Bab 3

Booting dan process init

3.1 Booting

Ada beberapa tahapan proses booting dalam system operasi linux. Pertama lilo akan meload kernel, kemudian kernel akan memeriksa setiap device yang ada di mesin, dan selanjutnya akan menjalankan script init. Init adalah proses pertama yang dijalankan oleh system, init sendiri kemudian menjalankan proses-proses lain yang dijalankan pada saat booting. Init menjalankan semua proses berdasarkan `/etc/inittab`. Dalam init dikenal istilah runlevel, dalam *nix dikenal **runlevel 0-6** dan **runlevel S**. Masing-masing runlevel dijalankan berdasarkan keadaan system, runlevel 0,1 dan 6 sudah disiapkan secara default (0 untuk halt, 1 untuk single-user, 6 untuk reboot system, 2,3,4,5 untuk multi user), selain itu dapat disesuaikan dengan keinginan dari administrator sistem.

Berikut adalah isi dari file `/etc/inittab` :

```
---------------------------------------------------------------------
# /etc/inittab: init(8) configuration.
# $Id: inittab,v 1.8 1998/05/10 10:37:50 miquels Exp $

# The default runlevel.
id:2:initdefault:

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.
si::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
~~:S:wait:/sbin/sulogin

# /etc/init.d executes the S and K scripts upon change
# of runlevel.
#
# Runlevel 0 is halt.
# Runlevel 1 is single-user.
# Runlevels 2-5 are multi-user.
# Runlevel 6 is reboot.

l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin

# What to do when CTRL-ALT-DEL is pressed.
```

```
#ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
ca:12345:ctrlaltdel:/root/ctrlaltdel

# Action on special keypress (ALT-UpArrow).
kb::kbrequest:/bin/echo "Keyboard Request--
edit /etc/inittab to let this work."

# What to do when the power fails/returns.
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
# Format:
#  <id>:<runlevels>:<action>:<process>
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6

# Example how to put a getty on a serial line (for a terminal)
#
#T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100
#T1:23:respawn:/sbin/getty -L ttyS1 9600 vt100

# Example how to put a getty on a modem line.
#
#T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS3

---------------------------------------------------------------------
```

Dalam Unix dikenal istilah **single-user.** Single-user biasanya digunakan pada saat perawatan sistem. Saat sistem diharuskan dalam kondisi tak memiliki gangguan dari luar (network) ataupun user lainnya.

3.2 Mekanisme log dan pesan sistem

Di dalam Linux dikenal dua cara logging, yaitu dengan

- `syslogd`
- `klogd.`

`Syslogd` digunakan oleh berbagai macam program yang menggunakan fungsi `syslog()` untuk memasukkan catatan (log) ke dalam log file yang disediakan fasilitasnya oleh `syslogd`. File konfigurasi dari syslogd terletak di `/etc/syslog.conf`, dari file ini administrator sistem dapat menentukan dimana log file diletakkan. Secara default log file akan diletakkan di `/var/log`. Berikut adalah contoh dari file `/etc/syslog.conf`

```
---------------------------------------------------------------
#  /etc/syslog.conf     Configuration file for syslogd.
#
# For more information see syslog.conf(5)
# manpage.

#
# First some standard logfiles.  Log by facility.
```

```
#

auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                         /var/log/cron.log
daemon.*                        -/var/log/daemon.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          /var/log/mail.log
user.*                          -/var/log/user.log
uucp.*                          -/var/log/uucp.log

#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                       -/var/log/mail.info
mail.warn                       -/var/log/mail.warn
mail.err                        /var/log/mail.err

# Logging for INN news system
#
news.crit                       /var/log/news/news.crit
news.err                        /var/log/news/news.err
news.notice                     -/var/log/news/news.notice

#
# Some 'catch-all' logfiles.
#
*.=debug;\
        auth,authpriv.none;\
        news.none;mail.none     -/var/log/debug
*.=info;*.=notice;*.=warn;\
        auth,authpriv.none;\
        cron,daemon.none;\
        mail,news.none          -/var/log/messages

#
# Emergencies are sent to everybody logged in.
#
*.emerg                         *

#
# I like to have messages displayed on the console, but only on a virtual
# console I usually leave idle.
#
#daemon,mail.*;\
#       news.=crit;news.=err;news.=notice;\
#       *.=debug;*.=info;\
#       *.=notice;*.=warn       /dev/tty8

# The named pipe /dev/xconsole is for the nsole' utility.  To use it,
# you must invoke nsole' with the -file' option:
#
#    $ xconsole -file /dev/xconsole [...]
#
# NOTE: adjust the list below, or you'll go crazy if you have a reasonably
#      busy site..
#
daemon.*;mail.*;\
        news.crit;news.err;news.notice;\
        *.=debug;*.=info;\
#
*.emerg                         *

#
# I like to have messages displayed on the console, but only on a virtual
```

```
# console I usually leave idle.
#
#daemon,mail.*;\
#       news.=crit;news.=err;news.=notice;\
#       *.=debug;*.=info;\
#       *.=notice;*.=warn       /dev/tty8

# The named pipe /dev/xconsole is for the nsole' utility.  To use it,
# you must invoke nsole' with the -file' option:
#
#    $ xconsole -file /dev/xconsole [...]
#
# NOTE: adjust the list below, or you'll go crazy if you have a reasonably
#      busy site..
#
daemon.*;mail.*;\
        news.crit;news.err;news.notice;\
        *.=debug;*.=info;\
#
*.emerg                         *

#
# I like to have messages displayed on the console, but only on a virtual
# console I usually leave idle.
#
#daemon,mail.*;\
#       news.=crit;news.=err;news.=notice;\
#       *.=debug;*.=info;\
#       *.=notice;*.=warn       /dev/tty8

# The named pipe /dev/xconsole is for the nsole' utility.  To use it,
# you must invoke nsole' with the -file' option:
#
#    $ xconsole -file /dev/xconsole [...]
#
# NOTE: adjust the list below, or you'll go crazy if you have a reasonably
#      busy site..
#
daemon.*;mail.*;\
        news.crit;news.err;news.notice;\
        *.=debug;*.=info;\
        *.=notice;*.=warn       |/dev/xconsole

------------------------------------------------------------------------
```

Setelah `syslogd` hal yang perlu diketahui adalah `klogd`. `Klogd` adalah system daemon yang mencatat segala aktifitas kernel dan kemudian mendokumentasikannya ke dalam file.