



Contents

1. [OpenRC position statement](#)
 1. [Pros](#)
 2. [Cons](#)
2. [Comparing OpenRC with others](#)
 1. [OpenRC vs Systemd](#)
 2. [OpenRC vs Upstart](#)
 3. [Rebuttals](#)
3. [A bit of OpenRC description](#)
 1. [Why this?](#)
 2. [Things you need to know about OpenRC](#)
 3. [How OpenRC works: Example runscript](#)
4. [Transition plan](#)
5. [Todo list for OpenRC](#)
 1. [invoke-rc.d](#)
 2. [Hurd port](#)
 3. [sysv-rc depends](#)
6. [Position statement maintainers](#)

Before editing this page, please see the instructions on how debates work. If you are not the maintainer of a position statement, and have a suggestion or change to make, please contact the maintainers.

OpenRC position statement

OpenRC wishes to continue implementing things in a small, simple and efficient way, with the minimum things written in C, and all of the needed features. It also aims at portability. OpenRC is used in Gentoo by default, but also works very well in a number of BSD systems, and it can already work under Debian GNU/kFreeBSD very easily. OpenRC doesn't "take over" the PID 1, it doesn't even implement it, it only implements what's after it: it is a replacement for sysv-rc, not for sysv-init.

Pros

- OpenRC is straightforward: we don't reinvent the wheel, and just keep what we used to have. `update-rc.d`, `invoke-rc.d`, etc just "work" as expected. OpenRC continues to use `start-stop-daemon`, which it enhanced. Though if we want to switch to the newer interface of OpenRC (`rc-update`, etc.) it is possible as well.
- The runscript file format allow both a completely declarative (something like 7 lines should be enough in most cases), but it also supports the old `sysv-rc` format, with the addition for the support of the LSB headers which has been added last summer by the GSoC student. So the transition would be extremely easy: just switch to OpenRC, then slowly move away from `sysv-rc` style. Nothing to force on anyone.
- OpenRC is a very good compromise between features and complexity of upstream code. As few as possible code is written in C (this is an upstream choice), the rest of is using scripting (Perl, shell...), making it very easy to maintain and debug.
- OpenRC does *not* re-implement `sysvinit`. Yes, this is a good thing. Rewriting the PID 1 process is very dangerous, and adding a lot of code in it is the recipe for disaster. Here, we would continue to reuse what has worked well for decades, and enhance the features later in the boot process.
- OpenRC has a runlevel namespace. Bye bye runlevel 0, 2 and 4, and say hello to runlevels with comprehensive names "shutdown" (instead of 0), "single" (instead of 1), "reboot" (instead of 6) or "default". Note that this is handled in an extremely easy way using `/etc/init.d/rc` (which is specific to Debian, for compatibility reasons). OpenRC doesn't even need `/etc/inittab` to be edited since last September, so we could even keep what we have currently, including any user modified `inittab` files.
- Everything stays as we are used to. No huge revolution needed, like rewriting `getty` and so on for no valid reason (are these tools broken? if so, how???)
- OpenRC uses control groups to ensure that any service, regardless of its state, can be shut down properly.

Though that's not a forced feature, it continues to work on systems who don't have CGroups available.

- OpenRC allows very fast boots. One of upstream author made a proof of concept with booting in 250ms.
- OpenRC supports nested runlevels this mean you can create complicated system layout by creating specialized runlevels: like default/X11, default/nonetwork, default/network
- OpenRC upstream is friendly and open as for contributions and for suggestions for feature and portability, they will try to find consensus for the every edge case. They have already been extremely helpful during this summer for the GSoC, with one of the upstream doing the mentoring work as well. They also helped with the porting to kFreeBSD (it took only a few hours).
- OpenRC supports a subset for event-driven action,e.g. services hotplugging (example: http://wiki.gentoo.org/wiki/Iphone_USB_Tethering#udev_trigger)
- Upstream version of OpenRC provides an interfaces for extending it with additional modules by plug-in system of s-s-d (openrc-plymouth is done in that way)
- OpenRC is designed to do service management and do it well, thus it acts as a central place and glue for other existing systems that can solve subtasks in the best way. i.e. for pid-1 (sysv5,s6..), supervision (s6/monit), logging utilities. So it's possible to construct scalable solutions on the top of it
- OpenRC is ready for contribution and work on openrc-daemon: additional package that can provide additional interfaces for OpenRC (e.g. some sort of a systemd compatibility layer)
- OpenRC provides librc that can be used in the 3rd party tools that want to extend OpenRC
- OpenRC tries to find a consensus between best practices and tools of the target OS and compatibility baseline, that means that core functionality depends only on compatible POSIX interface, and additional functionality (modules) may be OS or environment

specific

- OpenRC tries to reuse existing tools instead of merging them into a single code base. It provides a way to reuse well known unix tools (at/ulimit/etc.).
- OpenRC provides a way to add service specific commands based on service states (seems this should to be in sysv-rc as well)
- OpenRC comes along with netifrc - very flexible framework for network configuration
- OpenRC init scripts are state-full (so you can store additional information while service is alive) but that state is not persistent between service restarts
- Just like Debian, OpenRC is community driven and doesn't have an agenda driven by commercial interests

Cons

- OpenRC doesn't have socket activation yet.
- While some porting efforts has been put on OpenRC, it still needs some more work to get into Debian, namely the rename of /sbin/rc which clashes with the /usr/bin/rc from the "rc" package. Upstream is working on that.
- No effort yet to port it to Hurd.
- Not everything from the plans are ready (i.e. there is no clean supervision integration), however work is done
- No big vendor behind OpenRC, thus there is no paid developers
- Since runscripts will live in /etc/init.d, we would have to completely get rid of sysv-rc in Debian to be able to start writing runscripts and replace the existing traditional sysv-rc scripts. Though that's not a bad thing, since OpenRC should be completely compatible with sysv-rc and that upgrades to it should be painless (at the exception of the first reboot maybe: we still didn't yet find a reliable way to shutdown cleanly all daemons after an upgrade to OpenRC from sysv-rc).

Comparing OpenRC with others

OpenRC vs Systemd

I have decided to not write anything in this section, considering the aggressive tone I'm getting in return, which is all but fun. Anyway, the problems with Systemd have been debated a lot already, so it is useless to list them here again.

OpenRC vs Upstart

- Even if upstream for Upstart would accept patches for it, there is currently no way to run Upstart on the non-Linux ports, and it is clear that it would take a non-negligible effort to achieve it. There is currently nobody working on that either. Edit: it seems this has change, and that's a very good thing!

Rebuttals

- From the Systemd page: "one of the primary improvement of OpenRC over the sysinit is cgroups" - it's wrong, cgroups are just a minor feature that can extend some use-cases, however [🌐 pros on this page](#) describes much more features that come with OpenRC (to not duplicate them here). Systemd authors and supporters are trying to let everyone believe that cgroups is the best thing invented since sliced bread, and that it is very complex, which isn't truth.
- From the Systemd page: "The primary selling point of OpenRC is that it is portable. Otherwise, it is merely a sysvinit on steroids.". Well, again, that's not truth, OpenRC does a lot more than just this, this shows very little understanding and investigation from the author of these words.
- From the Systemd page: "There have been many comments on supposed lack of documentation on systemd's internals or interfaces. 10% of the source code (14 kLoC) are comments. Each component has its detailed manual page." Then we can read suppositions

about the hardness/ease of reimplementing Logind. Well, what has been said by the authors of OpenRC is that re-implementing logind isn't hard, but making a replacement behave exactly like it means that one needs to reverse engineer Logind and read the source code, which shouldn't be the case. This has nothing to do with the (useless) hard dependency on Systemd that has been added on version 205 of Systemd.

- From the Systemd page: "Many discussions are centered on systemd-logind as in being the only problem to address by other proposals, wildly proposing seemingly easy-to-develop replacements." Well yes, because it seems that it's being forced to everyone (for example by Gnome), and it imposes systemd starting at version 205. So having a way to replace it is important, otherwise, we are effectively locked-in both systemd and logind. Then on the same page, we can read: "It is hard because without systemd, you are missing the necessary features to implement this service."
- The systemd page uses Linux as being monolithic as an argument, which has absolutely nothing to do with the discussion about too many things in the PID1 (it is by the way my opinion that Linux is a **way** too monolithic though).
- From the systemd page: "Many are complaining about the absence of shell scripts, quoting ease of administration or embedded systems. Shell scripts are not easy to maintain or debug; systemd unit files are." Probably. Though Upstart jobs and OpenRC runfiles are equally easy to maintain, and that's not what has been discussed: others wrote that **systemd** is hard to maintain, and debugging a system which refuses to boot with systemd isn't as easy as it use to be systemd. Anyone who pretends something else is just lying.
- Overall, the systemd page is using very aggressive tone like: "boil down to rumors or irrelevant political stances". Saying that your opponent is just "spreading FUD" or that all arguments just "boil down to rumors"

isn't helping the debate. Plus highlighting that the systemd author clearly stated that he would refuse any non-linux port is very relevant, and **is** about a policy from the upstream authors.

- From the sysvinit page: "Was only really brought up as a counter-argument for a switch to Systemd, there may not have been such motivation to switch to this from sysvinit otherwise." This is untruth. My interest for OpenRC came after discussing (face to face) with one of the OpenRC upstream maintainers (Patrick Lauer also lives in Shanghai), and after searching for ways to more or less do what OpenRC is capable of (I even went to start writing my own lib, then scrapped that project), with as motivation to offer an interface for simplifying existing init scripts in Debian packages without changing every components of the system.
- From the sysvinit page: "No objections seen yet [note: to OpenRC] in discussions, but this init system is the least familiar to us yet". Well, to a degree, this is the init system with which everyone is already familiar with, since it doesn't change any of the interfaces we are already used to: update-rc.d and invoke-rc.d have been re-implemented so that they call rc-update transparently, runscripts live at the exact same place were init scripts use to live and just need to replace them. Read what's below (the example runscript and so on) and you will have to agree! 😊

A bit of OpenRC description

Why this?

Because, even if OpenRC is used more or less the same way as sysv-rc, is the least known init systems, so I think this is useful.

Things you need to know about OpenRC

- OpenRC **used to** require a modified /etc/inittab

(renaming the runlevel numbers with names like "shutdown" instead of "0", "single" instead of "1", etc). That isn't the case anymore since September, you can keep your unmodified inittab file. A convenient `/etc/init.d/rc` init script now makes the translations between numbers to the namespaced runlevel system of OpenRC.

- OpenRC now supports LSB header in init scripts, and calculates dependencies using it. Both LSB headers and runscripts dependencies can coexist nicely.

How OpenRC works: Example runscript

Because it seems OpenRC is not known enough, I thought it'd be nice to explain how it works, and show what a runscript looks like. So I wrote the below runscript for rsyslog replacing the default init.d sysv-rc style script shipped with the current version of rsyslog:

```
#!/sbin/runscript

command=/usr/sbin/rsyslogd
name="enhanced syslogd"

depend()
{
    provide rsyslogd syslog
    need $remote_fs $time
}
```

The only thing that is needed to replace the old init.d script of rsyslog is replace its content by the above, then do:

```
update-rc.d rsyslog defaults
```


It is just like we have been used to for years, and that is it, nothing more to it: it auto-magically integrates with the rest of the (traditional) init scripts with correct dependencies as one would expect. Internally of course, OpenRC in Debian re-implements update-rc.d and calls rc-update, but nobody needs to know that (except the maintainers of OpenRC in Debian). Pretty strait forward, for anyone that is used to the old sysv-rc scripts, isn't it? Also, if the above runscript looks a bit like a shell script, it is because it is one! In fact, maintainers can decide to implement or not the start() and stop() functions, and do any complex thing **if** they decide to do it (it is of course recommended to avoid it, but sometimes, you really do need to do complex stuff).

Transition plan

The transition should be very easy if we decide to get rid of the old sysv-rc package. Just have sysv-rc become a transition package which depends on OpenRC, and let everyone upgrade.

Todo list for OpenRC

The Debian package has still some things to be fixed. Rather than hiding the truth, I'm listing them here, hoping that this transparency will be appreciated.

invoke-rc.d

The Debian package for OpenRC keeps things as they were with sysv-rc as much as possible. Note that it is this way because of efforts from the package maintainers to allow a smooth trandition. The OpenRC version in Gentoo for example, doesn't provide update-rc.d or invoke-rc.d, which had to be included in OpenRC. These currently need some more work, so that symlinks in /run/openrc/started are correctly added or removed when calling invoke-rc.d in the case of a shell script (as opposed to a runscript which

already works perfectly). This will make it possible to have "rc-status" display things correctly in all cases, which isn't the case yet when writing these lines (it currently only works for runscripts).

Hurd port

The current status of compatibility with Hurd is currently unknown. Some time should be spent to at least know what work should be done for the porting work.

sysv-rc depends

It used to be very easy to install OpenRC: `apt-get / dpkg -i` the package, and that's it. Though recently, some dependency change in the sysvinit source package made the situation more complicated: `ifupdown` and `openssh-server` depends on a version of `sysv-rc`, which makes it impossible to install OpenRC even if it Provides: `sysv-rc`. A solution should be found.

Position statement maintainers

- Thomas Goirand (package maintainer) with help from upstream authors below
- Alexander Vershilov
- Patrick Lauer