SysVinit vs Systemd

runlevels vs targets
(/etc/init.d, /etc/rcX.d) vs units

Documento de repaso sobre:

- Transición entre runlevels(SysVinit) e targets(systemd)
- Compatibilidade entre /etc/init.d, /etc/rcX.d de SystVinit e os units en Systemd.

Introdución

Historicamente, os sistemas GNU/Linux usaban **SysVinit** para xestionar o arranque e os servizos, baseado en **runlevels** e scripts localizados en directorios como **`/etc/init.d/`** e **`/etc/rcX.d/`**. Actualmente, a maioría das distribucións, incluíndo Debian, usan **Systemd** para xestionar o arranque e os servizos, substituíndo os antigos runlevels por **targets** e os scripts de `/etc/init.d/` por **units** de `systemd`.

Runlevels vs Targets

No sistema tradicional **SysVinit**, os *runlevels* eran números (0-6) que definían os diferentes estados do sistema. Os runlevels determinaban que servizos se cargaban ou se detiñan ao arrincar ou apagar o sistema. O ficheiro `/etc/inittab` era o que se utilizaba para configurar o runlevel por defecto.
En **Systemd**, o concepto de runlevel foi substituído polos *targets*. Estes targets son unidades máis flexibles que definen o estado do sistema e xestionan a carga de servizos.
Exemplos de targets en Systemd e a súa comparativa cos antigos runlevels:

- **Runlevel 0** (apagado) → `poweroff.target`
- **Runlevel 1** (modo de usuario único) → `rescue.target`
- **Runlevel 2** (multiusuario sen rede) → `multi-user.target`
- **Runlevel 3** (multiusuario con rede) → `multi-user.target`
- **Runlevel 5** (modo gráfico) → `graphical.target`
- **Runlevel 6** (reinicio) → `reboot.target`

Cambios no arranque por defecto

En **Systemd**, o runlevel por defecto non se configura en `/etc/inittab`. Para establecer o **target** por defecto, utilízase o comando:

```
# systemctl set-default <target>
```

Por exemplo, para configurar o sistema para arrancar en modo gráfico por defecto, usarías:

```
# systemctl set-default graphical.target
```

Comandos útiles en Systemd relacionados cos targets

- **Listar todos os targets:**

  ```
  # systemctl list-units --type target
  ```

- **Cambiar ao target desexado:**

  ```
  # systemctl isolate graphical.target
  ```

- **Cambiar ao modo de recuperación:**

  ```
  # systemctl isolate rescue.target
  ```

- **Verificar o target actual:**

  ```
  # systemctl get-default
  ```

Resumo

En **Systemd**, os runlevels foron substituídos polos **targets**, e a configuración do sistema e dos niveis de execución agora realízase a través de `systemctl`, sen a necesidade de editar o ficheiro `/etc/inittab` como se facía antes.

/etc/init.d e /etc/rcX.d vs Units en Systemd

Nos sistemas **SysVinit**, os scripts de servizo estaban localizados en `/etc/init.d/`. Estes scripts xestionaban a iniciación e detención dos servizos. Os directorios `/etc/rcX.d/` eran os que controlaban a execución dos servizos segundo o runlevel. Os scripts en `/etc/rcX.d/` tiñan prefixos `S` (start) e `K` (kill) que indicaban que servizos debían ser iniciados ou parados cando se cambiaba de runlevel.
Aínda que Debian e outras distribucións que usan **Systemd** manteñen a compatibilidade co antigo sistema, agora os servizos son xestionados a través de **units** en **Systemd**. As unidades de `systemd` (como `*.service`, `*.target`, `*.socket`, etc.) substitúen aos scripts antigos e xestionan os servizos de maneira máis flexible.
Sen embargo, en sistemas modernos, os directorios `/etc/rcX.d/` e os scripts de `/etc/init.d/` seguen existindo por compatibilidade, e **Systemd** pode chamar aos scripts antigos cando é necesario.

Comparativa

- **SysVinit**: Usaba `/etc/init.d/` para os scripts e `/etc/rcX.d/` para xestionar os servizos. Estes scripts eran chamados cando se cambiaba de runlevel. Exemplo:
  O script para o servidor de rede podería ser `/etc/init.d/networking`
- **Systemd**: Substitúe os scripts de `/etc/init.d/` por unidades `*.service`, `*.target`, etc., pero mantén a compatibilidade cos antigos scripts e directorios por se se require.
  A xestión dos servizos é feita a través de unidades .service, que se atopan en /etc/systemd/system/ ou /lib/systemd/system/, dependendo da distribución. Os servizos non se xestionan con scripts, senón que se definen por unidades que indican como iniciar, parar e reiniciar os servizos.
  Exemplo:
  Un servizo en **Systemd** sería algo como `networking.service`
  Para iniciar un servizo en **Systemd**, empregarías un comando como:

```
# systemctl start networking.service
```

/etc/rcX.d (directorios de runlevel)

SysVinit

Nos antigos sistemas **SysVinit**, os directorios `/etc/rcX.d/` correspondían aos diferentes runlevels (onde *X* era o número do runlevel, de 0 a 6).
Dentro de cada directorio `/etc/rcX.d/`, os scripts con nomes como **S** (de start) ou **K** (de kill) indicaban que servizo debía iniciarse ou deterse en cada runlevel. O sistema buscaba os scripts que comezaban con **S** para iniciar servizos e con **K** para detelos.
Exemplo:

- `/etc/rc3.d/S20networking`: iniciaría o servizo de rede no runlevel 3.
- `/etc/rc3.d/K10apache2`: detería o servizo apache2 no runlevel 3.

Systemd

En **Systemd**, a estrutura `/etc/rcX.d/` xa non existe. O concepto de runlevels foi substituído polos **targets**.
Systemd utiliza unidades para xestionar servizos. Cando cambias de **target**, **Systemd** activa ou desactiva as unidades correspondentes.
En lugar de usar scripts en `/etc/rcX.d/`, **Systemd** usa `.service` para servizos, `.target` para os diferentes niveis de execución (como `multi-user.target` ou `graphical.target`), `.socket` para xestionar as conexións de rede e comunicacións entre servizos, e `.mount` para puntos de montaxe, entre outros.
Exemplo: Para iniciar un servizo de rede no `multi-user.target`, usarías:

```
# systemctl start network.service
```

Comandos Básicos con systemctl

Algúns comandos útiles para xestionar os servizos e os targets en **Systemd** son:

- **Listar todos os targets:**

```
# systemctl list-units --type target
```

- **Iniciar un servizo:**

```
# systemctl start networking.service
```

- **Deter un servizo:**

```
# systemctl stop networking.service
```

- **Cambiar ao target desexado (por exemplo, modo gráfico):**

```
# systemctl isolate graphical.target
```

- **Verificar o target por defecto:**

```
# systemctl get-default
```

- **Habilitar un servizo para que arranque automaticamente:**

```
# systemctl enable networking.service
```

Conclusión

A transición de **SysVinit** a **Systemd** introduce melloras significativas na xestión do arranque e dos servizos, a pesar de manter a compatibilidade cos métodos antigos. O sistema baseado en **runlevels** foi substituído polos máis flexibles **targets**, e os antigos scripts en `/etc/init.d/` e directorios `/etc/rcX.d/` foron substituídos por **units** `systemd`. Con todo, **Systemd** permite seguir utilizando as ferramentas e scripts antigos cando sexa necesario.

Ricardo Feijoo Costa