

Upstart Roadmap

Linux Plumbers 2013

James Hunt <james.hunt@ubuntu.com>

and

Dmitrijs Ledkovs <dmitrijs.ledkovs@ubuntu.com>

September, 2013



Table of Contents

1. Introduction
 2. `ptrace`
 3. cgroups Support
 4. States
 5. `upstart-time-bridge`
 6. Possible Future Features
 7. New Features
 8. Contact
 9. Links
- Platform Support
Other Event Sources
Namespaces



Introduction

- Current thoughts and ideas.
- In no particular order.
- Don't hold us to what follows! :-)
- If you'd like to get involved, please let us know!



ptrace

- Upstart uses `ptrace(2)` for cross-platform PID tracking and "service readiness".
- Current design works extremely well but small number of daemons don't conform to expected behaviour.
- Plans to refine current design to account for these:
 - Track calls to `exit(3)` as well as `fork(2)`.
 - Track calls for *all* Job processes.



cgroups Support

- Intention to add a **cgroup** stanza.
- *Purely* for job resource management (cpu, memory, disk, *et cetera*)
- NOP for platforms that do not provide cgroups.



States

- Add support for "states".
- Solves problem of events not re-occurring:

```
description "foo job - runs when bar and baz are running"  
start on started bar and started baz  
stop on stopping bar or stopping baz  
exec mydaemon
```

What happens if **baz** stops then re-starts?

- Would allow jobs to specify something akin to the psuedo-code:

```
while bar and baz  
exec mydaemon
```

Would ensure that if **baz** stopped (causing **foo** to stop) and then **baz** restarted that the **foo** would also restart.

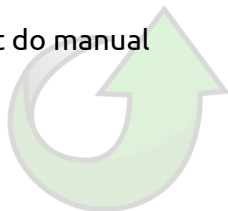
Plan to introduce a new **upstart-time-bridge**.

- Provide **cron(8)**-like system.
- To run jobs periodically (up to some maximum number of times?)
- To run jobs delayed *after* some other events:
 - *time* after system startup.
 - *time* after all filesystems mounted.
 - *time* after a job started/stopped.



Platform Support

- Alternative kernels (FreeBSD/kFreeBSD/Hurd)?
 - Upstart/**libnih** strive to encapsulate & abstract Linux-specific behaviour.
 - **libnih** uses Linux specific API: inotify.
 - Does use **/proc** in limited fashion though:
 - OOM.
 - Chroot support.
- Porting to FreeBSD: use **kevents** or compat wrappers, e.g. **libkevent-inotify**.
- Porting to Hurd: no file notification API. Must do manual calls to **initctl reload-configuration**



Other Event Sources

- Transport-layer (specifically `listen(2)`) event source would be extremely useful.
- Plan to add simple NM `initctl` script for Ubuntu so Session Jobs (non-priv) can react when changing wifi network for example.
- Could even add in suspend/resume events (although note that jobs using those could block either operation).



Namespaces

- Could introduce support for running processes in separate namespaces (network, user, pid, etc).
- Using namespacing simply to disable a particular namespace can already be accomplished with current Upstart Apparmor support though.

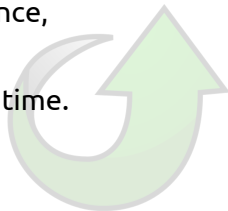


New Features

New Upstart features are added relatively slowly since each new feature:

- Must provide a clean design.
- Must provide a generic and generally useful facility.
- Needs all code to pass clean through all appropriate analysis tools.
- Must be fully documented.
- Must provide a full set of tests.
- Must not negatively impact overall performance, reliability or safety of the system.

That said, we are working on new features all the time.
What features would you like to see in Upstart?



Contact

- Email addresses on first slide.
- IRC channel
#upstart on freenode.net
- Mailing List
<https://lists.ubuntu.com/mailman/listinfo/upstart-devel>
- Raise a bug
<https://bugs.launchpad.net/upstart/+filebug>



Links

- Upstart Home
<http://upstart.ubuntu.com>
- Upstart Cookbook (HTML and PDF)
<http://upstart.ubuntu.com/cookbook/>
- Upstart Code
<http://launchpad.net/upstart>
- "The Bible"
`man 5 init`
- Ideas page
<https://wiki.ubuntu.com/Upstart/Ideas>



THE END.

