2024/03/18 04:13 (UTC) 1/8 OpenRC

OpenRC

OpenRC is a dependency based init system.

Features

OpenRC provides a number of features like hardware initiated initscript run and cgroups support, without requiring large layout changes.

Installation

Two Slackbuilds are available, openrc, which contains the OpenRC init system, and openrc-services, which contains various services for use with OpenRC.

Post installation

After installation, some steps are needed to boot with OpenRC.

Setup the agetty services

```
# main tty
ln -s /etc/openrc/init.d/agetty /etc/openrc/init.d/agetty.tty1
/sbin/rc-update add agetty.tty1 default
cp /etc/openrc/conf.d/agetty /etc/openrc/conf.d/agetty.tty1
echo 'agetty_options="--noclear"' >> /etc/openrc/conf.d/agetty.tty1

# additional ttys
for i in {2..6}; do
    ln -s /etc/openrc/init.d/agetty /etc/openrc/init.d/agetty.tty${i}
    /sbin/rc-update add agetty.tty${i} default
done

# serial tty (for servers)
ln -s /etc/openrc/init.d/agetty /etc/openrc/init.d/agetty.tty50
/sbin/rc-update add agetty.tty50 default

cp /etc/openrc/conf.d/agetty /etc/openrc/conf.d/agetty.tty50
echo 'agetty_options="--noclear"' >> /etc/openrc/conf.d/agetty.tty50
```

Update boot parameters

Add the following to your boot parameters (via /etc/lilo.conf for lilo or /etc/default/grub for grub):

```
init=/sbin/openrc-init
```

Regenerate boot configuration ('lilo -v' or 'grub-mkconfig -o /boot/grub/grub.cfg').

Enable boot logging

The main configuration file for OpenRC is /etc/openrc/rc.conf, and contains various options.

A common option that could be changed is to enable boot logging by setting rc_logger="YES", this way any errors encountered via booting could be logged and examined later (default log location is \(\frac{\var}{\log} \) \(

On reboot

On rebooting, one is booted to a command line with only the bare minimum of services enabled.

It may show a warning about deprecated support for /etc/mtab as a file, and how to correct it:

```
# cp /etc/mtab /etc/mtab.bkp
# ln -snf /proc/self/mounts /etc/mtab
```

It may also complain about missing /etc/sysctl.conf file which can be created as:

```
# touch /etc/sysctl.conf
```

Enabling some services

Services can be enabled as:

```
# rc-service add <service> <runlevel>
```

They can be disabled as:

```
# rc-service del <service> <runlevel>
```

Some common services that could be enabled:

```
# rc-update add dbus default
# rc-update add sysklogd default
# rc-update add dcron default
# rc-update add alsasound boot # for desktop
```

2024/03/18 04:13 (UTC) 3/8 OpenRC

```
# rc-update add consolekit default # for desktop
# rc-update add sshd default # for server
```

Enabled services in current runlevel can be gueried with:

```
$ rc-status
```

Enabled services across all runlevels can be queried with:

```
$ rc-update
```

Some services that are enabled by default for the **sysinit** runlevel include udev and udev-postmount.

Users who use dmcrypt, lvm, or mdraid could add the following services respectively to the **boot** runlevel:

```
# rc-update add dmcrypt boot  # dmcrypt
# rc-update add device-mapper boot # lvm
# rc-update add lvm boot  # lvm
# rc-update add mdraid boot  # mdraid
```

Network

If using ethernet via DHCP, the **dhcpcd** service could be enabled:

```
# rc-update add dhcpcd default
```

If using wifi with laptops the **NetworkManager** or **wicd** service could be enabled. The combination of dhcpcd and wpa_supplicant (along with a frontend like wpa_gui or wpa_cli) could also be used as a lightweight alternative.

A static network can be configured by editing /etc/openrc/conf.d/network.

Display manager

To boot to a graphical display manager, /etc/openrc/conf.d/xdm can be edited to specify the display manager, and the xdm service could be enabled:

```
# rc-update add xdm default
```

If using a laptop, the **acpid** service could be enabled as well.

Migrating existing enabled services

To check existing enabled services, following code could be used:

```
for file in /etc/rc.d/*; do
  if [ -x "${file}" ]; then
    echo "enabled ${file}"
  fi
done
```

For these services corresponding OpenRC services could be enabled, for example:

```
# rc-update add acpid default
# rc-update add cgmanager default
# rc-update add gpm default
...
```

All services present can be gueried via:

```
$ rc-update -v
```

Configuration

OpenRC services are present in the /etc/openrc/init.d folder, and corresponding configuration files are present in /etc/openrc/conf.d

Some common configuration files include:

```
/etc/openrc/conf.d/modules # modules to be loaded at boot
/etc/openrc/conf.d/hostname # hostname of the system
/etc/openrc/conf.d/keymaps # console keymap
```

Service management

Services can be started/stopped/restarted as:

```
# rc-service <service> <action>
```

For example,

```
# rc-service sshd start
```

Shutdown/reboot

To shutdown or reboot, one will need to use **openrc-shutdown**. To shutdown from a desktop environment using ConsoleKit2, check out this PR.

2024/03/18 04:13 (UTC) 5/8 OpenRC

Note

The **ck-system-stop** and **ck-system-restart** files in the PR can be saved to /usr/local/sbin/poweroff and /usr/local/sbin/reboot respectively. This allows using the **poweroff** and **reboot** commands like before.

Troubleshooting

At the moment only a subset of the included services have been tested, so some of them may not run correctly.

If some service does not work, try the system provided one in /etc/rc.d (if available).

mysqld

Try adding the following lines to /etc/my.cnf

```
[mysqld]
user = mysql
basedir = /usr
datadir = /var/lib/mysql
pid-file = /run/mysql/mysql.pid
socket = /var/run/mysql/mysql.sock
```

Dealing with crashed services

Sometimes openrc reports a service's status as "crashed". The process may have died or its pid file disappeared/changed.

If one tries to start a crashed service, `rc-service` reports:

WARNING: <service> has already been started

So one tries to stop it before starting again.

However in some situations, the service does not stop. This leads to a deadlock where one can neither stop the service nor start it. For such cases:

```
# rc-service <service> zap
* Manually resetting <service> to stopped state
```

zap resets the service state, allowing us to start it again.

openrc-init

Since version **0.25**, openrc provides *openrc-init* which can be used to boot the system.

This allows switching between OpenRC and other init systems, for example sysvinit, just by changing boot parameters.

For more info, check out the Gentoo wiki.

Migrating from /etc/init.d to /etc/openrc/init.d

OpenRC 0.39+, as shipped via SBo, changes the service and configuration directory to /etc/openrc instead of /etc.

This is being done to maintain greater compatibility with a vanilla Slackware install.

Existing services will need to be migrated, details below on how it can be done.

1. Install openrc-0.39.2 and openrc-services-20181107

These packages ship the configuration and service files in /etc/openrc.

2. Migrate existing services

This involves 4 steps:

```
# migrate existing services
for service in /etc/init.d/*; do
  svcname=$(basename "$service")
  if [ ! -e "/etc/openrc/init.d/${svcname}" ]; then
    if [ -f "/etc/init.d/${svcname}" ] && grep -q openrc
"/etc/init.d/${svcname}"; then
      cp -v "/etc/init.d/${svcname}" "/etc/openrc/init.d/${svcname}"
    elif [ -L "/etc/init.d/${svcname}" ]; then
      # check if symlink is of a service
      service target=$(readlink -f "$service")
      service target name=$(basename "$service target")
      if [ $(dirname "$service target") = "/etc/init.d" ]; then
        ln -sv "/etc/openrc/init.d/${service target name}"
"/etc/openrc/init.d/${svcname}"
      fi
    fi
  fi
done
# enable existing services
```

2024/03/18 04:13 (UTC) 7/8 OpenRC

```
for runlevel in /etc/runlevels/*; do
  for service in ${runlevel}/*; do
    svcname=$(basename "$service")
    rvlname=$(basename "$runlevel")
    if [ ! -e /etc/openrc/runlevels/${rvlname}/${svcname} ] && [ -e
/etc/openrc/init.d/${svcname} ]; then
      ln -sv /etc/openrc/init.d/${svcname}
/etc/openrc/runlevels/${rvlname}/${svcname}
  done
done
# check config changes
for file in /etc/conf.d/*; do
  filename=$(basename "$file")
  extension="${filename##*.}"
  if [ "$extension" = orig ] || [ "$extension" == new ]; then
    continue # dont need to check extra
  fi
  if [ -e "/etc/conf.d/${filename}" ] && [ -e
"/etc/openrc/conf.d/${filename}" ]; then
    diff -Nupr "/etc/conf.d/${filename}" "/etc/openrc/conf.d/${filename}"
  fi
done
# main config file
diff -Nupr /etc/rc.conf /etc/openrc/rc.conf
# check local.d changes
for file in /etc/local.d/*; do
  filename=$(basename "$file")
  if [ ! -e "/etc/openrc/local.d/${filename}" ]; then
    cp -v "/etc/local.d/${filename}" "/etc/openrc/local.d/${filename}"
  elif [ -e "/etc/local.d/${filename}" ] && [ -e
"/etc/openrc/local.d/${filename}" ]; then
    # show changes
    diff -Nupr "/etc/local.d/${filename}" "/etc/openrc/local.d/${filename}"
  fi
done
```

Config changes found above can be updated by hand or copy pasted in new location.

3. Check and reboot

Verify if rc-status is showing all services (it may show them as stopped).

Reboot and check whether things are working expected, make changes as necessary.

The old files and folders can be renamed (or removed).

```
mv /etc/init.d /etc/init.d-openrc-bkp
```

```
mv /etc/conf.d /etc/conf.d-openrc-bkp
mv /etc/local.d /etc/local.d-openrc-bkp
mv /etc/runlevels /etc/runlevels-openrc-bkp
mv /etc/rc.conf /etc/rc.conf-openrc.bkp
```

Finally, any changes done to the default Slackware system can be reverted:

mv /etc/init.d-bkp /etc/init.d

See Also

Wikipedia

Gentoo Wiki

Github

User guide

LQ thread

openrc-services repo

Sources

* Originally written by Aaditya

howtos, init, author aaditya

From:

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

Permanent link:

https://docs.slackware.com/howtos:general_admin:openrc

Last update: 2021/09/12 22:18 (UTC)

