks of one install client
- Easy specification of size, mount point and options
- fixed or variable partitions size
- User data on a partition can be preserved
- Option for mkfs and mount, e.g. nosuid, ro
- Windows partitions can also be included to `/etc/fstab`
- Automatic generation of `/etc/fstab`

Software configuration

Installing software is done by the small Perl script

`install_packages`. Trick: yes '' |

Example for the class `COMPILE`:

```
# COMPILE: packages for developing software
PACKAGES install
task-c++-dev task-debug
bin86 m4 g77 byacc cvs
```

- Actions: install, remove
- Dependencies are solved by `apt-get`
- `dpkg --get-selections` format is possible

Example for configuration scripts

```
.../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 $YPSRVR; 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"
}
```

Install prototypes

- Many configuration files have prepared prototypes
- Each installation needs a different prototype
- Copying of these files depending on classes using `fcopy`
- Extracting of archives depending on classes using `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`

Information about FAI

- Homepage: www.informatik.uni-koeln.de/fai
- Mailling list, CVS repository
- Additional software available during installation: lvm, raidtools, dump, restore, ext2resize, hdparm, parted,...
- Access to Debian mirror via NFS, FTP oder HTTP
- Log files are also written to the install server
- FAI with action `sysinfo` can be used as a rescue system. It mounts all local partitions automaticly using `/etc/fstab` if available
- Successful tests on SUN SPARC (sfdisk problem)

Summary

- Fully unattended installation ! No interaction needed
- Identical, consistent installations guaranteed
- Easy creation of configuration using classes
- Constant disk space on install server
- All configuration and log files are store centrally on the install server auf dem Server
- **FAI is a scalable method for installing Debian GNU/Linux**

What's next

1. Woody support ! (debootstrap)
2. Internationalization (gettext in shell scripts ?)
3. Kernel 2.4
4. new disk partition tool (replacement for sethup_harddisks)
 - will be written in perl
 - using parted or fdisk
 - non DOS disklabels support (for SPARC architecture)
 - devfsd, raid, reiserfs and better vfat support
 - always clean `/`, `/usr`, `/var`
 - preserve partition layout, preserve data inside partition
 - RFC will be announced in the FAI mailing list

Have you any hints for me ?