Last tweaked December 4, 2014 22:42 Mountain

BACKUP the file before you modify it!

Cron Log Location

/var/log/cron
/var/log/cron.1
etc.

Back to Menu Back to Last

Crontab What's Active

To see what crons are actually active in ram,

As root:
crontab -l

Back to Menu Back to Last

Cron Tips

While troubleshooting a cron that just wouldn't work for me, someone suggested a simple trick.

"One trick I use is to replace the time variables with * * * * * and kick the bugger in the butt and see if it runs every minute (it should)."

Once you get it working, change the time variables back to what you want them to be. It's such a simple thing to do, but I didn't think of it when I was fighting a stubborn problem.

Should you need to restart crons, try the following:

As root:
/etc/rc.d/init.d/crond restart

To be sure custom.crons is loaded:

As repeater:
update files

Where are the cron logs? Try /var/log/cron

BACK to Menu Back to Last

Custom.crons

When you want to schedule tasks on your node's computer, you should create a file in the /custom directory called custom.crons. Unless you've already done this, there won't be one there. I'll use the same example that I used to illustrate how you might schedule your node to be disabled from midnight until 0600 automatically. That section is elsewhere in these pages.

As repeater:
pico -w custom/custom.crons

In the custom.crons file, you'll enter what looks like a bunch of gibberish to tell the system what you want done and when.

You might have two lines that look like:
0 0 * * * (/home/irlp/scripts/disable &>/dev/null 2>&1)
0 6 * * * (/home/irlp/scripts/enable &>/dev/null 2>&1)

Note that there are spaces between the asterisks above. Each one is a place holder for a scheduling parameter.

From left to right, the parameters are:
1-minute
2-hour
3-day of the month
4-month of the year
5-weekday

The asterisk tells the cron to run the job EVERY instance of whatever slot it is in. If there is an asterisk in the month parameter, it will run every month. A special symbol is used in the minute parameter to tell it to run every X minutes. For instance, */10 would say to run every 10 minutes. The number should divide evenly into 60, or you could get an odd timing at the top of the hour. Further, multiple timings for a single parameter may be specified by separating them with a comma,

such as:
3 23 * * 3,6,7 (/home/irlp/scripts/disable &>dev/null 2>&1)

The above line will disable the node at 2303 only on Wednesday, Saturday and Sunday nights. Monday is 1, Tuesday is 2, etc. Sunday can be 0 or 7.

The hyphen is legal too, meaning:
3 23 * * 4-7 (/home/irlp/scripts/disable &>dev/null 2>&1)
would do the same function on Thursday through Sunday.

Combining is allowed.
(3 23 * * 1-3,5-7 for Monday through Wednesday and Friday through Sunday)

After putting what you want in the custom.crons file with pico, write the file to disk with Control-O, confirming the filename with a return, and then Control-X to exit back to the command prompt. Make sure permissions are correct with:
chmod 750 custom/custom.crons

After creating or changing the custom.crons file, you'll need to reload the custom.crons to import that new data into the main crons list.

As repeater (node must not be connected to another at the time):
update files

Check to see that the new lines are included and correct in the crons with:
crontab -l

BACK to Menu Back to Last

Custom.crons Doesn't Work After OS Update

This also works in CentOS (I just did it)

Even after reloading custom.crons with update files, some upgraded installations aren't running the crons. They even show up with the crontab -l command. They just don't work. I confirmed this problem with the last 2 installs I did and applied the one time fix below. As others have reported, that fixed it. Reference thread attached to this message on the IRLP Yahoo group, in particular msg 28892 from VE7LTD.

As repeater:
ln -s /home/irlp/.bash_profile /home/irlp/.profile

This is a one time fix, so you shouldn't have to do it again. You may want to reload custom.crons (not sure it is necessary for this fix, but it won't hurt).

As repeater:
update files

Remember, you must not be connected when you update files.

Very similarly, from WØANM, who had helped get some non-working crons going for someone:

...found that the "BASH_ENV=/home/irlp/.profile" was set. On his system, the "home/irlp/.profile" file does not exist. I had him copy the "/home/irlp/bash_profile" to "/home/irlp/.profile"; a symlink could accomplish the same thing.

Bottom line is that the irlp environment was not being imported correctly.

Chris

BACK to Menu Back to Last

Custom.crons Manual (Re)load

Reloading custom.crons can only be done while the node is DISCONNECTED!

After making a change to custom.crons,

As repeater:
/home/irlp/scripts/update files

Or as repeater:
./scripts/update files

Notice the period at the beginning of that line.

Or as repeater:
update files

All three methods work on my node.
After updating, you can check to see that the changes are loaded with:
crontab -l

This action is automatically done daily during the 0300 automatic updates.

BACK to Menu Back to Last

Custom.crons Automatic Connect

Want to connect somewhere for a net on a regular basis?

If you want to connect to a certain reflector or node on a regular basis (daily, weekly, monthly, etc.) at a certain time, all you need to do is put it in your custom.cron file. Let's say there is a net on reflector 9250 at 2100 your time every Sunday night. Let's say you also want to disconnect at 2200 local time. You'll want to connect a few minutes beforehand.

```
#--Early connect to 9250 for Sunday night net at 2055 and
disconnect at 2200
55 20 * * 0 (/home/irlp/scripts/connect_to_reflector ref9250
&>/dev/null 2>&1)
00 22 * * 0 (/home/irlp/scripts/end > /dev/null 2>&1)
```

The first line is a rem statement (#) to explain the lines below it.

The second line connects to 9250 at 2055 (5 min early for the net).

The third line disconnects at 2200. Note, this is an abrupt disconnect. It will happen whether the node is in use or not, whether someone is transmitting either dirction or not. You could create and call your own script for the disconnect and build in some protection in case the node was in use at the time.

The above is merely an example of what you can do. There are lots of possibilities.

BACK to Menu Back to Last