

Dernière mise-à-jour : 2020/01/30 03:28

LSF116 - Gestion des Paramètres du matériel et les Ressources

Fichiers Spéciaux

Dans l'ordinateur les périphériques sont reliés à un **contrôleur** qui communique avec le processeur à l'aide d'un **bus**. Le contrôleur ainsi que les périphériques nécessitent des pilotes. Sous Linux, les pilotes sont généralement fournis sous la forme d'un **module**. Chaque périphérique est représenté par un **fichier spécial** dans le répertoire **/dev** et c'est dans ce fichier que le système trouve les informations nécessaires pour s'adresser au pilote.



Important : Les périphériques qui nécessitent à ce que l'ordinateur soit éteint afin des les brancher/débrancher sont appelés communément **Cold Plug Devices**. Les périphériques qui peuvent être brancher/débrancher à chaud sont appelés des **Hot Plug Devices**.

Consultez le contenu du répertoire **/dev** :

```
SLES12SP1:~ # ls -l /dev | more
total 0
crw----- 1 root root    10, 235 Oct 24 18:15 autofs
drwxr-xr-x 2 root root    120 Oct 24 18:15 block
drwxr-xr-x 2 root root     80 Oct 24 18:15 bsg
crw----- 1 root root   10, 234 Oct 24 18:15 btrfs-control
drwxr-xr-x 3 root root     60 Oct 24 18:15 bus
lrwxrwxrwx 1 root root     3 Oct 24 18:15 cdrom -> sr0
drwxr-xr-x 2 root root  3340 Oct 24 18:40 char
crw----- 1 root root     5,  1 Oct 24 18:15 console
lrwxrwxrwx 1 root root    11 Oct 24 18:15 core -> /proc/kcore
```

```
drwxr-xr-x 2 root root      60 Oct 24 18:15 cpu
crw----- 1 root root    10, 62 Oct 24 18:15 cpu_dma_latency
lrwxrwxrwx 1 root root      14 Oct 24 18:15 device-mapper -> mapper/control
drwxr-xr-x 5 root root    100 Oct 24 18:15 disk
lrwxrwxrwx 1 root root       3 Oct 24 18:15 dvd -> sr0
crw-rw---- 1 root video  29,  0 Oct 24 18:15 fb0
lrwxrwxrwx 1 root root     13 Oct 24 18:15 fd -> /proc/self/fd
crw-rw-rw- 1 root root     1,  7 Oct 24 18:15 full
crw-rw-rw- 1 root root    10, 229 Oct 24 18:15 fuse
crw----- 1 root root   251,  0 Oct 24 18:40 hidraw0
crw----- 1 root root    10, 228 Oct 24 18:15 hpet
drwxr-xr-x 2 root root       0 Oct 24 18:15 hugepages
prw----- 1 root root       0 Oct 24 18:15 initctl
--More--
```

On peut noter dans la sortie de la commande que certains fichiers sont de type **bloc (b)**, tandis que d'autre sont de type **caractère (c)**.

```
...
brw-rw---- 1 root disk     8,  1 Oct 24 18:15 sda1
...
crw-rw-rw- 1 root tty      5,  0 Oct 24 18:15 tty
...
```

La différence entre les deux repose sur le type de communication entre le système et le module. Dans le premier cas le système accède au périphérique par des coordonnées du bloc de données sur le support tandis que dans le deuxième cas la communication d'échange de données se fait octet par octet sans utiliser des tampons.

Les deux informations clefs du fichier spécial sont situées à la place de la taille d'un fichier normal et se nomment le **majeur** et le **mineur** :

- le **majeur** identifie le pilote du périphérique et donc son contrôleur,
- le **mineur** identifie le périphérique ou une particularité du périphérique telle une partition d'un disque.

Commandes

La Commande lspci

Cette commande vous renseigne sur les adaptateurs reliés aux bus PCI, AGP et PCI express :

```
SLES12SP1:~ # lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphics Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0b.0 USB controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI Controller
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)
```

Pour obtenir de l'information sur un adaptateur spécifique, il convient d'utiliser la même commande avec l'option **-v** en spécifiant l'identifiant concerné :

```
SLES12SP1:~ # lspci -v -s 00:03.0
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
    Subsystem: Intel Corporation PR0/1000 MT Desktop Adapter
    Flags: bus master, 66MHz, medium devsel, latency 64, IRQ 19
    Memory at f0000000 (32-bit, non-prefetchable) [size=128K]
    I/O ports at d010 [size=8]
    Capabilities: [dc] Power Management version 2
    Capabilities: [e4] PCI-X non-bridge device
    Kernel driver in use: e1000
    Kernel modules: e1000
```

ou :

```
SLES12SP1:~ # lspci -vv -s 00:03.0
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
  Subsystem: Intel Corporation PRO/1000 MT Desktop Adapter
  Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
  Status: Cap+ 66MHz+ UDF- FastB2B- ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
  Latency: 64 (63750ns min)
  Interrupt: pin A routed to IRQ 19
  Region 0: Memory at f0000000 (32-bit, non-prefetchable) [size=128K]
  Region 2: I/O ports at d010 [size=8]
  Capabilities: [dc] Power Management version 2
    Flags: PMEClk- DSI+ D1- D2- AuxCurrent=0mA PME(D0-,D1-,D2-,D3hot-,D3cold-)
    Status: D0 NoSoftRst- PME-Enable- DSel=0 DScale=0 PME-
  Capabilities: [e4] PCI-X non-bridge device
    Command: DPERE- ERO+ RBC=512 OST=1
    Status: Dev=ff:1f.0 64bit- 133MHz- SCD- USC- DC=simple DMMRBC=2048 DMOST=1 DMCRS=8 RSCEM- 266MHz- 533MHz-
  Kernel driver in use: e1000
  Kernel modules: e1000
```

Options de la commande

Les options de cette commande sont :

```
SLES12SP1:~ # lspci --help
lspci: invalid option -- '-'
Usage: lspci [<switches>]

Basic display modes:
-mm      Produce machine-readable output (single -m for an obsolete format)
-t       Show bus tree

Display options:
```

```
-v      Be verbose (-vv for very verbose)
-k      Show kernel drivers handling each device
-x      Show hex-dump of the standard part of the config space
-xxx    Show hex-dump of the whole config space (dangerous; root only)
-xxxx   Show hex-dump of the 4096-byte extended config space (root only)
-b      Bus-centric view (addresses and IRQ's as seen by the bus)
-D      Always show domain numbers
```

Resolving of device ID's to names:

```
-n      Show numeric ID's
-nn     Show both textual and numeric ID's (names & numbers)
-q      Query the PCI ID database for unknown ID's via DNS
-qq     As above, but re-query locally cached entries
-Q      Query the PCI ID database for all ID's via DNS
```

Selection of devices:

```
-s [[:<domain>]:]<bus>[:<slot>][.<func>] Show only devices in selected slots
-d [<vendor>]:<device> Show only devices with specified ID's
```

Other options:

```
-i <file> Use specified ID database instead of /usr/share/pci.ids.gz
-p <file> Look up kernel modules in a given file instead of default modules.pciimap
-M       Enable `bus mapping' mode (dangerous; root only)
```

PCI access options:

```
-A <method> Use the specified PCI access method (see `-A help' for a list)
-O <par>=<val> Set PCI access parameter (see `-O help' for a list)
-G       Enable PCI access debugging
-H <mode> Use direct hardware access (<mode> = 1 or 2)
-F <file> Read PCI configuration dump from a given file
```

La Commande lsusb

Cette commande vous renseigne sur les adaptateurs reliés au bus usb :

```
SLES12SP1:~ # lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 003: ID 80ee:0021 VirtualBox USB Tablet
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

SLES12SP1:~ # lsusb -vt
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=ohci-pci/12p, 12M
   |__ Port 1: Dev 3, If 0, Class=Human Interface Device, Driver=usbhid, 12M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/12p, 480M
```

Options de la commande

Les options de cette commande sont :

```
SLES12SP1:~ # lsusb --help
Usage: lsusb [options]...
List USB devices
  -v, --verbose
      Increase verbosity (show descriptors)
  -s [[bus:][devnum]
      Show only devices with specified device and/or
      bus numbers (in decimal)
  -d vendor:[product]
      Show only devices with the specified vendor and
      product ID numbers (in hexadecimal)
  -D device
      Selects which device lsusb will examine
  -t, --tree
```

```
Dump the physical USB device hierarchy as a tree
-V, --version
    Show version of program
-h, --help
    Show usage and help
```

La Commande dmidecode

La commande **dmidecode** lit la table **DMI** (*Desktop Management Interface*) aussi appelée **SMBIOS** (*System Management BIOS*) et fourni les informations sur :

- l'état du matériel actuel,
- les extensions possibles.

```
SLES12SP1:~ # dmidecode
# dmidecode 2.12
SMBIOS 2.5 present.
10 structures occupying 450 bytes.
Table at 0x000E1000.

Handle 0x0000, DMI type 0, 20 bytes
BIOS Information
  Vendor: innotek GmbH
  Version: VirtualBox
  Release Date: 12/01/2006
  Address: 0xE0000
  Runtime Size: 128 kB
  ROM Size: 128 kB
  Characteristics:
    ISA is supported
    PCI is supported
    Boot from CD is supported
    Selectable boot is supported
```

```
8042 keyboard services are supported (int 9h)
CGA/mono video services are supported (int 10h)
ACPI is supported
```

Handle 0x0001, DMI type 1, 27 bytes

System Information

```
Manufacturer: innotek GmbH
Product Name: VirtualBox
Version: 1.2
Serial Number: 0
UUID: 1AB45D82-1CA4-4C86-8B78-9A8E4C0CE646
Wake-up Type: Power Switch
SKU Number: Not Specified
Family: Virtual Machine
```

Handle 0x0008, DMI type 2, 15 bytes

Base Board Information

```
Manufacturer: Oracle Corporation
Product Name: VirtualBox
Version: 1.2
Serial Number: 0
Asset Tag: Not Specified
Features:
  Board is a hosting board
Location In Chassis: Not Specified
Chassis Handle: 0x0003
Type: Motherboard
Contained Object Handles: 0
```

Handle 0x0003, DMI type 3, 13 bytes

Chassis Information

```
Manufacturer: Oracle Corporation
Type: Other
Lock: Not Present
```



```
Version: Not Specified
Serial Number: Not Specified
Asset Tag: Not Specified
Boot-up State: Safe
Power Supply State: Safe
Thermal State: Safe
Security Status: None
```

```
Handle 0x0007, DMI type 126, 42 bytes
Inactive
```

```
Handle 0x0005, DMI type 126, 15 bytes
Inactive
```

```
Handle 0x0006, DMI type 126, 28 bytes
Inactive
```

```
Handle 0x0002, DMI type 11, 7 bytes
OEM Strings
  String 1: vboxVer_5.1.26
  String 2: vboxRev_117224
```

```
Handle 0x0008, DMI type 128, 8 bytes
OEM-specific Type
  Header and Data:
    80 08 08 00 FF F8 15 00
```

```
Handle 0xFEFF, DMI type 127, 4 bytes
End Of Table
```

Options de la commande

Les options de cette commande sont :

```
SLES12SP1:~ # dmidecode --help
Usage: dmidecode [OPTIONS]
Options are:
  -d, --dev-mem FILE      Read memory from device FILE (default: /dev/mem)
  -h, --help              Display this help text and exit
  -q, --quiet             Less verbose output
  -s, --string KEYWORD    Only display the value of the given DMI string
  -t, --type TYPE         Only display the entries of given type
  -u, --dump              Do not decode the entries
  --dump-bin FILE         Dump the DMI data to a binary file
  --from-dump FILE        Read the DMI data from a binary file
  -V, --version           Display the version and exit
```

Répertoire /proc

Le répertoire /proc contient des fichiers et des répertoires virtuels. Le contenu de ces fichiers est créé dynamiquement lors de la consultation. Seul root peut consulter la totalité des informations dans le répertoire /proc.

```
SLES12SP1:~ # ls /proc
1      205   2450  271  7      cgroups      kallsyms      schedstat
10     206   2451  272  723    cmdline      kcore         scsi
11     209   2452  273  725    config.gz    key-users     self
12     2095  25    274  750    consoles     kmsg          slabinfo
13     21    2521  275  760    cpuinfo      kpagecount    softirqs
14     218   2522  276  773    crypto       kpageflags    stat
15     22    2558  277  8      devices      latency_stats swaps
16     220   260   278  870    diskstats    loadavg       sys
1618   2200  2609  3    872    dma          locks          sysrq-trigger
1625   222   261   35   873    driver       meminfo       sysvipc
1626   224   262   36   875    execdomains  misc          timer_list
1649   2248  263   362  876    fb           modules       timer_stats
17     228   264   37   9      filesystems  mounts        tty
```

18	23	266	380	93	fs	mtrr	uptime
19	232	267	384	acpi	interrupts	net	version
195	24	268	5	asound	iomem	pagetypeinfo	vmallocinfo
196	2446	269	56	buddyinfo	ioports	partitions	vmstat
2	2449	270	57	bus	irq	sched_debug	zoneinfo

Répertoires

ide/scsi

Ce répertoire contient des répertoires dans lesquels se trouvent des informations sur la capacité, le type et la géométrie des disques.

acpi

Ce répertoire contient des informations sur la gestion de l'énergie, les températures, les vitesses de ventilateurs, la charge des batteries.

bus

Ce répertoire contient un sous-répertoire par bus.

net

Ce répertoire contient des informations sur le réseau.

sys

Ce répertoire contient des paramètres du noyau. Certains des fichiers dans ce répertoire sont accessibles en écriture par root en temps réel. Par

exemple pour éviter des attaques réseau **DoS** utilisant la commande **ping**, saisissez la commande suivante :

```
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all [Entrée]
```

Cette commande a pour résultat d'ignorer les requêtes ping.

La commande sysctl

Les fichiers dans le répertoire **/proc/sys** peuvent être administrés par la commande **sysctl** en temps réel.

La commande **sysctl** applique les règles consignés dans le fichier **/etc/sysctl.conf** au démarrage de la machine.

Saisissez la commande :

```
SLES12SP1:~ # cat /etc/sysctl.conf
#####
#
# /etc/sysctl.conf is meant for local sysctl settings
#
# sysctl reads settings from the following locations:
# /boot/sysctl.conf-<kernelversion>
# /lib/sysctl.d/*.conf
# /usr/lib/sysctl.d/*.conf
# /usr/local/lib/sysctl.d/*.conf
# /etc/sysctl.d/*.conf
# /run/sysctl.d/*.conf
# /etc/sysctl.conf
#
# To disable or override a distribution provided file just place a
# file with the same name in /etc/sysctl.d/
#
# See sysctl.conf(5), sysctl.d(5) and sysctl(8) for more information
#
```

```
####  
kernel.sysrq = 0  
net.ipv4.ip_forward = 0  
net.ipv4.tcp_syncookies = 1  
net.ipv6.conf.all.forwarding = 0
```

Options de la commande

Les options de la commande **sysctl** sont :

```
SLES12SP1:~ # sysctl --help  
  
Usage:  
  sysctl [options] [variable[=value] ...]  
  
Options:  
  -a, --all           display all variables  
  -A                 alias of -a  
  -X                 alias of -a  
  --deprecated       include deprecated parameters to listing  
  -b, --binary        print value without new line  
  -e, --ignore        ignore unknown variables errors  
  -N, --names         print variable names without values  
  -n, --values        print only values of a variables  
  -p, --load[=<file>] read values from file  
  -f                 alias of -p  
  --system           read values from all system directories  
  -r, --pattern <expression>  
                    select setting that match expression  
  -q, --quiet         do not echo variable set  
  -w, --write         enable writing a value to variable  
  -o                 does nothing  
  -x                 does nothing
```

```
-d                alias of -h

-h, --help       display this help and exit
-V, --version    output version information and exit
```

For more details see `sysctl(8)`.



Important : Consultez la page de la traduction du manuel de [sysctl ici](#) pour comprendre la commande.

Fichiers

Processeur

```
SLES12SP1:~ # cat /proc/cpuinfo
processor      : 0
vendor_id    : GenuineIntel
cpu family   : 6
model        : 76
model name   : Intel(R) Atom(TM) x5-Z8350 CPU @ 1.44GHz
stepping     : 4
cpu MHz      : 1439.999
cache size   : 1024 KB
fpu          : yes
fpu_exception : yes
cpuid level  : 11
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2
syscall nx  rdtscp lm constant_tsc rep_good nopl xtopology nonstop_tsc pn1 pclmulqdq monitor ssse3 cx16 sse4_1
sse4_2 x2apic movbe popcnt aes rdrand hypervisor lahf_lm 3dnowprefetch
```

```
bogomips      : 2879.99
clflush size  : 64
cache_alignment : 64
address sizes  : 36 bits physical, 48 bits virtual
power management:
```

Interruptions système

```
SLES12SP1:~ # cat /proc/interrupts
          CPU0
 0:         46 IO-APIC-edge      timer
 1:         10 IO-APIC-edge      i8042
 8:          0 IO-APIC-edge      rtc0
 9:          1 IO-APIC-fasteoi    acpi
12:        156 IO-APIC-edge      i8042
14:          0 IO-APIC-edge      ata_piix
15:       2288 IO-APIC-edge      ata_piix
19:       2846 IO-APIC-fasteoi    ehci_hcd:usb1, eth0
21:      17309 IO-APIC-fasteoi    ahci, snd_intel8x0
22:         53 IO-APIC-fasteoi    ohci_hcd:usb2
NMI:          0 Non-maskable interrupts
LOC:      31134 Local timer interrupts
SPU:          0 Spurious interrupts
PMI:          0 Performance monitoring interrupts
IWI:        2129 IRQ work interrupts
RTR:          0 APIC ICR read retries
RES:          0 Rescheduling interrupts
CAL:          0 Function call interrupts
TLB:          0 TLB shootdowns
TRM:          0 Thermal event interrupts
THR:          0 Threshold APIC interrupts
MCE:          0 Machine check exceptions
MCP:          8 Machine check polls
```

```
THR:      0  Hypervisor callback interrupts
ERR:      0
MIS:      0
```



Important : Un pilote de périphérique demande au processeur de fournir un service en utilisant un IRQ. Quand la demande est faite, le processeur interrompt ses activités et passe le contrôle au pilote identifié par l'IRQ. Techniquement l'attribution d'un IRQ à un périphérique doit être exclusive. Dans le cas où deux périphériques demandent un service en même temps, c'est le périphérique ayant l'IRQ le plus bas qui est prioritaire.

Canaux DMA

```
SLES12SP1:~ # cat /proc/dma
4: cascade
```

Plages d'entrée/sortie

```
SLES12SP1:~ # cat /proc/ioports | more
0000-0cf7 : PCI Bus 0000:00
  0000-001f : dma1
  0020-0021 : pic1
  0040-0043 : timer0
  0050-0053 : timer1
  0060-0060 : keyboard
  0064-0064 : keyboard
  0070-0071 : rtc_cmos
    0070-0071 : rtc0
  0080-008f : dma page reg
  00a0-00a1 : pic2
```



```
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : 0000:00:01.1
    0170-0177 : ata_piix
01f0-01f7 : 0000:00:01.1
    01f0-01f7 : ata_piix
0376-0376 : 0000:00:01.1
    0376-0376 : ata_piix
03c0-03df : vesafb
03f6-03f6 : 0000:00:01.1
    03f6-03f6 : ata_piix
0cf8-0cff : PCI conf1
--More--
```



Important : Si deux périphériques ont le même port, les **deux** périphériques seront inutilisables.

Périphériques

```
SLES12SP1:~ # cat /proc/devices
Character devices:
 1 mem
 4 /dev/vc/0
 4 tty
 4 ttyS
 5 /dev/tty
 5 /dev/console
 5 /dev/ptmx
 7 vcs
10 misc
```

```
13 input
21 sg
29 fb
99 ppdev
116 alsa
128 ptm
136 pts
180 usb
189 usb_device
251 hidraw
252 bsg
253 watchdog
254 rtc
```

Block devices:

```
259 blkext
 8 sd
11 sr
65 sd
66 sd
67 sd
68 sd
69 sd
70 sd
71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
```

254 device-mapper

Modules

```
SLES12SP1:~ # cat /proc/modules | more
xt_pkttype 12504 3 - Live 0xffffffffa04ad000
xt_LOG 17852 12 - Live 0xffffffffa04b7000
xt_limit 12711 12 - Live 0xffffffffa04b2000
af_packet 39847 0 - Live 0xffffffffa04a2000
iscsi_ibft 12862 0 - Live 0xffffffffa049d000
iscsi_boot_sysfs 16051 1 iscsi_ibft, Live 0xffffffffa0498000
ip6t_REJECT 12939 3 - Live 0xffffffffa0493000
xt_tcpudp 12884 7 - Live 0xffffffffa048e000
nf_conntrack_ipv6 14798 3 - Live 0xffffffffa0485000
nf_defrag_ipv6 34651 1 nf_conntrack_ipv6, Live 0xffffffffa0477000
ip6table_raw 12683 1 - Live 0xffffffffa0472000
ipt_REJECT 12541 3 - Live 0xffffffffa046d000
ip6table_raw 12678 1 - Live 0xffffffffa0468000
xt_CT 12956 4 - Live 0xffffffffa0463000
ip6table_filter 12810 1 - Live 0xffffffffa045e000
ip6table_mangle 12700 0 - Live 0xffffffffa0459000
nf_conntrack_ipv4 15012 3 - Live 0xffffffffa0450000
nf_defrag_ipv4 12758 1 nf_conntrack_ipv4, Live 0xffffffffa044b000
ip_tables 23019 2 iptable_raw,iptable_filter, Live 0xffffffffa041d000
xt_conntrack 12760 6 - Live 0xffffffffa0334000
nf_conntrack 114978 4 nf_conntrack_ipv6,xt_CT,nf_conntrack_ipv4,xt_conntrack, Live 0xffffffffa0429000
ip6table_filter 12815 1 - Live 0xffffffffa0322000
--More--
```

Statistiques de l'utilisation des disques

```
SLES12SP1:~ # cat /proc/diskstats
 8      0 sda 16771 21164 6999936 2273700 2373 7375 1206384 910412 0 287112 3183788
 8      1 sda1 106 141 1976 140 0 0 0 0 0 128 140
 8      2 sda2 16576 20913 6996368 2273432 2264 7375 1206384 908668 0 285324 3181776
11      0 sr0 0 0 0 0 0 0 0 0 0 0 0
```

Partitions

```
SLES12SP1:~ # cat /proc/partitions
major minor #blocks name
 8      0 20971520 sda
 8      1 2103296 sda1
 8      2 18867200 sda2
11      0 1048575 sr0
```

Espaces de pagination

```
SLES12SP1:~ # cat /proc/swaps
Filename          Type          Size      Used      Priority
/dev/sda1         partition    2103292    0        -1
```

Statistiques d'utilisation du processeur

```
SLES12SP1:~ # cat /proc/loadavg
0.00 0.04 0.12 1/100 2624
```

Statistiques d'utilisation de la mémoire

```
SLES12SP1:~ # cat /proc/meminfo
MemTotal:      1686744 kB
MemFree:       1015144 kB
MemAvailable:  1494572 kB
Buffers:       876 kB
Cached:        585424 kB
SwapCached:    0 kB
Active:        479792 kB
Inactive:      130412 kB
Active(anon):  24348 kB
Inactive(anon): 9392 kB
Active(file):  455444 kB
Inactive(file): 121020 kB
Unevictable:   80 kB
Mlocked:       80 kB
SwapTotal:     2103292 kB
SwapFree:      2103292 kB
Dirty:         0 kB
Writeback:     0 kB
AnonPages:    23984 kB
Mapped:        11980 kB
Shmem:         9836 kB
Slab:          42960 kB
SReclaimable: 31172 kB
SUnreclaim:   11788 kB
KernelStack:  792 kB
PageTables:    2292 kB
NFS_Unstable: 0 kB
Bounce:        0 kB
WritebackTmp: 0 kB
CommitLimit:  2946664 kB
```

```
Committed_AS:      108620 kB
VmallocTotal:     34359738367 kB
VmallocUsed:       0 kB
VmallocChunk:      0 kB
HardwareCorrupted: 0 kB
AnonHugePages:    0 kB
HugePages_Total:   0
HugePages_Free:    0
HugePages_Rsvd:    0
HugePages_Surp:    0
Hugepagesize:     2048 kB
DirectMap4k:      55232 kB
DirectMap2M:     1779712 kB
```

Version du noyau

```
SLES12SP1:~ # cat /proc/version
Linux version 3.12.74-60.64.40-default (geeko@buildhost) (gcc version 4.8.5 (SUSE Linux) ) #1 SMP Wed May 10
05:07:16 UTC 2017 (4eb35ec)
```

Interprétation des informations dans /proc

Les informations brutes stockées dans /proc peuvent être interprétées grâce à l'utilisation des commandes dites de *gestion des performances* :

- free,
- uptime et w,
- iostat,
- vmstat,
- mpstat,
- sar.

Commandes

free

La commande **free** permet de donner l'état de la mémoire totale, libre, partagée, swap et bufferisée. Saisissez donc la commande suivante :

```
SLES12SP1:~ # free -m
              total        used         free       shared    buffers     cached
Mem:          1647         656         990           9           0         602
-/+ buffers/cache:          52        1594
Swap:         2053           0        2053
```

Dans le cas de cet exemple, nous pouvons constater que l'affichage montre :

- 1647 Mo de mémoire physique totale,
- 656 Mo de mémoire physique utilisée et 990 Mo de mémoire physique libre,
- 2053 Mo de mémoire swap totale et 0 Mo de swap utilisé

Les options de cette commande sont :

```
SLES12SP1:~ # free --help

Usage:
  free [options]

Options:
  -b, --bytes          show output in bytes
  -k, --kilo           show output in kilobytes
  -m, --mega          show output in megabytes
  -g, --giga          show output in gigabytes
  --tera              show output in terabytes
  -h, --human         show human-readable output
  --si                use powers of 1000 not 1024
```

```
-l, --lohi      show detailed low and high memory statistics
-o, --old      use old format (without -/+buffers/cache line)
-t, --total    show total for RAM + swap
-s N, --seconds N  repeat printing every N seconds
-c N, --count N  repeat printing N times, then exit

--help      display this help and exit
-V, --version output version information and exit
```

For more details see `free(1)`.

uptime ou w

Chacune des ces commandes indique la charge moyenne du ou des processeurs depuis 1 minute, 5 minutes et 15 minutes :

```
SLES12SP1:~ # uptime
18:59pm up 0:43, 1 user, load average: 0.00, 0.01, 0.09
SLES12SP1:~ #
SLES12SP1:~ # w
18:59:28 up 44 min, 1 user, load average: 0.00, 0.01, 0.09
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU WHAT
trainee   pts/0    2.2.0.10.rev.sfr 18:41   0.00s  0.40s  0.03s sshd: trainee [priv]
```

Les valeurs **load average** ou *charge moyenne* indiquent le nombre moyen de processus en cours de traitement ou en attente pour la période concernée.

Par exemple si les valeurs sur un système muni d'un seul processeur étaient **3,48 4,00 3,85** ceci indiquerait que le processeur a du mal à traiter les processus mettant en moyenne :

- 2,48 processus en attente dans la dernière minute,
- 3,00 processus en attente dans les dernières 5 minutes,
- 2,85 processus en attente dans les dernières 15 minutes.

Les options de ces commandes sont :

```
SLES12SP1:~ # uptime --help
Usage: uptime [OPTION]... [FILE]
Print the current time, the length of time the system has been up,
the number of users on the system, and the average number of jobs
in the run queue over the last 1, 5 and 15 minutes. Processes in
an uninterruptible sleep state also contribute to the load average.
If FILE is not specified, use /var/run/utmp. /var/log/wtmp as FILE is common.

    --help      display this help and exit
    --version   output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report uptime translation bugs to <http://translationproject.org/team/>
For complete documentation, run: info '(coreutils) uptime invocation'
SLES12SP1:~ # w --help

Usage:
  w [options]

Options:
  -h, --no-header      do not print header
  -u, --no-current     ignore current process username
  -s, --short          short format
  -n, --no-truncat    non truncated listing (large)
  -f, --from           show remote hostname field
  -o, --old-style     old style output
  -i, --ip-addr       display IP address instead of hostname (if possible)

    --help      display this help and exit
  -V, --version output version information and exit

For more details see w(1).
```



```
Retrieving: sysstat-10.2.1-9.2.x86_64.rpm ..... [done]
Checking for file conflicts: ..... [done]
(1/1) Installing: sysstat-10.2.1-9.2.x86_64 ..... [done]
```

La commande **iostat** affiche des statistiques sur l'utilisation des disques, des terminaux et des lecteurs de cartouche :

```
SLES12SP1:~ # iostat
Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.23    0.16   1.18   2.63    0.00   95.80

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda                 6.12       1017.47       222.36      3543976     774512
```

Au-dessous de la première ligne indiquant la version du noyau du système et son nom d'hôte ainsi que la date actuelle, **iostat** affiche une vue d'ensemble de l'utilisation CPU moyenne du système depuis le dernier démarrage. Le rapport d'utilisation du CPU inclut les pourcentages suivants :

- Pourcentage de temps passé en mode utilisateur (exécutant des applications, etc.)
- Pourcentage de temps passé en mode utilisateur (pour les processus qui ont modifié leur priorité de programmation à l'aide de la commande **nice**)
- Pourcentage de temps passé en mode noyau
- Pourcentage de temps passé en inactivité

Notez la valeur de **%iowait**. Dans le cas où ce pourcentage est trop élevé, ceci indique que le processeur passe son temps à attendre les entrées et les sorties de disque.

Pour surveiller la vitesse des entrées et des sorties du disque, vous pouvez utiliser la commande **hdparm**.

```
SLES12SP1:~ # hdparm -t /dev/sda

/dev/sda:
Timing buffered disk reads: 722 MB in 3.00 seconds = 240.46 MB/sec
```

Au-dessous du rapport d'utilisation du CPU de la sortie de la commande **iostat** figure le rapport d'utilisation des périphériques. Ce dernier contient une ligne pour chaque périphérique disque du système et inclut les informations suivantes :

- La spécification du périphérique, apparaissant sous la forme dev<major-number>-sequence-number où <major-number> correspond au nombre majeur du périphérique et <sequence-number> correspond à un numéro de séquence commençant par zéro.
- Le nombre de transferts (ou opérations d'E/S) par seconde.
- Le nombre de blocs de 512 octets lus par seconde.
- Le nombre de blocs de 512 octets écrits par seconde.
- Le nombre total de blocs de 512 octets lus par seconde.
- Le nombre total de blocs de 512 octets écrits par seconde.

Les options de cette commande sont :

```
SLES12SP1:~ # iostat --help
Usage: iostat [ options ] [ <interval> [ <count> ] ]
Options are:
[ -c ] [ -d ] [ -h ] [ -k | -m ] [ -N ] [ -t ] [ -V ] [ -x ] [ -y ] [ -z ]
[ -j { ID | LABEL | PATH | UUID | ... } ]
[ [ -T ] -g <group_name> ] [ -p [ <device> [,...] | ALL ] ]
[ <device> [...] | ALL ]
```

vmstat

La commande **vmstat** affiche des statistiques sur la mémoire, la pagination et la charge ponctuelle du processeur :

```
SLES12SP1:~ # vmstat 1 10
procs -----memory----- ---swap-- -----io----- -system-- -----cpu-----
 r  b    swpd   free   buff  cache   si   so    bi    bo    in   cs  us  sy  id  wa  st
 0  0      0 915724   420 712804   0   0  1151   210   21   61  0  1 96  2  0
 0  0      0 915700   420 712844   0   0    0     0   13   15  0  1 99  0  0
 0  0      0 915700   420 712844   0   0    0     0    9   13  0  0 100  0  0
 0  0      0 915700   420 712844   0   0    0     0   16   22  0  0 100  0  0
 0  0      0 915700   420 712844   0   0    0     0   10   13  0  0 100  0  0
```

0	0	0	915700	420	712844	0	0	0	0	9	11	0	0	100	0	0
0	0	0	915700	420	712844	0	0	0	0	11	15	0	1	99	0	0
0	0	0	915700	420	712844	0	0	0	0	18	24	0	0	100	0	0
0	0	0	915080	420	713632	0	0	0	1704	31	58	0	3	24	73	0
0	0	0	915080	420	713632	0	0	0	0	9	8	0	0	100	0	0

La première ligne subdivise le champ en six catégories à savoir : processus, mémoire, swap, E/S, système et CPU sur lesquelles elle donne des statistiques. La seconde ligne identifie de manière encore plus détaillée chacun des champs, permettant ainsi de parcourir simplement et rapidement l'ensemble des données lors de la recherche de statistiques spécifiques.

Les champs relatifs aux processus sont les suivants :

- r — Le nombre de processus exécutables attendant d'avoir accès au CPU
- b — Le nombre de processus exécutables dans un état de veille qui ne peut être interrompu

Les champs relatifs à la mémoire sont les suivants :

- swpd — La quantité de mémoire virtuelle utilisée
- free — La quantité de mémoire libre
- buff — La quantité de mémoire utilisée par les tampons (ou buffers)
- cache — La quantité de mémoire utilisée comme cache de pages

Les champs relatifs au swap sont les suivants :

- si — La quantité de mémoire chargée depuis le disque
- so — La quantité de mémoire déchargée sur le disque

Les champs relatifs aux Entrées/Sorties (E/S) sont les suivants :

- bi — Blocs envoyés vers un périphérique blocs
- bo — Blocs reçus d'un périphérique blocs

Les champs relatifs au système sont les suivants :

- in — Nombre d'interruptions par seconde
- cs — Nombre de changements de contexte par seconde

Les champs relatifs au CPU sont les suivants :

- us — Le pourcentage de temps pendant lequel le CPU exécute un code de niveau utilisateur
- sy — Le pourcentage de temps pendant lequel le CPU exécute un code de niveau système
- id — Le pourcentage de temps pendant lequel le CPU était inoccupé
- wa — Attente d'E/S

Les options de cette commande sont :

```
SLES12SP1:~ # vmstat --help
```

Usage:

```
vmstat [options] [delay [count]]
```

Options:

```
-a, --active          active/inactive memory
-f, --forks           number of forks since boot
-m, --slabs           slabinfo
-n, --one-header      do not redisplay header
-s, --stats           event counter statistics
-d, --disk            disk statistics
-D, --disk-sum        summarize disk statistics
-p, --partition <dev> partition specific statistics
-S, --unit <char>    define display unit
-w, --wide            wide output
-t, --timestamp       show timestamp

-h, --help           display this help and exit
-V, --version        output version information and exit
```

For more details see `vmstat(8)`.



Important : Par défaut la commande `vmstat` affiche des informations depuis le démarrage du système.

mpstat

La commande **mpstat** affiche des statistiques détaillées sur le CPU :

```
SLES12SP1:~ # mpstat
Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

09:37:30 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
09:37:30 all 0.21 0.15 1.01 2.45 0.00 0.14 0.00 0.00 0.00 96.03
```

Dans le cas où vous avez plusieurs processeurs ou coeurs, vous pouvez visualiser ces mêmes informations par unité de traitement :

```
SLES12SP1:~ # mpstat -P ALL
Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

09:38:03 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
09:38:03 all 0.21 0.15 1.01 2.43 0.00 0.14 0.00 0.00 0.00 96.06
09:38:03 0 0.21 0.15 1.01 2.43 0.00 0.14 0.00 0.00 0.00 96.06
```

Pour afficher 5 jeux de statistiques à des intervalles de 2 secondes pour tous les unités de traitement, il convient d'utiliser la commande suivante :

```
SLES12SP1:~ # mpstat -P ALL 2 5
Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

09:38:54 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
09:38:56 all 0.00 0.00 0.50 0.00 0.00 0.00 0.00 0.00 0.00 99.50
09:38:56 0 0.00 0.00 0.50 0.00 0.00 0.00 0.00 0.00 0.00 99.50

09:38:56 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
09:38:58 all 0.00 0.00 0.50 0.00 0.00 0.00 0.00 0.00 0.00 99.50
09:38:58 0 0.00 0.00 0.50 0.00 0.00 0.00 0.00 0.00 0.00 99.50

09:38:58 CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle
```

```

09:39:00    all    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    100.00
09:39:00      0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    100.00

09:39:00    CPU    %usr    %nice    %sys %iowait    %irq    %soft    %steal    %guest    %gnice    %idle
09:39:02    all    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    100.00
09:39:02      0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    100.00

09:39:02    CPU    %usr    %nice    %sys %iowait    %irq    %soft    %steal    %guest    %gnice    %idle
09:39:04    all    0.00    0.00    0.00    0.49    0.00    0.49    0.00    0.00    0.00    99.01
09:39:04      0    0.00    0.00    0.00    0.49    0.00    0.49    0.00    0.00    0.00    99.01

Average:    CPU    %usr    %nice    %sys %iowait    %irq    %soft    %steal    %guest    %gnice    %idle
Average:    all    0.00    0.00    0.20    0.10    0.00    0.10    0.00    0.00    0.00    99.60
Average:      0    0.00    0.00    0.20    0.10    0.00    0.10    0.00    0.00    0.00    99.60

```

Les options de cette commande sont :

```

SLES12SP1:~ # mpstat --help
Usage: mpstat [ options ] [ <interval> [ <count> ] ]
Options are:
[ -A ] [ -u ] [ -V ] [ -I { SUM | CPU | SCPU | ALL } ]
[ -P { <cpu> [,...] | ON | ALL } ]ifftrent

```

sar

La commande **sar** permet de surveiller toutes les ressources du système selon l'option qui est passée en argument à la commande. Quelques options importantes sont :

Option	Description
-u	Pourcentage d'utilisation du CPU
-q	Nombre de processus en attente
-r	Utilisation de la mémoire centrale

Option	Description
-w	Surveillance du swapping
-p	Surveillance de la pagination
-b	Utilisation des tampons
-d	Utilisation des disques

La commande **/usr/lib64/sa/sadc** permet de collecter les informations :

```
SLES12SP1:~ # ls /usr/lib64/sa
sa1 sa2 sadc
```

Le script **/usr/lib64/sa/sa1** exécute la commande **sadc**. Ce script prend deux options :

Option	Description
-t	L'interval entre les collectes
-n	Nombre de collectes

Le script **/usr/lib64/sa/sa2** exécute la commande **sar** et consigne les informations dans un fichier au format **/var/log/sa/sar<jj>**.

Pour pouvoir fonctionner correctement, ces scripts doivent être appelés par **cron**.

Copiez le fichier **/etc/sysstat/sysstat.cron** vers **/etc/cron.d/sysstat** :

```
SLES12SP1:~ # cp /etc/sysstat/sysstat.cron /etc/cron.d/sysstat
```

Modifiez le fichier **/etc/cron.d/sysstat** ainsi :

```
SLES12SP1:~ # vi /etc/cron.d/sysstat
SLES12SP1:~ # cat /etc/cron.d/sysstat
# crontab for sysstat

# Activity reports every 10 minutes everyday
# */10 * * * * root [ -x /usr/lib64/sa/sa1 ] && exec /usr/lib64/sa/sa1 -S ALL 1 1
*/2 * * * * root [ -x /usr/lib64/sa/sa1 ] && exec /usr/lib64/sa/sa1 -S ALL 1 1
```

```
# Update reports every 6 hours
55 5,11,17,23 * * * root [ -x /usr/lib64/sa/sa2 ] && exec /usr/lib64/sa/sa2 -A
```

Attendez deux minutes puis saisissez les commandes suivantes :

```
SLES12SP1:~ # sar
Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

14:38:01      CPU      %user      %nice      %system      %iowait      %steal      %idle
14:40:01      all       0.04       0.00       0.18       0.07       0.00       99.71
Average:      all       0.04       0.00       0.18       0.07       0.00       99.71
```

```
SLES12SP1:~ # sar -u 5 3
Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

14:41:52      CPU      %user      %nice      %system      %iowait      %steal      %idle
14:41:57      all       0.00       0.00       0.00       0.00       0.00      100.00
14:42:02      all       1.02       0.00       1.23       0.00       0.00       97.75
14:42:07      all       0.00       0.00       0.20       0.00       0.00       99.80
Average:      all       0.34       0.00       0.47       0.00       0.00       99.19
```

```
SLES12SP1:~ # sar -r 5 3
Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

14:47:55      kbmemfree kbmemused  %memused  kbbuffers  kbcached  kbcommit  %commit  kbactive  kbinact  kbdirty
14:48:00      840964    845780    50.14     420        730588    103676    2.74     612208  143996    0
14:48:05      839992    846752    50.20     420        730600    104456    2.76     612244  143976    52
14:48:10      840088    846656    50.19     420        730600    104456    2.76     612464  143976    52
Average:      840348    846396    50.18     420        730596    104196    2.75     612305  143983    35
```

```
SLES12SP1:~ # sar -w 5 3
Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

14:48:36      proc/s    cswch/s
```

14:48:41	0.00	16.60
14:48:46	0.00	15.60
14:48:51	0.40	20.48
Average:	0.13	17.56

SLES12SP1:~ # sar -b 5 3

Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

14:49:28	tps	rtps	wtps	bread/s	bwrtn/s
14:49:33	0.00	0.00	0.00	0.00	0.00
14:49:38	0.00	0.00	0.00	0.00	0.00
14:49:43	0.00	0.00	0.00	0.00	0.00
Average:	0.00	0.00	0.00	0.00	0.00

SLES12SP1:~ # sar -d 5 3

Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

14:50:15	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
14:50:20	dev8-0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:50:20	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
14:50:25	dev8-0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:50:25	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
14:50:30	dev8-0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average:	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
Average:	dev8-0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

SLES12SP1:~ # sar -p 5 3

Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)

14:51:12	CPU	%user	%nice	%system	%iowait	%steal	%idle
14:51:17	all	0.00	0.00	0.20	0.00	0.00	99.80

```
14:51:22      all      0.00      0.00      0.20      0.40      0.00      99.40
14:51:27      all      0.00      0.00      0.00      0.00      0.00     100.00
Average:      all      0.00      0.00      0.13      0.13      0.00      99.73
```

```
SLES12SP1:~ # sar -v 5 3
```

```
Linux 3.12.74-60.64.40-default (SLES12SP1.fenestros.loc) 10/26/17 _x86_64_ (1 CPU)
```

```
14:52:04      dentunusd  file-nr  inode-nr  pty-nr
14:52:09          38542      704     42844         1
14:52:14          38542      704     42844         1
14:52:19          38542      704     42844         1
Average:          38542      704     42844         1
```

Les options de cette commande sont :

```
SLES12SP1:~ # sar --help
```

```
Usage: sar [ options ] [ <interval> [ <count> ] ]
```

Options are:

```
[ -A ] [ -B ] [ -b ] [ -C ] [ -d ] [ -F ] [ -H ] [ -h ] [ -p ] [ -q ] [ -R ]
[ -r ] [ -S ] [ -t ] [ -u [ ALL ] ] [ -V ] [ -v ] [ -W ] [ -w ] [ -y ]
[ -I { <int> [,...] | SUM | ALL | XALL } ] [ -P { <cpu> [,...] | ALL } ]
[ -m { <keyword> [,...] | ALL } ] [ -n { <keyword> [,...] | ALL } ]
[ -j { ID | LABEL | PATH | UID | ... } ]
[ -f [ <filename> ] | -o [ <filename> ] | -[0-9]+ ]
[ -i <interval> ] [ -s [ <hh:mm:ss> ] ] [ -e [ <hh:mm:ss> ] ]
```

Utilisation des commandes en production

Identifier un système limité par le processeur

Dans ce cas utilisez les commandes suivantes :

- uptime ou w
- vmstat
- mpstat -P ALL
- sar -u
- iostat -c

Identifier un système ayant un problème de mémoire

Dans ce cas utilisez les commandes suivantes :

- free
- sar -B

Identifier un système ayant un problème d'E/S

Utilisez la commande :

- iostat -d -x

Modules usb

L'**USB** (*Universal Serial Bus*) est un bus de données qui peut offrir des taux de transfert jusqu'à 480Mb/s sous la version 2.0 et jusqu'à 4.8 Gb/s sous la version 3.0. Les modules nécessaires pour les contrôleurs USB sont :

Version USB	Module	Nom Complet
1.0\1.1	UHCI	<i>Universal Controller Host Interface</i>
	OHCI	<i>Open Controller Host Interface</i>
2.0	EHCI	<i>Enhanced Host Controller Interface</i>
3.0	XHCI	<i>Extensible Host Controller Interface</i>

Le tableau suivant liste les modules couramment chargés en fonction du périphérique utilisé :

Module	Type de Périphérique
usb_storage	Supports de masse
usbhid	Périphériques HID (<i>Human Interface Device</i>)
snd-usb-audio	Cartes son usb
usbvidéo	Cartes vidéo et d'acquisition
irda-usb	Périphériques infrarouges
usbnet	Cartes réseaux usb

Les modules peuvent être chargés par un des moyens suivants :

- INITrd,
- Le processus init,
- kmod, d'une manière dynamique et transparente lors du branchement du périphérique, en utilisant le fichier **/lib/modules/2.6.18-194.3.1.el5/modules.usbmap**,
- udev,
- manuellement.



A faire : Branchez une clef USB avant de continuer. Si vous utilisez VirtualBox, activez la clef dans votre machine virtuelle grâce aux menus **Périphériques > Périphériques USB > le_nom_de_votre_clef**.

udev

Depuis le noyau Linux 2.6 Linux est capable de détecter des périphériques branchés à chaud. Cette technologie s'appelle le **hotplugging**. Le **hotplugging** est obtenu grâce à l'utilisation de trois composants :

- Udev,
- HAL,

- Dbus.

Les rôles de chaque composant sont les suivants :

- Udev se charge de créer et supprimer d'une manière dynamique les nœuds dans le répertoire **/dev**,
- HAL obtient des informations à partir d'Udev et crée un fichier au format XML représentant le périphérique branché. Il informe ensuite Nautilus en utilisant le Dbus,
- Dbus joue le rôle d'un bus système qui est utilisé pour la communication inter-processus.

Lors de démarrage de Linux, Udev joue un rôle important :

- Au démarrage **tmpfs** est monté sur **/dev**,
- Udev copie les éventuels nœuds statiques de **/lib/udev/devices** vers **/dev**,
- le démon **udev** collecte des données appelées **uevents** du noyau et cherche une règle correspondante dans le répertoire **/lib/udev/rules.d/**,
- Udev crée les nœuds et liens symboliques spécifiés dans la règle identifiée,
- Udev stocke les règles contenues dans **/lib/udev/rules.d/*.rules** en mémoire,
- En cas de modification des ces règles, Udev met à jour la mémoire.

Udev repose sur le filesystem **sysfs** monté sur **/sys** qui permet de rendre les périphériques visibles à Udev dans l'*User Space*. Par exemple, lors du branchement d'une clé USB, Udev crée **/dev/sdb1** automatiquement et utilise les informations contenues dans le fichier **/lib/modules/`uname -r`/modules.alias** pour trouver le pilote nécessaire :

Le fichier de configuration principal d'Udev est **/etc/udev/udev.conf** :

```
SLES12SP1:~ # cat /etc/udev/udev.conf
# see udev(7) for details

#udev_log="info"
```

Les fichiers de règles se trouvent dans **/lib/udev/rules.d/** :

```
SLES12SP1:~ # ls /lib/udev/rules.d/
10-dm.rules                60-persistent-alsa.rules    70-kdump.rules            85-
usb_autosuspend_devices.rules
11-dm-lvm.rules            60-persistent-input.rules   70-power-switch.rules     85-
```

usb_elotouch_wakeup.rules			
11-dm-mpath.rules	60-persistent-serial.rules	70-printers.rules	85-
usbmuxd.rules			
13-dm-disk.rules	60-persistent-storage-tape.rules	70-uaccess.rules	90-alsa-
restore.rules			
40-libgphoto2.rules	60-persistent-storage.rules	71-seat.rules	90-
haveged.rules			
40-usb-blacklist.rules	60-persistent-v4l.rules	73-seat-late.rules	90-libinput-
model-quirks.rules			
42-hd-audio-pm.rules	60-ssd-scheduler.rules	73-seat-numlock.rules	90-
mcelog.rules			
42-usb-hid-pm.rules	61-accelerometer.rules	75-net-description.rules	90-
pulseaudio.rules			
50-iscsi-firmware-login.rules	61-gnome-bluetooth-rfkill.rules	75-persistent-net-generator.rules	90-
vconsole.rules			
50-udev-default.rules	63-md-raid-arrays.rules	75-probe_mtd.rules	95-dm-
notify.rules			
55-scsi-sg3_id.rules	64-btrfs.rules	75-tty-description.rules	95-udev-
late.rules			
56-idedma.rules	64-md-raid-assembly.rules	78-sound-card.rules	95-upower-
csr.rules			
56-multipath.rules	65-wacom.rules	80-drivers.rules	95-upower-
hid.rules			
58-scsi-sg3_symlink.rules	66-kpartx.rules	80-hotplug-cpu-mem.rules	95-upower-
wup.rules			
60-cdrom_id.rules	67-kpartx-compat.rules	80-libinput-device-groups.rules	97-
hid2hci.rules			
60-drm.rules	69-dm-lvm-metad.rules	80-net-setup-link.rules	99-iwlwifi-
led.rules			
60-keyboard.rules	69-libmtp.rules	80-udisks2.rules	99-
systemd.rules			
60-openct.rules	69-xorg-vmmouse.rules	81-mptctl.rules	



Important : Il vous est possible d'ajouter des règles si besoin est. Dans ce cas, créez un fichier **99-local.rules** et éditez-le au lieu d'éditer les fichiers existants.

Comme indique le nom de chaque fichier, le contenu est composé de règles à l'attention d'udev. Le fichier des règles par défaut est le **50-udev-default.rules** :

```
SLES12SP1:~ # cat /lib/udev/rules.d/50-udev-default.rules | more
# do not edit this file, it will be overwritten on update

SUBSYSTEM=="virtio-ports", KERNEL=="vport*", ATTR{name}=="?*", SYMLINK+="virtio-ports/${attr{name}}"

# select "system RTC" or just use the first one
SUBSYSTEM=="rtc", ATTR{hctosys}=="1", SYMLINK+="rtc"
SUBSYSTEM=="rtc", KERNEL=="rtc0", SYMLINK+="rtc", OPTIONS+="link_priority=-100"

SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", IMPORT{builtin}="usb_id", IMPORT{builtin}="hwdb --subsystem=usb"
SUBSYSTEM=="input", ENV{ID_INPUT}=="", IMPORT{builtin}="input_id"
ENV{MODALIAS}!="", IMPORT{builtin}="hwdb --subsystem=$env{SUBSYSTEM}"

ACTION!="add", GOTO="default_permissions_end"

SUBSYSTEM=="tty", KERNEL=="ptmx", GROUP="tty", MODE="0666"
SUBSYSTEM=="tty", KERNEL=="tty", GROUP="tty", MODE="0666"
SUBSYSTEM=="tty", KERNEL=="tty[0-9]*", GROUP="tty", MODE="0620"
SUBSYSTEM=="vc", KERNEL=="vcs*|vcsa*", GROUP="tty"
KERNEL=="tty[A-Z]*[0-9]|pppox[0-9]*|ircomm[0-9]*|noz[0-9]*|rfcomm[0-9]*", GROUP="dialout"

SUBSYSTEM=="mem", KERNEL=="mem|kmem|port", GROUP="kmem", MODE="0640"

SUBSYSTEM=="input", KERNEL=="js[0-9]*", MODE="0664"
```

```

SUBSYSTEM=="video4linux", GROUP="video"
SUBSYSTEM=="misc", KERNEL=="agpgart", GROUP="video"
SUBSYSTEM=="graphics", GROUP="video"
SUBSYSTEM=="drm", GROUP="video"
SUBSYSTEM=="dvb", GROUP="video"

SUBSYSTEM=="sound", GROUP="audio", \
    OPTIONS+="static_node=snd/seq", OPTIONS+="static_node=snd/timer"

SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", MODE="0664"

--More--

```

Chaque règle prend la forme suivante :

KEY, [KEY, ...] NAME [, SYMLINK]

Chaque KEY est un champ au format **type=valeur** qui doit correspondre à un périphérique unique. La valeur de type peut prendre plusieurs formes :

Type	Description	Exemples
BUS	Type de bus	usb, scsi, ide
KERNEL	Le nom par défaut du périphérique donné par le noyau	hda, ttyUSB0, lp0
SUBSYSTEM	Le nom noyau du sous-système, généralement identique à la valeur du BUS	usb, scsi
DRIVER	Le nom du pilote qui contrôle le périphérique	usb-storage
ID	Le numéro du périphérique sur son bus	PCI bus id, USB id
PLACE	Ne concerne que les périphériques USB et donne la position topologique du périphérique sur son bus	S/O
SYSFS{filename}	Le nom du fichier dans /sys pour le périphérique. Ce fichier contient le fabricant, le label, le numéro de série et UUID du périphérique. La vérification de jusqu'à 5 fichiers est possible par règle	S/O
PROGRAM	Ceci permet à Udev d'appeler un programme externe pour nommer un périphérique	S/O
RESULT	Valeur à comparer au résultat de PROGRAM	S/O

NAME et SYMLINK sont utilisées pour stipuler ce que Udev doit faire avec le périphérique :

Type	Description	Exemples
NAME	Le nome du nœud dans /dev	S/O
SYMLINK	Le ou les lien(s) symbolique(s) qui pointe(nt) vers le NAME	S/O

La commande udevadm

Pour obtenir de l'information sur un périphérique il convient d'utiliser la commande **udevadm** :

```
SLES12SP1:~ # udevadm info --query=all -n /dev/sda
P: /devices/pci0000:00/0000:00:0d.0/ata1/host0/target0:0:0/0:0:0:0/block/sda
N: sda
S: disk/by-id/ata-VBOX_HARDDISK_VB10a694bd-6639b517
S: disk/by-id/scsi-0ATA_VBOX_HARDDISK_VB10a694bd-6639b517
S: disk/by-id/scsi-1ATA_VBOX_HARDDISK_VB10a694bd-6639b517
S: disk/by-id/scsi-SATA_VBOX_HARDDISK_VB10a694bd-6639b517
S: disk/by-path/pci-0000:00:0d.0-ata-1.0
S: disk/by-path/pci-0000:00:0d.0-scsi-0:0:0:0
E: DEVLINKS=/dev/disk/by-id/ata-VBOX_HARDDISK_VB10a694bd-6639b517 /dev/disk/by-id/scsi-0ATA_VBOX_HARDDISK_VB10a694bd-6639b517 /dev/disk/by-id/scsi-1ATA_VBOX_HARDDISK_VB10a694bd-6639b517 /dev/disk/by-id/scsi-SATA_VBOX_HARDDISK_VB10a694bd-6639b517 /dev/disk/by-path/pci-0000:00:0d.0-ata-1.0 /dev/disk/by-path/pci-0000:00:0d.0-scsi-0:0:0:0
E: DEVNAME=/dev/sda
E: DEVPATH=/devices/pci0000:00/0000:00:0d.0/ata1/host0/target0:0:0/0:0:0:0/block/sda
E: DEVTYPEDISK
E: ID_ATA=1
E: ID_BUS=ata
E: ID_MODEL=VBOX_HARDDISK
E: ID_MODEL_ENC=VBOX\x20HARDDISK\x20\x20\x20
E: ID_PART_TABLE_TYPE=dos
E: ID_PART_TABLE_UUID=000093fa
E: ID_PATH=pci-0000:00:0d.0-ata-1.0
E: ID_PATH_COMPAT=pci-0000:00:0d.0-scsi-0:0:0:0
E: ID_PATH_TAG=pci-0000_00_0d_0-ata-1_0
```

```
E: ID_REVISION=1.0
E: ID_SCSI=1
E: ID_SCSI_COMPAT=SATA_VBOX_HARDDISK_VB10a694bd-6639b517
E: ID_SCSI_COMPAT_TRUNCATED=SATA_VBOX_HARDDISK_VB10a694bd-6639b517
E: ID_SCSI_DI=1
E: ID_SCSI_SN=1
E: ID_SERIAL=VBOX_HARDDISK_VB10a694bd-6639b517
E: ID_SERIAL_SHORT=VB10a694bd-6639b517
E: ID_TYPE=disk
E: ID_VENDOR=ATA
E: ID_VENDOR_ENC=ATA\x20\x20\x20\x20\x20
E: MAJOR=8
E: MINOR=0
E: MPATH_SBIN_PATH=/sbin
E: SCSI_IDENT_LUN_ATA=VBOX_HARDDISK_VB10a694bd-6639b517
E: SCSI_IDENT_LUN_T10=ATA_VBOX_HARDDISK_VB10a694bd-6639b517
E: SCSI_IDENT_LUN_VENDOR=VB10a694bd-6639b517
E: SCSI_IDENT_SERIAL=VB10a694bd-6639b517
E: SCSI_MODEL=VBOX_HARDDISK
E: SCSI_MODEL_ENC=VBOX\x20HARDDISK\x20\x20\x20
E: SCSI_REVISION=1.0
E: SCSI_TPGS=0
E: SCSI_TYPE=disk
E: SCSI_VENDOR=ATA
E: SCSI_VENDOR_ENC=ATA\x20\x20\x20\x20\x20
E: SUBSYSTEM=block
E: TAGS=:systemd:
E: USEC_INITIALIZED=10361
```

Les options de la commande

Les options de la commande udevadm sont :

```
SLES12SP1:~ # udevadm --help
```

```
Usage: udevadm [--help] [--version] [--debug] COMMAND [COMMAND OPTIONS]
```

```
  info          query sysfs or the udev database
  trigger       request events from the kernel
  settle        wait for the event queue to finish
  control       control the udev daemon
  monitor       listen to kernel and udev events
  hwdb          maintain the hardware database index
  test          test an event run
  test-builtin  test a built-in command
```

```
SLES12SP1:~ # udevadm info --help
```

```
Usage: udevadm info [OPTIONS] [DEVPATH|FILE]
```

```
-q,--query=TYPE          query device information:
  name                   name of device node
  symlink                pointing to node
  path                   sys device path
  property               the device properties
  all                    all values
-p,--path=SYSPATH       sys device path used for query or attribute walk
-n,--name=NAME           node or symlink name used for query or attribute walk
-r,--root                prepend dev directory to path names
-a,--attribute-walk     print all key matches walking along the chain
                        of parent devices
-d,--device-id-of-file=FILE print major:minor of device containing this file
-x,--export              export key/value pairs
-P,--export-prefix      export the key name with a prefix
-e,--export-db           export the content of the udev database
-c,--cleanup-db         cleanup the udev database
  --version              print version of the program
-h,--help                print this message
```

Système de fichiers /sys

Le système de fichiers virtuel **/sys** a été introduit avec le noyau Linux **2.6**. Son rôle est de décrire le matériel pour udev.

Saisissez la commande suivante :

```
SLES12SP1:~ # ls -l /sys
total 0
drwxr-xr-x  2 root root 0 0ct 24 18:15 block
drwxr-xr-x 22 root root 0 0ct 24 18:15 bus
drwxr-xr-x 37 root root 0 0ct 24 18:15 class
drwxr-xr-x  4 root root 0 0ct 24 18:15 dev
drwxr-xr-x 11 root root 0 0ct 24 18:15 devices
drwxr-xr-x  4 root root 0 0ct 24 18:15 firmware
drwxr-xr-x  5 root root 0 0ct 24 18:15 fs
drwxr-xr-x  8 root root 0 0ct 24 18:15 kernel
drwxr-xr-x 122 root root 0 0ct 24 18:15 module
drwxr-xr-x  2 root root 0 0ct 24 18:15 power
```

Chaque répertoire contient des informations :

- **block**
 - contient des informations sur les périphériques bloc
- **bus**
 - contient des informations sur les bus de données
- **class**
 - contient des informations sur des classes de matériel
- **devices**
 - contient des informations sur la position des périphériques sur les bus
- **firmware**
 - contient, entre autre, des informations sur l'ACPI
- **module**
 - contient des informations sur les modules du noyau

- **power**
 - contient des informations sur la gestion de l'énergie
- **fs**
 - contient des informations sur les systèmes de fichiers

Pour illustrer ceci, saisissez la commande suivante :

```
SLES12SP1:~ # cat /sys/block/sda/sda1/size
4206592
```

Ce chiffre correspond au nombre de secteurs.

Limitation des ressources

ulimit

Les ressources disponibles aux utilisateurs peuvent être limitées par l'utilisation de la commande **ulimit**.

La commande **ulimit** gère deux types de limite, la limite *hard* en utilisant l'option **-H** et la limite *soft* en utilisant l'option **-S**. Seul root peut positionner une limite *hard* et ceci à condition que la limite ne dépasse pas les ressources réelles.

La limite *soft* est la limite imposée à l'utilisateur par défaut tandis que la limite *hard* est la limite que l'utilisateur peut atteindre en utilisant la commande `ulimit` lui-même.

L'utilisateur root peut paramétrer les limites accordées en éditant le fichier **/etc/security/limits.conf** :

```
SLES12SP1:~ # cat /etc/security/limits.conf
# /etc/security/limits.conf
#
#Each line describes a limit for a user in the form:
#
#<domain>      <type> <item> <value>
```

```
#
#Where:
#<domain> can be:
#     - a user name
#     - a group name, with @group syntax
#     - the wildcard *, for default entry
#     - the wildcard %, can be also used with %group syntax,
#       for maxlogin limit
#
#<type> can have the two values:
#     - "soft" for enforcing the soft limits
#     - "hard" for enforcing hard limits
#
#<item> can be one of the following:
#     - core - limits the core file size (KB)
#     - data - max data size (KB)
#     - fsize - maximum filesize (KB)
#     - memlock - max locked-in-memory address space (KB)
#     - nofile - max number of open files
#     - rss - max resident set size (KB)
#     - stack - max stack size (KB)
#     - cpu - max CPU time (MIN)
#     - nproc - max number of processes
#     - as - address space limit (KB)
#     - maxlogins - max number of logins for this user
#     - maxsyslogins - max number of logins on the system
#     - priority - the priority to run user process with
#     - locks - max number of file locks the user can hold
#     - sigpending - max number of pending signals
#     - msgqueue - max memory used by POSIX message queues (bytes)
#     - nice - max nice priority allowed to raise to values: [-20, 19]
#     - rtprio - max realtime priority
#
#<domain>      <type> <item>      <value>
```



```
#
#*          soft   core       0
#*          hard   rss        10000
#@student   hard   nproc      20
#@faculty   soft   nproc      20
#@faculty   hard   nproc      50
#ftp        hard   nproc      0
#@student   -      maxlogins  4

# End of file
```



Important : La valeur de la limite peut être un **nombre** ou le mot **unlimited**.

Par exemple, si root inscrit les deux ligne suivantes dans le fichier `/etc/security/limits.conf` :

```
...
trainee      soft   nofile    1024
trainee      hard   nofile    4096
...
```

la limite du nombre de fichiers ouverts simultanément par trainee est de 1 024. Par contre, trainee a la possibilité d'augmenter cette limite jusqu'à 4 096 en utilisant la commande suivante :

```
$ ulimit -n 4096
```

Pour consulter la liste des limites actuelles, il convient d'utiliser la commande `ulimit` avec l'option **-a** :

```
SLES12SP1:~ # ulimit -a
core file size      (blocks, -c) 0
data seg size       (kbytes, -d) unlimited
```

```
scheduling priority      (-e) 0
file size                (blocks, -f) unlimited
pending signals         (-i) 13039
max locked memory      (kbytes, -l) 64
max memory size        (kbytes, -m) unlimited
open files              (-n) 1024
pipe size               (512 bytes, -p) 8
POSIX message queues   (bytes, -q) 819200
real-time priority     (-r) 0
stack size             (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes     (-u) 13039
virtual memory         (kbytes, -v) unlimited
file locks             (-x) unlimited
```

Options de la commande

Les options de **ulimit** sont :

```
SLES12SP1:~ # help ulimit
ulimit: ulimit [-SHacdefilmnpqrstuvx] [limit]
  Modify shell resource limits.
  Provides control over the resources available to the shell and processes
  it creates, on systems that allow such control.
  Options:
  -S    use the `soft' resource limit
  -H    use the `hard' resource limit
  -a    all current limits are reported
  -b    the socket buffer size
  -c    the maximum size of core files created
  -d    the maximum size of a process's data segment
  -e    the maximum scheduling priority (`nice')
  -f    the maximum size of files written by the shell and its children
```

```
-i    the maximum number of pending signals
-l    the maximum size a process may lock into memory
-m    the maximum resident set size
-n    the maximum number of open file descriptors
-p    the pipe buffer size
-q    the maximum number of bytes in POSIX message queues
-r    the maximum real-time scheduling priority
-s    the maximum stack size
-t    the maximum amount of cpu time in seconds
-u    the maximum number of user processes
-v    the size of virtual memory
-x    the maximum number of file locks
```

If LIMIT is given, it is the new value of the specified resource; the special LIMIT values `soft', `hard', and `unlimited' stand for the current soft limit, the current hard limit, and no limit, respectively. Otherwise, the current value of the specified resource is printed. If no option is given, then -f is assumed.

Values are in 1024-byte increments, except for -t, which is in seconds, -p, which is in increments of 512 bytes, and -u, which is an unscaled number of processes.

Exit Status:

Returns success unless an invalid option is supplied or an error occurs.

Groupes de Contrôle

Les **Groupes de Contrôles** (*Control Groups*) aussi appelés **CGroups**, sont une nouvelle façon sous SLES 12 de contrôler et de limiter des ressources. Les groupes de contrôle permettent l'allocation de ressources, même d'une manière dynamique pendant que le système fonctionne, telles le temps processeur, la mémoire système, la bande réseau, ou une combinaison de ces ressources parmi des groupes de tâches (processus) définis par l'utilisateur et exécutés sur un système.

Les groupes de contrôle sont organisés de manière hiérarchique, comme des processus. Par contre, la comparaison entre les deux démontre que tandis que les processus se trouvent dans une arborescence unique descendant tous du processus init et héritant de l'environnement de leurs parents, les contrôles groupes peuvent être multiples donnant lieu à des arborescences ou **hiérarchies** multiples qui héritent de certains attributs de leurs

groupes de contrôle parents.

Ces hiérarchies multiples et séparés sont nécessaires parce que chaque hiérarchie est attaché à un ou plusieurs **sous-système(s)** aussi appelés des **Contrôleurs de Ressources** ou simplement des **Contrôleurs**. Les contrôleurs disponibles sous SLES 12 sont :

- **blkio** - utilisé pour établir des limites sur l'accès des entrées/sorties à partir et depuis des périphériques blocs,
- **cpu** - utilisé pour fournir aux tâches des groupes de contrôle accès au CPU grâce au planificateur,
- **cpuacct** - utilisé pour produire des rapports automatiques sur les ressources CPU utilisées par les tâches dans un groupe de contrôle,
- **cpuset** - utilisé pour assigner des CPU individuels sur un système multicoeur et des noeuds de mémoire à des tâches dans un groupe de contrôle,
- **devices** - utilisé pour autoriser ou pour refuser l'accès des tâches aux périphériques dans un groupe de contrôle,
- **freezer** - utilisé pour suspendre ou pour réactiver les tâches dans un groupe de contrôle,
- **memory** - utilisé pour établir les limites d'utilisation de la mémoire par les tâches d'un groupe de contrôle et pour générer des rapports automatiques sur les ressources mémoire utilisées par ces tâches,
- **net_cls** - utilisé pour repérer les paquets réseau avec un identifiant de classe (*classid*) afin de permettre au contrôleur de trafic Linux, **tc**, d'identifier les paquets provenant d'une tâche particulière d'un groupe de contrôle.
- **perf_event** - utilisé pour permettre le monitoring des CGroups avec l'outil perf,
- **hugetlb** - utilisé pour limiter des ressources sur des pages de mémoire virtuelle de grande taille.

Pour visualiser les hiérarchies, il convient d'utiliser la commande suivante **lssubsys**. Sous SLES cette commande n'est pas installée par défaut :

```
SLES12SP1:~ # lssubsys -am
If 'lssubsys' is not a typo you can use command-not-found to lookup the package that contains it, like this:
  cnf lssubsys
SLES12SP1:~ # cnf lssubsys
The program 'lssubsys' can be found in following packages:
 * libcgroup-tools [ path: /usr/bin/lssubsys, repository: zypp (SLES12-SP1-12.1-0) ]
 * libcgroup-tools [ path: /usr/bin/lssubsys, repository: zypp
(SUSE_Linux_Enterprise_Server_12_SP1_x86_64:SLES12-SP1-Pool) ]
 * libcgroup-tools [ path: /usr/bin/lssubsys, repository: zypp
(SUSE_Linux_Enterprise_Server_12_SP1_x86_64:SLES12-SP1-Updates) ]

Try installing with:
  zypper install libcgroup-tools
```

```
SLES12SP1:~ # zypper install libcgroup-tools
Refreshing service 'SUSE_Linux_Enterprise_Server_12_SP1_x86_64'.
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following 2 NEW packages are going to be installed:
  libcgroup-tools libcgroup1

2 new packages to install.
Overall download size: 128.8 KiB. Already cached: 0 B. After the operation, additional 351.1 KiB will be used.
Continue? [y/n/? shows all options] (y): y
Retrieving package libcgroup1-0.41.rc1-9.1.x86_64 (1/2), 43.2 KiB
( 95.6 KiB unpacked)
Retrieving: libcgroup1-0.41.rc1-9.1.x86_64.rpm
.....[done]
Retrieving package libcgroup-tools-0.41.rc1-9.1.x86_64 (2/2), 85.7 KiB
(255.5 KiB unpacked)
Retrieving: libcgroup-tools-0.41.rc1-9.1.x86_64.rpm
.....[done]
Checking for file conflicts:
.....[done]
(1/2) Installing: libcgroup1-0.41.rc1-9.1.x86_64
.....[done]
(2/2) Installing: libcgroup-tools-0.41.rc1-9.1.x86_64
.....[done]
Additional rpm output:
Updating /etc/sysconfig/cgred...
cgconfig: WARNING: this version doesn't create /sysdefault cgroup
```

Exécutez maintenant la commande suivante :

```
SLES12SP1:~ # lssubsys -am
cpuset /sys/fs/cgroup/cpuset
```

```
cpu,cpuacct /sys/fs/cgroup/cpu,cpuacct
memory /sys/fs/cgroup/memory
devices /sys/fs/cgroup/devices
freezer /sys/fs/cgroup/freezer
blkio /sys/fs/cgroup/blkio
perf_event /sys/fs/cgroup/perf_event
hugetlb /sys/fs/cgroup/hugetlb
```

Sous SLES 12, **Systemd** organise les processus dans chaque CGroup. Par exemple tous les processus démarrés par le serveur Apache se trouveront dans le même CGroup, y compris les scripts CGI. Ceci implique que la gestion des ressources en utilisant des hiérarchies est couplé avec l'arborescence des unités de Systemd.

En haut de l'arborescence des unités de Systemd se trouve la tranche root - **-.slice**, dont dépend :

- le **system.slice** - l'emplacement des services système,
- le **user.slice** - l'emplacement des sessions des utilisateurs,
- le **machine.slice** - l'emplacement des machines virtuelles et conteneurs.

En dessous des tranches peuvent se trouver :

- des **scopes** - des processus créés par **fork**,
- des **services** - des processus créés par une **Unité**.

Les slices peuvent être visualisés avec la commande suivante :

```
SLES12SP1:~ # systemctl list-units --type=slice
UNIT          LOAD   ACTIVE SUB    DESCRIPTION
-.slice       loaded active active Root Slice
system-getty.slice loaded active active system-getty.slice
system.slice  loaded active active System Slice
user-1000.slice loaded active active user-1000.slice
user.slice    loaded active active User and Session Slice
```

LOAD = Reflects whether the unit definition was properly loaded.

ACTIVE = The high-level unit activation state, i.e. generalization of SUB.

SUB = The low-level unit activation state, values depend on unit type.

5 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.

L'arborescence des unités de Systemd est la suivante :

```
SLES12SP1:~ # systemd-cgls
├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 20
├─user.slice
│   └─user-1000.slice
│       ├──session-6.scope
│       │   ├──3042 sshd: trainee [priv]
│       │   ├──3047 sshd: trainee@pts/0
│       │   ├──3048 -bash
│       │   ├──3119 su -
│       │   ├──3120 -bash
│       │   ├──6957 systemd-cgls
│       │   └─6958 systemd-cgls
│       └─user@1000.service
│           ├──3045 /usr/lib/systemd/systemd --user
│           └─3046 (sd-pam)
├─system.slice
│   ├──sshd.service
│   │   └─2200 /usr/sbin/sshd -D
│   ├──cron.service
│   │   └─1649 /usr/sbin/cron -n
│   ├──postfix.service
│   │   ├──1618 /usr/lib/postfix/master -w
│   │   ├──1626 qmgr -l -t fifo -u
│   │   └─4451 pickup -l -t fifo -u
│   ├──wickedd-nanny.service
│   │   └─876 /usr/sbin/wickedd-nanny --systemd --foreground
│   └─wickedd.service
```

```
├─875 /usr/sbin/wickedd --systemd --foreground
├─wickedd-auto4.service
│   └─873 /usr/lib/wicked/bin/wickedd-auto4 --systemd --foreground
├─wickedd-dhcp4.service
│   └─872 /usr/lib/wicked/bin/wickedd-dhcp4 --systemd --foreground
├─wickedd-dhcp6.service
│   └─870 /usr/lib/wicked/bin/wickedd-dhcp6 --systemd --foreground
├─systemd-logind.service
│   └─760 /usr/lib/systemd/systemd-logind
lines 1-35
```

En utilisant Systemd, plusieurs ressources peuvent être limitées :

- **CPUShares** - par défaut 1024,
- **MemoryLimit** - limite exprimée en Mo ou en Go. Pas de valeur par défaut,
- **BlockIOWeight** - valeur entre 10 et 1000. Pas de valeur par défaut,
- **StartupCPUShares** - comme CPUShares mais uniquement appliqué pendant le démarrage,
- **StartupBlockIOWeight** - comme BlockIOWeight mais uniquement appliqué pendant le démarrage,
- **CPUQuota** - utilisé pour limiter le temps CPU, même quand le système ne fait rien.



Important : Consultez le manuel `systemd.resource-control(5)` pour voir les paramètres CGroup qui peuvent être passés à `systemctl`.

LAB #1 - Travailler avec les cgroups sous SLES 12

Créez un service appelé **foo** :

```
SLES12SP1:~ # vi /etc/systemd/system/foo.service
SLES12SP1:~ # cat /etc/systemd/system/foo.service
[Unit]
Description=The foo service that does nothing useful
```



```
After=remote-fs.target nss-lookup.target

[Service]
ExecStart=/usr/bin/shasum /dev/zero
ExecStop=/bin/kill -WINCH ${MAINPID}

[Install]
WantedBy=multi-user.target
```

Consultez le statut du service foo :

```
SLES12SP1:~ # systemctl status foo.service
● foo.service - The foo service that does nothing useful
   Loaded: loaded (/etc/systemd/system/foo.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
```

Démarrez et activez le service :

```
SLES12SP1:~ # systemctl start foo.service
SLES12SP1:~ # systemctl enable foo.service
ln -s '/etc/systemd/system/foo.service' '/etc/systemd/system/multi-user.target.wants/foo.service'
SLES12SP1:~ # systemctl status foo.service
foo.service - The foo service that does nothing useful
   Loaded: loaded (/etc/systemd/system/foo.service; enabled)
   Active: active (running) since Thu 2017-10-26 16:09:10 CEST; 51s ago
 Main PID: 6982 (shasum)
   CGroup: /system.slice/foo.service
           └─6982 /usr/bin/shasum /dev/zero

Oct 26 16:09:10 SLES12SP1.fenestros.loc systemd[1]: Started The foo service that does nothing useful.
```

Notez que notre service a été placé dans la tranche **system.slice** :

```
SLES12SP1:~ # systemctl show -p Slice foo.service
```

```
Slice=system.slice
```

Utilisez ps pour voir le pourcentage du CPU utilisé par ce service :

```
SLES12SP1:~ # ps -p 6982 -o pid,comm,cputime,%cpu
  PID COMMAND          TIME %CPU
 6982 shalsum          00:02:38 99.3
```

Fixez maintenant la valeur de CPUShares pour ce service à 250 :

```
SLES12SP1:~ # systemctl set-property foo.service CPUShares=250
```

Cette limite est permanente et a été inscrite dans le fichier **50-CPUShares.conf** qui se trouve dans le répertoire **/etc/systemd/system/foo.service.d** :

```
SLES12SP1:~ # ls /etc/systemd/system/foo.service.d
90-CPUShares.conf
SLES12SP1:~ # cat /etc/systemd/system/foo.service.d/90-CPUShares.conf
[Service]
CPUShares=250
```



Important : En utilisant l'option **-runtime** avec la commande **systemctl set-property** il est possible d'appliquer la limite d'une manière provisoire.

Appliquez cette modification en rechargeant systemd et en re-démarrant le service foo.service :

```
SLES12SP1:~ # systemctl daemon-reload
SLES12SP1:~ # systemctl restart foo.service
```

Vérifiez maintenant que la limite a été appliquée :

```
SLES12SP1:~ # cat /sys/fs/cgroup/cpu/system.slice/foo.service/cpu.shares
250
SLES12SP1:~ # systemctl show -p MainPID foo.service
MainPID=7184
SLES12SP1:~ # cat /proc/7184/cgroup | grep foo
4:cpuacct,cpu:/system.slice/foo.service
2:name=systemd:/system.slice/foo.service
```

Créez maintenant le service **bar** :

```
SLES12SP1:~ # vi /etc/systemd/system/bar.service
SLES12SP1:~ # cat /etc/systemd/system/bar.service
[Unit]
Description=The bar service that does nothing useful
After=remote-fs.target nss-lookup.target

[Service]
ExecStart=/usr/bin/md5sum /dev/zero
ExecStop=/bin/kill -WINCH ${MAINPID}

[Install]
WantedBy=multi-user.target
```

Fixez maintenant la limite de CPUShares pour ce service à 2000 :

```
SLES12SP1:~ # systemctl set-property bar.service CPUShares=2000
Failed to set unit properties on bar.service: Unit bar.service is not loaded.
SLES12SP1:~ # systemctl enable bar.service
ln -s '/etc/systemd/system/bar.service' '/etc/systemd/system/multi-user.target.wants/bar.service'
SLES12SP1:~ # systemctl set-property bar.service CPUShares=2000
SLES12SP1:~ # systemctl daemon-reload
SLES12SP1:~ # systemctl start bar.service
SLES12SP1:~ # systemctl status bar.service
bar.service - The bar service that does nothing useful
```

```
Loaded: loaded (/etc/systemd/system/bar.service; enabled)
Drop-In: /etc/systemd/system/bar.service.d
         └─90-CPUShares.conf
Active: active (running) since Thu 2017-10-26 16:27:11 CEST; 2min 41s ago
Main PID: 7246 (md5sum)
CGroup: /system.slice/bar.service
        └─7246 /usr/bin/md5sum /dev/zero

Oct 26 16:27:11 SLES12SP1.fenestros.loc systemd[1]: Starting The bar service that does nothing useful...
Oct 26 16:27:11 SLES12SP1.fenestros.loc systemd[1]: Started The bar service that does nothing useful.
Oct 26 16:27:56 SLES12SP1.fenestros.loc systemd[1]: Started The bar service that does nothing useful.
```

Re-démarrez les services foo et bar :

```
SLES12SP1:~ # systemctl restart foo.service
SLES12SP1:~ # systemctl restart bar.service
SLES12SP1:~ # systemctl status foo.service
foo.service - The foo service that does nothing useful
  Loaded: loaded (/etc/systemd/system/foo.service; enabled)
  Drop-In: /etc/systemd/system/foo.service.d
           └─90-CPUShares.conf
  Active: active (running) since Thu 2017-10-26 16:35:05 CEST; 26s ago
  Process: 7381 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=0/SUCCESS)
  Main PID: 7384 (shasum)
  CGroup: /system.slice/foo.service
          └─7384 /usr/bin/shasum /dev/zero

Oct 26 16:35:05 SLES12SP1.fenestros.loc systemd[1]: Starting The foo service that does nothing useful...
Oct 26 16:35:05 SLES12SP1.fenestros.loc systemd[1]: Started The foo service that does nothing useful.
SLES12SP1:~ # systemctl status bar.service
bar.service - The bar service that does nothing useful
  Loaded: loaded (/etc/systemd/system/bar.service; enabled)
  Drop-In: /etc/systemd/system/bar.service.d
           └─90-CPUShares.conf
```

```
Active: active (running) since Thu 2017-10-26 16:35:11 CEST; 28s ago
Process: 7387 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=0/SUCCESS)
Main PID: 7390 (md5sum)
CGroup: /system.slice/bar.service
        └─7390 /usr/bin/md5sum /dev/zero
```

```
Oct 26 16:35:11 SLES12SP1.fenestros.loc systemd[1]: Starting The bar service that does nothing useful...
Oct 26 16:35:11 SLES12SP1.fenestros.loc systemd[1]: Started The bar service that does nothing useful.
```

Utilisez ps pour voir le pourcentage du CPU utilisé par les deux services :

```
SLES12SP1:~ # ps -p 7384,7390 -o pid,comm,cputime,%cpu
  PID COMMAND          TIME %CPU
  7384 shasum             00:00:41 11.0
  7390 md5sum            00:05:26 88.6
```

<html>

Copyright © 2004-2017 I2TCH LIMITED.

</html>
