

# ePinger Management

## What is active?

One can find out what processes are normally running via `ps ux`:

```
voyage:~# ps ux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2   1940   588 ?        Ss   17:34   0:01 init [2]
root         2  0.0  0.0     0     0 ?        S<   17:34   0:00 [kthreadd]
root         3  0.0  0.0     0     0 ?        S<   17:34   0:00 [ksoftirqd/0]
root         4  0.0  0.0     0     0 ?        S<   17:34   0:00 [events/0]
root         5  0.0  0.0     0     0 ?        S<   17:34   0:00 [khelper]
root        36  0.0  0.0     0     0 ?        S<   17:34   0:00 [kblockd/0]
root        41  0.0  0.0     0     0 ?        S<   17:34   0:00 [ata/0]
root        42  0.0  0.0     0     0 ?        S<   17:34   0:00 [ata_aux]
root        44  0.0  0.0     0     0 ?        S<   17:34   0:00 [kseriod]
root        80  0.0  0.0     0     0 ?        S    17:34   0:00 [pdflush]
root        81  0.0  0.0     0     0 ?        S    17:34   0:00 [pdflush]
root        82  0.0  0.0     0     0 ?        S<   17:34   0:00 [kswapd0]
root        83  0.0  0.0     0     0 ?        S<   17:34   0:00 [aio/0]
root       707  0.0  0.0     0     0 ?        S<   17:34   0:00 [kpsmoused]
root       710  0.0  0.0     0     0 ?        S<   17:34   0:00 [rpciod/0]
root      1495  0.0  0.0     0     0 ?        S<   17:34   0:00 [khubd]
root      2909  0.0  0.2   1724   704 ?        Ss   17:34   0:00 /sbin/syslogd
root      2915  0.0  0.1   1576   384 ?        Ss   17:34   0:00 /sbin/klogd -x
root      3006  0.0  0.2   1736   552 ?        Ss   17:34   0:00 /usr/sbin/pptpd
root      3040  0.0  0.2   2192   768 ?        Ss   17:34   0:00 /usr/sbin/cron
root      3048  0.0  0.6   1620  1620 ?        SLs  17:34   0:01 /usr/sbin/watchdog
root      3062  0.0  0.8   4756  2120 ?        Ss   17:34   0:00 /usr/sbin/apache
root      3097  0.0  0.1   1572   500 ttyS0    Ss+  17:34   0:00 /sbin/getty -L ttyS0 38400
root      3186  0.0  0.9   7624  2376 ?        Ss   17:41   0:00 sshd: root@pts/0
root      3191  0.0  0.9   3516  2428 pts/0    Ss   17:41   0:00 -bash
root      4447  0.0  0.2   2172   572 ?        S<S  19:46   0:00 udevd --daemon
root      6032  0.0  0.4   4852  1080 ?        Ss   19:47   0:00 /usr/sbin/sshd
root      8573  0.0  0.0     0     0 ?        Z    21:09   0:00 [watchdog] <defunct>
root      8574  0.0  0.3   2216   896 pts/0    R+   21:09   0:00 ps ux
```

One can find the active Internet listeners via `netstat -nlp`:

```
voyage:~# netstat -nlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      2830/portmap
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      3062/apache
tcp        0      0 0.0.0.0:53              0.0.0.0:*               LISTEN      6063/dnsmasq
tcp        0      0 0.0.0.0:1723            0.0.0.0:*               LISTEN      3006/pptpd
tcp6       0      0 :::53                   :::*                    LISTEN      6063/dnsmasq
tcp6       0      0 :::22                   :::*                    LISTEN      6032/sshd
udp        0      0 0.0.0.0:53              0.0.0.0:*               6063/dnsmasq
udp        0      0 0.0.0.0:67              0.0.0.0:*               6063/dnsmasq
udp        0      0 0.0.0.0:111             0.0.0.0:*               2830/portmap
udp6       0      0 :::53                   :::*                    6063/dnsmasq
```

```
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node PID/Program name  Path
```

## Finding things

The data is kept in `/usr/local/share/pinger/data/` with files of the form `ping-yyyy-mm.txt` where `yyyy` is the year and `mm` the month (1-12).

```
voyage:~# ls -l /usr/local/share/pinger/data/
total 3404
-rw-r{~}{~}r~- 1 root staff 3480630 May 18 17:31 ping-2009-05.txt
voyage:~# date
Mon May 18 21:43:31 GMT 2009
```

The CGI scripts are kept in /usr/lib/cgi-bin:

```
voyage:~# ls -l /usr/lib/cgi-bin
total 72
-rwxr-xr-x 1 root root 15101 Jan&nbsp; 7&nbsp; 2000 cgi-lib.pl
-rwxr-xr-x 1 root root 20101 May 15 23:17 ping_data.pl
-rwxr-xr-x 1 root root 36226 May 12 22:00 traceroute.pl
```

## Updating crontab

Edit /etc/crontab and save. then reboot to make the new version take effect (alternatively use kill -HUP <PID> (e.g. kill -HUP 3039) to force an update of the crontab process, where you can find the PID from the ps ux command - see above). There is an [example of a crontab](#):

## Logs

The log is kept in /var/log/messages. We believe this is "rotated" (compressed, saved and deleted) automatically by syslog using /etc/logrotate.conf and /etc/syslog.conf, so memory is not eventually exhausted.

## Applying Patches

On the one hand one wants to keep the system and applications patched to the latest releases. On the other hand this may require considerable bandwidth to download the patches, or timeouts, it could gobble up memory in the host etc.

To find out how much memory is available in the file system, use df -k:

```
voyage:~# df -k
Filesystem      1K-blocks      Used Available Use% Mounted on
rootfs          3853536      249880   3407904   7% /
udev            10240         20      10220    1% /dev
/dev/disk/by-label/ROOT_FS
3853536  249880  3407904   7% /
/dev/disk/by-label/ROOT_FS
3853536  249880  3407904   7% /dev/.static/dev
tmpfs           128416         0    128416    0% /lib/init/rw
tmpfs           128416         0    128416    0% /dev/shm
tmpfs            8192         412     7780    6% /rw
```

To get the latest patches use apt-get update:

```
voyage:~# apt-get update
Get:1 [http://security.debian.org] etch/updates Release.gpg [197B]
Hit [http://security.debian.org] etch/updates Release
Ign [http://security.debian.org] etch/updates/main Packages/DiffIndex
Hit [http://security.debian.org] etch/updates/main Packages
Get:2 [http://www.voyage.hk] ./ Release.gpg [189B]
Hit [http://www.voyage.hk] ./ Release
Ign [http://www.voyage.hk] ./ Packages/DiffIndex
Ign [http://www.voyage.hk] ./ Packages
Hit [http://www.voyage.hk] ./ Packages
Get:3 [http://ftp.hk.debian.org] etch Release.gpg [394B]
Hit [http://ftp.hk.debian.org] etch Release
Ign [http://ftp.hk.debian.org] etch/main Packages/DiffIndex
Hit [http://ftp.hk.debian.org] etch/main Packages
Fetched 3B in 6s (0B/s)
Reading package lists... Done
```

One applies the patches using apt-get update. However this can take over 45KBytes of memory. We have not decided if or how to automate the patch process.

## Testing

Look at the last date the current file in /usr/local/share/pinger/data/ was updated and compare this to date, the times should be within 30 minutes:

```
voyage:~# ls -l /usr/local/share/pinger/data/
total 3404
-rw-r--r-- 1 root staff 3480630 May 18 17:31 ping-2009-05.txt
voyage:~# date
Tue May 19 01:15:38 GMT 2009
```

If not see if pinger2 is executing properly:

```
voyage:~# cd /usr/local/share/pinger
getIPV4: voyage
BeaconList is still up to date.
Going to sleep for 29.035192998008 minutes.
pingAllHosts(): Starting to ping all hosts.
pingHost(): Enter.
queryDNSCache: www.univ-batna.dz
getIPV4: www.univ-batna.dz
pingHost(): www.univ-batna.dz.
doPing(): Expanded pingCmd resolved to '/bin/ping -n -w 21 -c 1 -i 1 -s 56 193.194.69.34'
pingHost(): Enter.
queryDNSCache: www.ucan.edu
getIPV4: www.ucan.edu
pingHost(): www.ucan.edu.
...
```

Look at the date stamps on pingerCronStat.\* and look at the contents if there is anything there:

```
voyage:~# ls -l /usr/local/share/pinger/pingerCronStat.*
-rw-r--r-- 1 root staff 0 May 18 10:30 /usr/local/share/pinger/pingerCronStat.stderr
-rw-r--r-- 1 root staff 52934 May 18 11:02 /usr/local/share/pinger/pingerCronStat.stdout
```

Check the cronjob is running

```
voyage:~# ps ux | grep cron
root      3039  0.0  0.3   2192   772 ?        Ss   May18   0:00 /usr/sbin/cron
root      4283  0.0  0.1   1640   508 pts/0    R+   01:43   0:00 grep cron
```

If pinger2 is running (e.g. soon after the 30 mins interval in which it is scheduled) you might see something like:

```
voyage:~# ps ux | grep pinger2
root      5613  0.0  0.4   2400  1060 ?        Ss   03:00   0:00 /bin/sh -c cd /usr/local/share/pinger; \
perl /usr/local/share/pinger/pinger2.pl > /usr/local/share/pinger/pingerCronStat.stdout 2> \
/usr/local/share/pinger/pingerCronStat.stderr
root      5614  0.2  2.5   9048  6604 ?        S    03:00   0:02 perl /usr/local/share/pinger/pinger2.pl
root      6091  0.0  2.1   9048  5408 ?        S    03:16   0:00 perl /usr/local/share/pinger/pinger2.pl
root      6105  0.0  2.1   9048  5408 ?        S    03:16   0:00 perl /usr/local/share/pinger/pinger2.pl
root      6157  0.0  0.1   1640   512 pts/1    R+   03:17   0:00 grep pinger2
```

We verified that the host recovered after a power cycle.

- It pinged so the host was up,
- [http://140.105.28.27/cgi-bin/ping\\_data.pl](http://140.105.28.27/cgi-bin/ping_data.pl) responds so Apache works.
- Submitting the request fetched appropriate data (i.e the latest was current within 30 mins, note the time-stamps are in Unix epoch time and are GMT).

## Configuration

### Logging onto the Host

Give yourself an IP in the same network as the ePinger machines. For example, you can give yourself 140.105.28.30 with netmask 255.255.255.128. You can then connect with any network cable to the Alix board and follow the instructions you already have. Of course, keep in mind that if you change the IP of the eth port on the right (as seen from the back) then you need to be on the same network to communicate with it.

### Ethernet

Here is how to setup the network interface on the ALIX board:

- login as root
- remountnw
- cd /etc/network/
- vi interfaces
- just copy what is already there for eth0 and change the IPs.
- /etc/init.d/networking restart (this will restart the networking services)
- ifconfig (to check if everything is OK).

That should be it.

Keep in mind that eth0 is the plug on the right (there is sticker too), while eth1 is the one on the left.

After reboot, it takes some time before you are able to ssh. Don't panic (I did).

When you setup eth1, you should disable eth0. In this way it answers to ssh much faster. And at then end, they don't use both eth ports. On the other side, it's a good idea to have both for local troubleshooting.

### Apache Logs

The Apache log rotation should be set up. This section just explains it a bit more:

1) check apache logs files:

```
ls /etc/log/apache/*.log
```

2) change directory:

```
cd /etc/logrotate.d
```

3) edit file: "apache" and change first line: /etc/log/apache/\*.log (default is /var/log/apache/\*.log)

4) run: logrotate -f /etc/logrotate.conf

5) check that logs have rotated:

ls /etc/log/apache/.