Thanks for this clear explanation.

Reading this article trigger some questions:

- Do I need that stuff?
- In other words, what do I miss if I don't have it?
- For what purpose(s) could I use it?
- Does it replace some components and/or provide new or better features than existing components, then which ones?

Maybe giving a few practical use cases and examples of utilization could help answering these questions.

— *Didier Spaier 2015/05/15 10:06*

Hi Didier,

Sorry, I did not notice your questions before.

For me, the primary reason for using OpenRC is that it provides standardization. Yes, it does offer some superior features, like dependency handling of services, cgroups support, service supervision, etc.

It does not replace any components that I know of, sysvinit is used for booting and then handles control to OpenRC, and the services provided by Slackware in /etc/rc.d can also be used.

It can also provide a faster boot experience and supports parallel starting of services as well.

— *Aaditya Bagga 2015/05/18 10:50*

Migrating to OpenRC 0.39+ from previous versions

OpenRC 0.39.2, as ***planned*** to be shipped via SBo, features 2 main changes:

- *openrc-init* is being used as init instead of */sbin/init*
- Service and configuration directory has been changed to */etc/openrc* instead of */etc*

Both these changes were done to have a clean{er} OpenRC install which can co-exist with other init systems like sysvinit (default), runit, etc, just by changing boot parameters.

Following articles have more info:

- To switch to openrc-init
- To switch to /etc/openrc/init.d

dummy heading 1

added for generating toc

dummy heading 2

added for generating toc

dummy heading 3

added for generating toc

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/talk:howtos:general_admin:openrc

Last update: **2018/11/07 07:36 (UTC)**