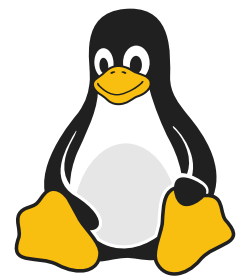
GNU/Linux

systemd - System adminstration and services

Nicolas Delanoue

Université d'Angers - Polytech Angers



System monitoring command

The following commands allow to know the state of the system :

- ps : lists the active processes,
- top : lists the active processes and indicates the busy resources (cpu, mem, ...),
- htop : like top but more explicit,
- free : gives the amount of free memory,
- df -h : indicates the available space on the hard disks (see also `du`),
- netstat : indicates the ports used (TCP addresses).

Remark

All these commands draw information from the files of `/proc`.

Definition - Service

A *deamon*, also called a service, is a process that runs in the that runs in the background.

Example of services

apache : `http` server with which browsers (client) `http` will exchange,

vsftp : `ftp` server with which `ftp` client like filezilla can copy files,

cups : Common Unix Printing System, a software like libreoffice just has to discuss with cups,

lightdm : graphic environment, if you stop it, you will have only a black and white screen, dots . . .

. . . : everything that runs in the background without making an icon in the taskbar ...

Deamon administration

Systemd is an initialization system and daemon that has been specifically designed for the Linux kernel. It allows, in particular, to configure the services that are started at startup.

More classical deamon

sshd	Secure Shell server,
proftpd	ftp server,
sendmail	SMTP server to send emails,
cron	job scheduler ,
mpd	music player deamon,
systemd	the deamon that manages the other deamons...
...	

Services management

`systemctl` is the main tool used to inspect and control the status of the system “systemd”.

Remark

For example, you can use `systemctl` to enable/disable services permanently or only for the current only for the current session.

List the active services

```
user@pc:~$ systemctl list-units --type=service
[...]
mailgraph.service      loaded active running SYSV: mailgraph postfix ...
mariadb.service        loaded active running MariaDB database server
memcached.service      loaded active running Memcached
named.service          loaded active running Berkeley Internet Name ...
network.service        loaded active exited  LSB: Bring up/down ...
ssh.service            loaded active running OpenBSD Secure Shell server
[...]
```

ou bien

```
user@pc:~$ systemctl status
```

Status of the service

```
user@pc:~$ systemctl is-active postfix
active
```

Stopping and Starting a service

```
user@pc:~$ systemctl is-active postfix
active
user@pc:~$ systemctl stop postfix
user@pc:~$ systemctl is-active postfix
inactive
user@pc:~$ systemctl start postfix
user@pc:~$ systemctl is-active postfix
active
```

Activate a service at startup

```
$ systemctl enable postfix
Created symlink from /etc/systemd/system/multi-...
```

Désactiver un service au démarrage

```
$ systemctl disable postfix
Removed symlink /etc/systemd/system/multi-user...
```

Configuration

The services are set up by editing files in the directory `/etc/`. These are often text files.

Taking into account configuration changes

restart Relaunch the service

```
$ systemctl restart postfix
```

reload Reload the service. Indeed, in order to do not kill a service in use, we prefer to type

```
$ systemctl reload postfix
```

Remarks about service configurations

- Each daemon has its own configuration files in a subdirectory of `/etc/` which often bears the name of the daemon.
- The syntax of these files changes from one daemon to another.
- Most often, one just partially modified the default files.

Definition

A *shell script* allows to automate a series of operations. of operations. It takes the form of a file containing one or more commands that will be executed in a sequential sequentially.

Example

```
#!/bin/bash
echo "Please enter your name"
read name
# Show the given name
echo The name given was :$name
```

Some remarks

- The first line tells your shell which interpreter should take care of the rest of the file, here it is `/bin/bash`.
- Don't forget to make the file containing this script executable with `chmod +x script.sh`.

Script parameters

- `$0` Script name
- `$1` First parameter
- `$2, $3, etc.` Second, third parameter, etc.
- `$*` All the parameters
- `$#` Number of arguments

Example of script called `script.sh`

```
#!/bin/bash
echo The first argument is : $1
```

```
user@pc: $ ./script.sh toto
The first argument is : toto
```

Bash language capability

- Variables,
- Arithmetic operations,
- loops,
- conditions,
- ...

Url

`https://devhints.io/bash`