minit-howto v0.2.2

Nico Schottelius (nico-linux-minit@schottelius.org)

October 13, 2004

Contents

1 Preamble

1.1 Disclaimer

This document is provided without any guarantees, including merchantability or fitness for a particular use. The maintainer cannot be responsible if following instructions in this document leads to any damage (including equipment or data, angry neighbors, strange habits, divorce, or any other calamity). This disclaimer is mostly stolen from the NFS-HOWTO[1].

1.2 Copying

Copying is allowed, as long as the content is unmodified. The original source for this document is [2].

1.3 Intended audience

This document should be read by system administrators, people who like to speedup their system and everyone else interested in a new init system. You should have basic knowledge of a standard GNU/Linux system (mkdir, vi, grub, lilo, ln, cat, echo and ls are well known to you).

1.4 Feedback

I welcome feedback about how the HOWTO can be improved. You can contact me at nico-linux-minit@schottelius.org (You need to reply to a confirmation mail, otherwise your mail will never reach me).

2 Introduction

2.1 What is minit?

Every UNIX-variant needs a program, which manages the start process, after the bootloader executed the kernel. Minit[3] is such a program. This howto explains how minit works and how to use it.

2.2 How it works

For minit, everything is a service, which should either be started once or respawned. Minit expects its configuration in the **/etc/minit** directory. Minit searches for a file called "'depends"' in the subdirectory **default**. This file tells minit what services should be started by default (in fact it tells minit that the service default depends on other services, but "'default"' is the directory minit starts to search in). A typical **default/depends** file looks like that:

```
scice% cat default/depends
mount
network
local-tuning
local-services
remote-services
getty
```

Minit now goes parallel to the directories mount, network, etc. Those directories can either include "'depends"' files again, or if no dependency exists, a "'run"' file. Let us have a look into the directory "'getty"':

```
scice% ls -l getty/
total 4
drwxr-x---  2 nico nico 43 Dec  9  2003 1
drwxr-x---  2 nico nico 43 Dec  9  2003 2
drwxr-x---  2 nico nico 43 Dec  9  2003 3
drwxr-x---  2 nico nico 43 Dec  9  2003 4
drwxr-x---  2 nico nico 43 Dec  9  2003 5
drwxr-x---  2 nico nico 43 Dec  9  2003 6
drwxr-x---  2 nico nico 43 Dec  9  2003 7
drwxr-x---  2 nico nico 43 Dec  9  2003 8
-rw-r-----  1 nico nico 81 Feb 24  2004 depends
scice% cat getty/depends
network/hostname
getty/1
getty/2
getty/3
getty/4
getty/5
getty/6
getty/7
getty/8
```

So getty itself depends on other services (which can depend on other services, too). The directory "'getty/1"' has no dependencies, but the three files "'param"', "'respawn"' and "'run"':

```
scice% ls -l getty/1
total 4
-rw-r-----  1 nico nico 20 Dec  9  2003 params
-rw-r-----  1 nico nico  0 Dec  9  2003 respawn
lrwxrwxrwx  1 nico nico 12 Aug 25 08:08 run -> /sbin/fgetty
scice% cat getty/1/params
/dev/vc/1
--noclear
```

run is the program minit should run, when starting this service. **params** contains the paramaters passed to the program. One parameter on each line. And the existence of the zero-sized file **respawn** tells minit to restart the service when it exits.

2.3 Using standards

Minit is not LSB-conform[4], nor does it won't to be. Minit uses the standard Un*x-Filesystem features like links, directories and files. We can describe it as a complete new init system, throwing away the problems of traditional init-systems.

3 Installing

3.1 Download and build

Either download the source package from [3] or use CVS. Attention: If you use CVS, **/sbin/shutdown** will be overwritten. That means if you call "'reboot"', "'halt"' or "'shutdown"' after installing minit, they call the shutdown from minit. Please do a backup of it before (`cp /sbin/shutdown /sbin/shutdown.sysvinit`), so you willl be able to reboot and halt with /sbin/shutdown.sysvinit. We will use the CVS version: `cvs -d :pserver:cvs@cvs.fefe.de:/cvs co minit`. If you don't have dietlibc[6] installed, you need to unset the variable "'DIET"'. You can either comment it out in the Makefile (edit the Makefile and replace "'DIET=diet"' with "'#DIET=diet"') or simply prefixing it to make:

```
# build process for systems with dietlibc
scice% make

# build process for systems without dietlibc
scice% DIET="" make
```

After that simple type `make` and wait some time. Then install minit (as root) with `make install` . Now minit is installed and created the directory **/etc/minit** with two FIFOs:

```
# your new /etc/minit dir
scice# ls -l /etc/minit
total 0
prw-------  1 root root 0 Aug 25 11:48 in
prw-------  1 root root 0 Aug 25 11:48 out
```

3.2 Creating basic directories

Now we need to create the services minit should manage. First of all we change to the directory **/etc/minit** and create the directory **default**

(`cd /etc/minit; mkdir default`). Then we add the service getty to the standard startup (`echo getty >> default/depends`). Dependencies are always relative to the **/etc/minit** directory. Now we create the service getty, which allows us to login to our system. First we create the directory **getty** (below **/etc/minit**) (`mkdir getty`), then we add the link to our getty and specify the parameters. There are different steps for different gettys and systems with and without devfs[5]. The example below uses devfs, if you don't use it, replace "'vc/..."' with "'tty..."' (vc/1 becomes tty1).

```
# agetty (Debian uses it as /sbin/getty)
scice# ln -s /sbin/getty getty/run
scice# echo 38400 > getty/params
scice# echo vc/1 >> getty/params

# fgetty and mingetty
scice# ln -s /sbin/fgetty getty/run
scice# echo /dev/vc/1 > getty/params
scice# echo --noclear >> getty/params

# every getty needs to restart
scice# touch getty/respawn
```

You can also use the mini-script[7] to create those basic entries. After you finished, you have to adjust your bootloader to use minit: With LILO you add **append="init=/sbin/minit"** to **/etc/lilo.conf**, with grub you simply add **init=/sbin/minit** behind the `kernel` entry. It is recommened to add a new bootloader entry for use with minit, so you can easily change back to your previous system. After that simply reboot with `/sbin/shutdown.sysvinit -r`. Your system should startup and display a (more or less) normal login prompt.

3.3 Things to do

Well, now you have a read-only mounted system, without keyboard settings, without anything like /proc or /sys mounted. This is just an early version of the minit howto, other parts are still missing. You can download a copy of my minit configuration[8].

3.4 Integrating Daemons

Minit tracks the programs it started. If a daemon forks to the background, minit cannot follow it. In the minit-suite is a program called "'pidfilehack"' included, which helps you to start daemons. I will show you how to do that with, for instance, Openssh[9]. Instead of linking **run** to the program, you link to **/sbin/pifilehack**. The **params** file now contains additional arguments for **pidfilehack**:

```
scice% ls -l remote-services/sshd/
total 4
-rw-r-----  1 nico nico 54 Dec 31  2003 params
-rw-r-----  1 nico nico  0 Nov 14  2001 respawn
lrwxrwxrwx  1 nico nico 17 Aug 25 08:08 run -> /sbin/pidfilehack
scice% cat remote-services/sshd/params
remote-services/sshd
/var/run/sshd.pid
/usr/sbin/sshd
```

The first paramater of **pidfilehack** is the servicename (Ask yourself: In which directory is the service?) and the second the pid-file of the daemon. The third is the path to the daemon, all following are parameters to the daemon itself.

3.5 Recommended hierarchy

After some testing you'll notice that some hierachies are less good than others. I recommened the one found in my minit configuration[8], as it is extendable and also well sorted:

Table 1: Recommened hierarchy

ctraltdel	ctr-alt-del service
default	default startup service (required)
getty	contains all gettys: scice% ls getty 1 2 3 4 5 6 7 8 depends scice% ls getty/1 params respawn run
halt	service called when shutting down
in	minit-in-fifo (required)
kbreq	keyboard request service
local-services	Services only locally reachable: For instance your local MTA (qmail without tcpserver) or mouse driver (gpm).
local-tuning	Cleanups, driver loading, system tuning loading keyboard driver (dvorak, qwertz, etc.) permissions configuration, kernel settings, ...
local-untuning	saving kernel settings (like sound settings at halt/reboot)
mount	mount root read/write, mount other (encrypted) devices
network	startup networking
out	minit-out-fifo (required)

reboot	reboot service
remote-services	web-servers, dns-servers, ssh-server, etc.

4 Internals

4.1 Special directories

4.1.1 /etc/minit

This is the base directory. Everything else is below this directory.

4.1.2 ctraltdel

What should be done if **ctr+alt+del** is pressed? If this directory does not exist, nothing will happen when you press ctr+alt+del.

4.1.3 kbreq

What to do when **alt+uparrow** is pressed. If this directory does not exist, nothing will happen when pressing the "'keyboard request"' (Alt+UP-Arrow under x86).

4.1.4 default

The services minit starts when booting the system. If this directory does not exist, nothing will happen at system startup. Quite bad.

4.1.5 halt

The services minit kills (newer versions of minit will also be able to start services) when halting the system. If this directory does not exist, nothing will happen when halting the system.

4.1.6 reboot

The services minit kills (newer versions of minit will also be able to start services) when halting the system. If this directory does not exist, nothing will happen at reboot.

4.1.7 log

If this directory exists [within another service directory], it is taken as service and minit creates a pipe between stdout of this service and stdin of the log service. If the log service can not be started, this service will block if it

writes to stdout. File descriptors will be reused, i.e. if the log process dies and is restarted, no log entries will be lost and there will be no SIGPIPE[1].

4.2 Special files

4.2.1 in and out

in and **out** are the FIFOs minit communicates with its helper programs.

4.2.2 depends

Contains the depenencies of the service relative to the **/etc/minit** directory.

4.2.3 run

The run file is either a link or an executable, which is run, when the service it is in is started.

4.2.4 params

The paramaters to pass to the program which is run. One parameter on one line.

4.2.5 respawn

If existent, restarts the service when it exits.

4.2.6 sync

All other services have to wait for this one. Useful for mounting.

4.3 Other directories and files

If not used in any depends file, other directories and files are silently ignored. This allows us to add distribution specific or own files like READMEs to the service directories.

5 Other init systems

There are many other init systems out there (including runit[11], needscheme[12]). The two best known are "'Sys-V-Init"' and the "'BSD-init"', which are also described here for reference.

[1] The log section was written by Fridtjof Busse, [10].

5.1 Sys-V-Init

Sys-V-Init[13] is the traditional init system under Linux. It has the central configuration file **/etc/inittab**. Sys-V-Init distinguishs between different system status, the so called "'runlevels"':

Table 2: Sys-V-Init runlevels

Runlevel	Description
0	**halt**: This level is called when shutting down the system.
1	**single-user**: This level is used for emergency repairing. Only one local user can work with the system
2-5	**multiuser**: Each UNIX, each Linux distribution has different settings: Some use runlevel 2 as normal runlevel, some use runlevel 2 for normal operating, but without the graphical environment.
6	**reboot**: This level is called when rebooting the system.

At bootup, init calls the system initialization script (**etc/init.d/rcS** for instance). When entering a new runlevel (at system boot the runlevel is undefined, sometimes described with "'N'"), Sys-V-Init executes a master script (**/etc/init.d/rc** for instance). This script begins to start all **start-scripts** for this runlevel and before to execute all **stop-scripts** from the previous runlevel. Those scripts are found either in **/etc/init.d**, **/etc/rc.d** or **/sbin/init.d**. Each runlevel has a directory (like **/etc/rc2.d**, **/etc/rc6.d**, ...). Those directories contain links to the real scripts. Sys-V-Init detects whether to stop or to start a script by its name: Start-scripts begin with a "'**S**"' at the beginning, stop-scripts with a "'**K**"' (K as in "'kill"'). When starting the start-scripts, they are called with "'start"' as the first and only parameter. When starting stop-scripts, they are called with "'stop"' as first parameter. This is how a configuration for runlevel 2 could look like:

```
scice% ls -l /etc/rc2.d
total 0
lrwxr-xr-x  1 root root 18 Feb 20  2004 S10sysklogd -> ../init.d/sysklogd
lrwxr-xr-x  1 root root 15 Feb 20  2004 S11klogd -> ../init.d/klogd
lrwxr-xr-x  1 root root 13 Feb 20  2004 S14ppp -> ../init.d/ppp
lrwxrwxrwx  1 root root 15 Feb 24  2004 S15bind9 -> ../init.d/bind9
lrwxrwxrwx  1 root root 18 Feb 20  2004 S18quotarpc -> ../init.d/quotarpc
[...]
scice% ls -l /etc/rc6.d
lrwxrwxrwx  1 root root 15 Aug 31 16:22 /etc/rc6.d/K19aumix -> ../init.d/aumix
lrwxrwxrwx  1 root root 15 Apr 19 09:55 /etc/rc6.d/K19samba -> ../init.d/samba
lrwxrwxrwx  1 root root 13 Feb 25  2004 /etc/rc6.d/K20gpm -> ../init.d/gpm
[...]
```

The number behind the S or the K is the priority. The lower the priority,

the earlier the script is started. That way one can make orders what to start or stop. Each service has an own init-script, which sometimes also accepts "'restart"' or "'reload"' instead of "'start"' or "'stop"'.

5.2 BSD-Init

Today there exists more than one "'BSD-init"'. This section describes the "'traditional"' BSD-Init, which is also implemented by OpenBSD[14]. At booting up **/etc/rc**, the master script, is called. The configuration is read from the master configuration file **/etc/rc.conf** (and also additional ones like **rc.conf.local**, **rc.local**, **rc.securelevel** and at shutdown **rc.shutdown**). As there is a very small number of scripts called, the boot process is finished quite fast, although there is no parallel execution. However, if a new package needs to install itself into the boot process, it needs to *edit* one of the existing files. This is very dangerous: If the package installer makes a small mistake, the whole startup process will fail.

References

[1] NFS-HOWTO; http://www.tldp.org/HOWTO/NFS-HOWTO/index.html

[2] This document; http://nico.schotteli.us/papers/linux/howto/minit-howto/

[3] Minit; http://www.fefe.de/minit/

[4] LSB; http://www.linuxbase.org

[5] Devfs; http://www.atnf.csiro.au/people/rgooch/linux/docs/devfs.html

[6] Dietlibc; http://www.fefe.de/dietlibc/

[7] Create-Mini-Minit entry; http://nico.schotteli.us/papers/linux/howto/minit-howto/create-mini-minit.sh

[8] My minit configuration; http://nico.schotteli.us/papers/linux/howto/minit-howto/minit-clinux-0.0.5.tar.bz2

[9] OpenSSH; http://www.openssh.org

[10] Fridtjof Busse; http://www.fbunet.de/minit.shtml

[11] runit; http://smarden.org/runit/

[12] Linux Boot Scripts; Richard Gooch; http://www.atnf.csiro.au/people/rgooch/linux/boot-scripts/

[13] Sys-V-Init; ftp://ftp.cistron.nl/pub/people/miquels/sysvinit/

[14] OpenBSD; http://openbsd.org