

# **hcron**

## **A Periodic Command Scheduler**

Dorval - 2009-10-29  
John Marshall, HPCS

# hcron

- An easier way to manage scheduled command execution, today!
- What you didn't know you were missing with `cron`

2 / 34

# What Is Right with crontab

- Allows one to execute commands at specific times
- Simple to use
- Text-based
- Ubiquitous

3 / 34

# What Is Wrong with crontab

- Difficult to manage
  - Everything is in one file
  - Minimal/unhelpful structure/organization for humans
- Difficult to edit
  - Table format
  - Not extensible
  - Often requires human, rather than automated, processing

# hcron Principles

- The filesystem as work area and organizing tool
- One event per file
- Attribute-oriented
- Network-oriented
- Extensible
- Automation-friendly

# hcron Configuration (User Perspective)

- User restriction
- Localhost restriction
- Command spawn timeout
- Limit of events per user

6 / 34

# .hcrn Hierarchy

- Everything lives at `~/ .hcrn` (unless otherwise indicated)
- Segregated by fully qualified hostname: `~/ .hcrn/<fqHostName>`
- Tree provides/encourages "meaningful" structure
  - Directories are containers
  - Files are event definitions
  - Names tag/identify

## .hcron Hierarchy (2)

```
~/  
  .hcron/  
    <fqHostName>/  
      events/  
        machine_checks/  
          uno  
          duo  
          trio  
        stats_updates/  
          personal/  
            bobby  
            mary  
          group/  
            red  
            green
```



# Event Definition

- Corresponds to `crontab` line
- Plain text file
- Line-oriented
- Blank lines and comments (starting with `#`) ignored
- `name=value` format
- Created using `hcron-event`

# Event Definition Template

Required attributes:

```
as_user=  
host=  
command=  
notify_email=  
notify_message=  
when_month=  
when_day=  
when_hour=  
when_minute=  
when_dow=
```

# Event Definition Example

Purpose: Append "hello world" to a file, every 10 minutes, and send an email notification.

```
as_user=  
host=mymachine.xyz  
command=echo "hello world" >> /tmp/hello  
notify_email=myself@xyz  
notify_msg=Message sent!  
when_month=*  
when_day=*  
when_hour=*  
when_minute=0,10,20,30,40,50  
when_dow=*
```

# Commands

- `hcron-conv` - convert crontab to hcron event files
- `hcron-event` - create and edit hcron event definition files
- `hcron-info` - prints hcron related information
- `hcron-reload` - signal `hcron-scheduler` to reload user hcron event definition files
- `hcron-scheduler` - server which schedules and executes hcron events (admin only)

# Commands - hcron-conv

- Simple first step for straight conversion
- Not necessarily lossless
- Comments are kept with next command
- Convert from crontab

```
hcron-conv --to-events [options] <hostName> <crontabPath>  
<dirPath>
```

- Convert to crontab

```
hcron-conv --to-crontab [options] <crontabPath> <dirPath>
```

# Commands - hcron-conv Example

crontab line:

```
* * * * * ssh hype.cc.ec.gc.ca echo "hello"
```

Event file:

```
as_user=  
host=hype.cmc.ec.gc.ca  
command=echo "hello"  
notify_email=  
notify_message=  
when_month=*  
when_day=*  
when_hour=*  
when_minute=*  
when_dow=*  
template_name=
```

# Commands - hcron-event

- Provides template for new event files
- Invokes editor (uses `EDITOR` if available)
- Non-interactive create (`-c`)
- Queries for reload by default
- Reload may be forced (`-y`) or bypassed (`-n`)

# Commands - hcron-reload

- Reload **required** to signal hcron-scheduler
  - Occurs at minute interval
- All event files reloaded
- Event list status updated (see `hcron-info -es`)
- Useful when updating multiple event files or automated use



# Commands - hcron-info

- Check if you are allowed by hcron (`--allowed`)
- Check fully qualified hostname (`--fqdn`)
- Print status of each event file loaded (`-es`)
  - Useful for debugging

17 / 34

# Commands - hcron-info - Event Lists

- User-specific
- **Format:** `<status>:<reason>:<eventPath>`
  - `<status>` is "accepted" or "rejected"
  - `<reason>` is possible reason for being rejected
  - `<eventPath>` is the full path of the event file

# Commands - hcron-info - Event Lists Example

Example:

```
rejected:bad definition:bad_event
rejected:bad definition:chain0/a
accepted::chain0/b
accepted::chain0/c
accepted::chain0/failed
rejected:bad definition:mail_on_the_even_minutes
rejected:bad definition:mail_on_the_minute
rejected:bad definition:mail_on_the_minute_at_19h
rejected:bad definition:mail_on_the_three_minutes
rejected:bad definition:say_hello
```

# Log File - hcron.log

- `/var/log/hcron/hcron.log`
- Provides tracking
  - **All** events are logged
  - Timestamped
  - Userstamped
  - Processing time
  - Typed: message, load-events, start, work, sleep, execute, discard-events, notify-email, chain-events

# Advanced Topics

★ Not available with cron

- One-time events
- Event chaining
- Variable substitution
- Template events

21 / 34

# One-time Events

- `when_year` attribute
  - Not **required** in event definition
  - Default/implied: `when_year=*`
  - Range: 2000-2050
- Use cases:
  - Notification
  - Automated event creation for non-recurring operations

# Event Chaining

- Works according to event "execution" results
  - Success: command executes
  - Failure: command fails to execute
    - E.g., Downed host, password/passphrase required
- Multiple events can be chained together using `next_event`
  - `a -> b -> c ...`
- Failover/fallback events can be activated using `failover_event`
  - `a -> alt_b -> c`

# Event Chaining Example

Example:

```
as_user=  
host=hype.cmc.ec.gc.ca  
command=echo "$HCRON_EVENT_NAME" >> ~/chain_test  
notify_email=John.Marshall@ec.gc.ca  
notify_message=  
when_month=*  
when_day=*  
when_hour=*  
when_minute=*  
when_dow=*  
next_event=step2  
failover_event=failed
```

- if success execute event `step2`
- if failure execute event `failed`



# Variable Substitution

- **Not** shell variables
- `$HCRON_HOST_NAME` - the host name of the host executing hcron
- `$HCRON_EVENT_CHAIN[<index>]` - `:-`-separated list of event names in the chain (`>=1`)
- `$HCRON_EVENT_NAME` - the full event name which consists of the path following the `.../events`
- `$HCRON_EVENT_NAME[<index>]` - a component of `$HCRON_EVENT_NAME`
- Indexing:
  - Positive **and** negative index values are allowed
  - Index of 0 is the first element
  - Index of -1 is the last element

# Variable Substitution Example

Purpose: Birthday reminder.

Event file:

```
as_user=  
host=mymachine.xyz  
command=date  
notify_email=bob@xyz  
notify_message=${HCRON_EVENT_NAME[-1]} was born on  
${HCRON_EVENT_NAME[-4]}-${HCRON_EVENT_NAME[-3]}-${HCRON_EVENT_NAME[-2]}!  
when_month=${HCRON_EVENT_NAME[-3]}  
when_day=${HCRON_EVENT_NAME[-2]}  
when_hour=9  
when_minute=0  
when_dow=*
```

# Variable Substitution Example (2)

Tree (paths condensed):

```
~/ .hcron/hcron1.cmc.ec.gc.ca/events/  
  birthdays/  
    US_Presidents/  
      1732/02/22/George_Washington  
    stooges/  
      1987/06/19/Moe_Howard  
      1902/10/05/Larry_Fine  
      1903/01/18/Curly_Howard  
  friends/  
    1978/10/23/marty
```

# Variable Substitution Example (3)

Event file (evaluated):

```
as_user=  
host=mymachine.xyz  
command=date  
notify_email=bob@xyz  
notify_message=marty was born on 1978-10-23!  
when_month=10  
when_day=23  
when_hour=9  
when_minute=0  
when_dow=*
```

# Template Event Definition

- Allowed in the `~/ .hcr on` tree
- Marked by:

```
template_name=
```

- Ignored by hcr on scheduler
- Especially useful when used along with variable substitution
- Symlinked to by other event files

# Template Event Definition Example

Purpose: Monitor one or more log files, at 30 minute intervals.

Template event file:

```
as_user=  
host=mymachine.xyz  
command=. ${HOME}/.profile; ls -lad $HCRON_EVENT_NAME[-1] >>  
~/log_status  
notify_email=  
notify_message=  
when_month=*  
when_day=*  
when_hour=*  
when_minute=0,30  
when_dow=*  
template_name=template_monitor_log_status
```

# Template Event Definition Example (2)

Tree:

```
~/.hcron/hcron1.cmc.ec.gc.ca/events/  
  monitoring/  
    template_monitor_log_status  
      syslog -> template_monitor_log_status  
      message -> template_monitor_log_status  
      auth -> template_monitor_log_status
```

# Template Event Definition Example (3)

Template event file (evaluated for syslog):

```
as_user=  
host=mymachine.xyz  
command=. ${HOME}/.profile; ls -lad syslog >> ~/log_status  
notify_email=  
notify_message=  
when_month=*  
when_day=*  
when_hour=*  
when_minute=0,30  
when_dow=*  
template_name=template_monitor_log_status
```



# Template Definition Example (Birthdays)

★ One template, multiple symlinks.

Tree (paths condensed):

```
~/ .hcron/hcron1.cmc.ec.gc.ca/events/  
  birthdays/  
    template_birthday  
    US_Presidents/  
      1732/02/22/George_Washington -> ../../..  
/../../template_birthday  
  stooges/  
    1987/06/19/Moe_Howard -> ../../../../../../template_birthday  
    1902/10/05/Larry_Fine -> ../../../../../../template_birthday  
    1903/01/18/Curly_Howard -> ../../../../../../template_birthday  
  friends/  
    1978/10/23/marty -> ../../../../../../template_birthday
```

# Servers and Support

- hcron Servers
  - cron-dev1.cmc.ec.gc.ca - running v0.8
  - hcron1.cmc.ec.gc.ca - running v0.11 (with advanced topics features)
- Support
  - Email: [✉ service.hpcs@ec.gc.ca](mailto:service.hpcs@ec.gc.ca)
  - Questions and feedback welcome
  - ⚠ If you discover you are not "allowed", please tell your user representative.