

# Infrastructure technologique et virtualisation

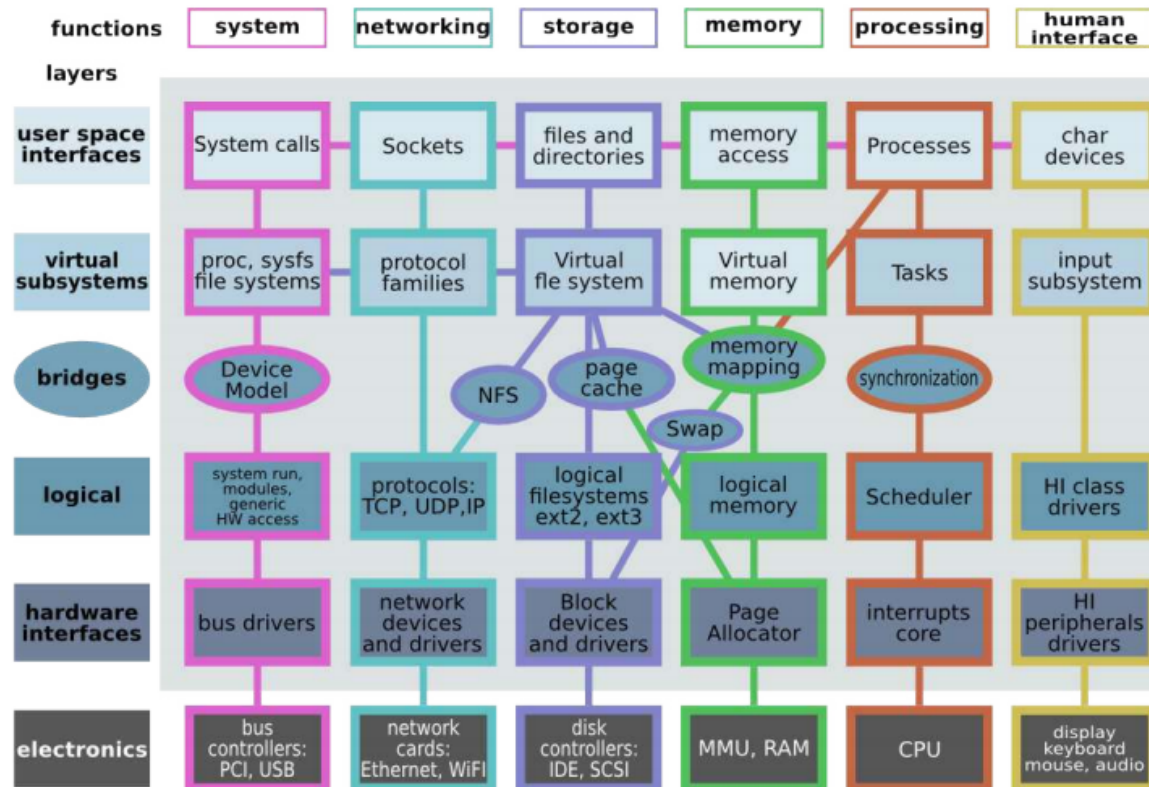
## **Noyau et processus**

Source : ROHAUT, Sébastien (2020), Linux Maîtrisez l'administration du système (6<sup>e</sup> édition) . Édition Eni ISBN : 9782409025716

Jean-Pierre Duchesneau, Été 2021

# Noyau Linux

Linux kernel diagram



© 2007-2009 Constantine Shulyupin <http://www.MakeLinux.net/kernel/diagram>

## Code source

---

- Le noyau Linux, créé en 1991 par Linus Torvalds, est distribué sous les termes de la GNU General Public License (GNU GPL) version 2. Il existe deux branches de développement mainline et vanilla. Cette dernière est généralement utilisée par les distributions. Le téléchargement du code source s'effectue depuis le site web <http://www.kernel.org>.
- Le code source est écrit principalement en langage C comme tous les systèmes de la famille Unix (Unix Like). Le code de bas niveau est écrit en langage assembleur.
- Il est compilé avec gcc (The GNU Compiler Collection).

# Numéros de version

La version du noyau est composée de plusieurs nombres :

```
5.4.0-37-generic
ipduches@VM-DevOps3
```

- Le premier est le numéro majeur.
- Le deuxième est le numéro mineur. Si sa valeur était impaire, il indiquait une version de développement, sinon la valeur paire indiquait alors une version stable. Cette notion n'est plus vraie à partir du noyau 2.6.
- Le troisième numéro de révision indique que des corrections de bugs, de sécurité ou de nouvelles fonctionnalités sont apportées. Par exemple : 2.6.39.
- Le quatrième numéro de « patch level » indique l'intégration de patches de correction de bugs, de sécurité ou d'optimisation sans ajouter de nouvelles fonctionnalités. Par exemple : 3.5.0-28.

Au-delà des quatre numéros de la version, vous avez des informations propres à la distribution.

# Version du noyau

Pour identifier la version utilisée, nous disposons d'au moins deux possibilités.

La première est de lire le contenu du fichier `/proc/version`.

La seconde est d'utiliser la commande `uname` avec les commutateurs suivants :

Option		Description
courte	longue	
-s	--kernel-name	Affiche le nom du noyau.
-r	--kernel-release	Affiche le numéro de version du noyau.
-v	--kernel-version	Affiche la version du noyau.

```
jpduches@VM-DevOpsJPD: ~  
jpduches@VM-DevOpsJPD:~$ uname -s  
Linux  
jpduches@VM-DevOpsJPD:~$ uname -r  
5.4.0-37-generic  
jpduches@VM-DevOpsJPD:~$ uname -v  
#41-Ubuntu SMP Wed Jun 3 18:57:02 UTC 2020  
jpduches@VM-DevOpsJPD:~$ uname  
Linux  
jpduches@VM-DevOpsJPD:~$ cat /proc/version  
Linux version 5.4.0-37-generic (buildd@lcy01-amd64-001) (gcc version 9.3.0 (Ubuntu 9.3.0-10ubuntu2)) #41  
-Ubuntu SMP Wed Jun 3 18:57:02 UTC 2020  
jpduches@VM-DevOpsJPD:~$
```

# Modules du noyau

- Depuis la version 2.0, le noyau Linux est modulaire ce qui signifie qu'il peut augmenter ses capacités grâce à l'utilisation de modules (Loadable Kernel Modules ou en abrégé LKM) tels que des pilotes de périphériques, un pare-feu (firewall), des protocoles réseau, etc. Ces derniers peuvent être chargés ou déchargés dynamiquement sans avoir besoin de recompiler le noyau.
- La commande `lsmod` affiche les modules chargés :

```
jpduches@VM-DevOpsJPD:~$ lsmod
Module                  Size  Used by
btrfs                   1253376  0
xor                      24576  1 btrfs
zstd_compress           167936  1 btrfs
raid6_pq                 114688  1 btrfs
ufs                       81920  0
qnx4                     16384  0
hfsplus                  110592  0
hfs                      61440  0
minix                    36864  0
ntfs                     106496  0
msdos                     20480  0
jfs                      188416  0
xfs                      1277952  0
libcrc32c                16384  2 btrfs,xfs
cpuid                    16384  0
binfmt_misc              24576  1
nls_utf8                 16384  1
isofs                    49152  1
vmw_vsock_vmci_transport 32768  2
vsock                    36864  3 vmw_vsock_vmci_transport
intel_rapl_msr           20480  0
intel_rapl_common        24576  1 intel_rapl_msr
sb_edac                  32768  0
crct10dif_pclmul         16384  1
ghash_clmulni_intel      16384  0
aesni_intel              372736  0
anitre Modules
```

# Gestion des services

---

Un service (dæmon ou démon) est un programme qui démarre lors du chargement du système d'exploitation et qui s'exécute en arrière-plan. Ces programmes peuvent être des serveurs web, de messagerie, de base de données, un pare-feu, entre autres...

- Historiquement les services étaient démarrés par le programme `/sbin/init` (System V initialization ou sysVinit)
- Upstart depuis Ubuntu 9.10
- **Systemd initié en 2010.**
- 2014 Certaines distributions retournent à sysVinit
- **Debian 9 et Red Hat 7 utilisent tous deux systemd.**

## La commande `systemctl` gère les services de `systemd`

systemd	Description
<code>systemctl start &lt;nom&gt;</code>	Lance un service.
<code>Systemctl stop &lt;nom&gt;</code>	Arrête un service.
<code>Systemctl restart &lt;nom&gt;</code>	Redémarre un service.
<code>Systemctl try-restart &lt;nom&gt;</code>	Redémarre un service uniquement s'il est en cours d'exécution.
<code>Systemctl reload &lt;nom&gt;</code>	Recharge la configuration.
<code>Systemctl status &lt;nom&gt;</code> <code>Systemctl is-active &lt;nom&gt;</code>	Vérifie si un service est en cours d'exécution.
<code>Systemctl lists-units --type service --all</code>	Affiche le statut de tous les services.

systemd	Description
<code>systemctl enable nom.service</code>	Active un service.
<code>systemctl disable nom.service</code>	Désactive un service.
<code>systemctl status nom.service</code> <code>systemctl is-enabled nom.service</code>	Vérifie si un service est activé.
<code>systemctl list-unit-files --type service</code>	Répertorie tous les services et vérifie s'ils sont activés.
<code>systemctl list-dependencies --after</code>	Répertorie tous les services qui doivent démarrer après l'unité spécifiée.
<code>systemctl list-dependencies --before</code>	Répertorie les services qui doivent démarrer avant l'unité spécifiée.



# systemctl

---

```
jpduches@VM-DevOpsJPD:~$ systemctl restart mysql
jpduches@VM-DevOpsJPD:~$ systemctl reload mysql
Failed to reload mysql.service: Job type reload is not applicable for unit mysql.service.
jpduches@VM-DevOpsJPD:~$ systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-06-29 11:19:49 EDT; 27s ago
     Process: 113464 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 113494 (mysqld)
      Status: "Server is operational"
        Tasks: 39 (limit: 4657)
       Memory: 358.5M
         CGroup: /system.slice/mysql.service
                └─113494 /usr/sbin/mysqld

S jun 29 11:19:48 VM-DevOpsJPD systemd[1]: Starting MySQL Community Server...
S jun 29 11:19:49 VM-DevOpsJPD systemd[1]: Started MySQL Community Server.
lines 1-13/13 (END)
```

## Top ou ps

```
top - 11:16:39 up 13 days, 30 min, 1 user, load average: 0,00, 0,00, 0,00
Tâches: 295 total, 2 en cours, 293 en veille, 0 arrêté, 0 zombie
%Cpu(s): 0,3 ut, 0,3 sy, 0,0 ni, 99,3 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 3936,2 total, 385,5 libr, 1488,0 util, 2062,7 tamp/cache
MiB Éch: 953,0 total, 947,1 libr, 5,9 util. 2117,2 dispo Mem
```

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
79709	jpduches	20	0	4160788	341704	121764	S	0,7	8,5	5:15.56	gnome-shell
933	mysql	20	0	1755592	361804	25844	S	0,3	9,0	37:13.48	mysqld
79428	jpduches	20	0	539396	80156	47964	S	0,3	2,0	0:32.69	Xorg
80384	jpduches	20	0	3421212	290592	138028	S	0,3	7,2	10:31.26	firefox
80522	jpduches	20	0	2465680	165860	121120	S	0,3	4,1	6:44.49	Web Content
111997	jpduches	20	0	967968	49940	38652	S	0,3	1,2	0:04.22	gnome-terminal-
1	root	20	0	171568	13104	8228	S	0,0	0,3	0:18.54	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.10	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-kblockd
0	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	...

```
jpduches@VM-DevOpsJPD:~$ ps aux |grep mysql
mysql      933  0.1  8.9 1755592 361804 ?        Ssl  jun16  37:13 /usr/sbin/mysqld
jpduches  113446  0.0  0.0 17680 732 pts/0    S+   11:17   0:00 grep --color=auto mysql
jpduches@VM-DevOpsJPD:~$
```

# Kill : Envoyer un signal à un processus.

## Fermer un processus proprement

```
1 $ pidof firefox
2 3143
```

Enfin pour demander au programme firefox ayant le PID 3143 de se fermer correctement, utilisez la **commande kill** de la manière suivante (sans option):

```
1 kill 3143
```

## Tuer un processus

```
1 kill -s 9 3143
```

## Fermer un processus en récupérant le PID

```
$ kill $(pidof firefox)
```

## Lister les signaux disponibles

```
1 $ kill -l
2 1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL 5) SIGTRAP
3 6) SIGABRT 7) SIGBUS 8) SIGFPE 9) SIGKILL 10) SIGUSR1
4 11) SIGSEGV 12) SIGUSR2 13) SIGPIPE 14) SIGALRM 15) SIGTERM
5 16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP
6 21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ
7 26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO 30) SIGPWR
8 31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
9 38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
10 43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
11 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
12 53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
13 58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
14 63) SIGRTMAX-1 64) SIGRTMAX
```