

Documentation valable pour AbulÉdu Pro/PLM 1.4 et AbulÉdu Pro 1.6

Cette section a pour objectif de détailler l'ensemble des commandes utiles pour l'administration d'un serveur AbulÉdu (entre autres) à distance, via le protocole SSH.

Informations relatives aux disques durs

Il est indispensable de s'assurer régulièrement de l'état des disques dur et des partitions du serveur. Certaines partitions sont amenées à se remplir très vite, comme par exemple la partition `/var/log/` qui sert à stocker tous les journaux de chacun des services lancés sur la machine (très utile pour le dépannage et la recherche d'erreurs et d'informations en cas de dysfonctionnement).

Commandes

- La première commande intéressante qui est disponible sous AbulÉdu (uniquement) est “`s`”. Attention, elle n'est disponible qu'en tant qu'`abecmd` ou `root`.

Elle affiche les partitions montées ainsi que leur point de montage, leur pourcentage d'utilisation et l'espace restant disponible, le tout en couleur :

```
abecmd@servecole ~ ♪ s
servecole(Horizon dapper) : Linux 2.6.15-28-686 i686, charge 0.00
/(hda1): 18% occ. 716M libre
/home(hda11): 1% occ. 141G libre
/opt(hda7): 8% occ. 8,3G libre
/tmp(hda10): 2% occ. 1,7G libre
/usr(hda6): 36% occ. 5,8G libre
/var(hda8): 17% occ. 6,0G libre
/var/log(hda9): 2% occ. 7,0G libre
```

Ça donne un bref aperçu de l'état général du disque principal.

Commandes `mount` et `umount`

- Une autre commande permet d'arriver au même résultat : la commande “`mount`”. Cette commande, utilisée sans option, affiche les partitions montées, ainsi que leurs points de montage, leurs droits d'accès (Lecture/Écriture, Lecture seule) et, pour certaines, le comportement à adopter en cas d'erreur. Elle est disponible en tant qu'`abuladmin`.

```
abuladmin@servecole:~$ mount
/dev/hda1 on / type ext3 (rw,errors=remount-ro)
proc on /proc type proc (rw)
/sys on /sys type sysfs (rw)
varrun on /var/run type tmpfs (rw)
varlock on /var/lock type tmpfs (rw)
procbusb on /proc/bus/usb type usbfs (rw)
udev on /dev type tmpfs (rw)
```

```
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
devshm on /dev/shm type tmpfs (rw)
lrm on /lib/modules/2.6.15-28-686/volatile type tmpfs (rw)
/dev/hda11 on /home type ext3 (rw,acl,usrquota,grpquota)
/dev/hda7 on /opt type ext3 (rw)
/dev/hda10 on /tmp type ext3 (rw)
/dev/hda6 on /usr type ext3 (rw)
/dev/hda8 on /var type ext3 (rw)
/dev/hda9 on /var/log type ext3 (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
```

On constate que pour hda1, en cas d'erreur lors de la première tentative de montage, elle doit se remonter en lecture seule (ro=read only).

Cette commande peut être donc utilisée également avec des options particulières, mais simplement en tant que root. Elle permettra entre autres de monter une partition, de la remonter avec des options spécifiques, ou de la démonter avec “umount”

```
root@servecole /var/log # mount -a
# permet de monter les partitions selon les règles décrites dans le fichier
/etc/fstab

root@servecole /var/log # mount -o remount /dev/hda[x]
# permet de remonter la partition /dev/hda[x].

root@servecole /var/log # mount -o remount,ro /dev/hda[x]
# permet de remonter la partition /dev/hda[x] en readonly

root@servecole /var/log # umount /dev/hda[x]
# permet de démonter la partition /dev/hda[x]
```

Plus d'informations disponibles en tapant

```
man mount
man umount
```

Commande du

- Une autre commande intéressante, “du”, qui veut dire “Disk Usage”. Elle donne la taille d'un répertoire, ce qui permet d'identifier les répertoires plus lourds, en cas de saturation d'une partition. Cette commande est disponible quel que soit l'utilisateur connecté.

```
abuladmin@servecole:~$ du -hs /var/log/mysql
1,6M    /var/log/mysql
```

Ici, “du” nous signale que le répertoire /var/log/mysql/ pèse en tout 1,6 Mo.

Plus d'informations sur la commande “du” et ses options disponibles en tapant :

man du

Commande fsck

- Il peut arriver que, pour une raison quelconque, une partition soit défectueuse. Soit parce qu'elle est trop fragmentée, soit parce que certains iNodes ne sont pas bons... Pour résoudre ce genre de problème, il existe la commande fsck. Disponible uniquement en tant que root.

```
root@servecole /var/log 🐛 fsck /dev/sda6
fsck 1.38 (30-Jun-2005)
e2fsck 1.38 (30-Jun-2005)
/dev/sda6 : propre, 172038/1251712 fichiers, 1105363/2500107 blocs
```

Une option intéressante est l'option “-y” qui permet de répondre automatiquement “oui” lors de la demande de fixation de problèmes rencontrés par fsck.

Voilà pour ce qui est des commandes de base pour la gestion des disques durs. Il existe bien entendu d'autres commandes pour le formatage d'une partition, la création/recréation d'une table de partition, etc. mais ce sont des opérations sensibles qui pourraient porter préjudice à l'intégrité des données. Donc je préfère en rester là pour le moment.

Informations relatives à la mémoire vive du serveur et aux processus en cours d'exécution

La quantité de mémoire utilisée, la mémoire disponible, les processus en cours d'exécution, sont des informations importantes dans l'administration d'un serveur AbulÉdu. Elles permettent de comprendre, entre autres, pour quelles raisons certains ralentissements surviennent, notamment lors de l'utilisation de programmes demandant beaucoup de ressources. Cela permet également d'identifier clairement les processus courants, ceux qui gèlent, ceux qui sont lancés en plusieurs exemplaires...

Commande ps

- La commande “ps” peut être utilisée par n'importe quel utilisateur. Elle permet d'afficher la liste des processus actifs sur le serveur.

```
abuladmin@servecole:~$ ps
  PID TTY          TIME CMD
 6608 pts/0    00:00:00 su
 6609 pts/0    00:00:00 bash
 6613 pts/0    00:00:00 ps
```

La commande utilisée sans option affiche une liste de processus réduite uniquement à ceux de l'utilisateur courant.

Si on lance par exemple une autre commande qui travaille en tâche de fond, ps nous l'affichera :

```
abuladmin@servecole:~$ tail -f /var/log/syslog&
[1] 6615
Apr 12 10:24:58 servecole kernel: [17179682.440000] eth1: link up, 100Mbps,
full-duplex, lpa 0x45E1
Apr 12 10:24:59 servecole dhclient: Listening on LPF/eth1/00:11:2f:ae:18:85
Apr 12 10:24:59 servecole dhclient: Sending on LPF/eth1/00:11:2f:ae:18:85
Apr 12 10:24:59 servecole dhclient: Sending on Socket/fallback
Apr 12 10:25:03 servecole dhclient: DHCPDISCOVER on eth1 to 255.255.255.255
port 67 interval 8
Apr 12 10:25:03 servecole dhclient: DHCPOFFER from 192.168.100.1
Apr 12 10:25:03 servecole dhclient: DHCPREQUEST on eth1 to 255.255.255.255
port 67
Apr 12 10:25:03 servecole dhclient: DHCPACK from 192.168.100.1
Apr 12 10:25:03 servecole dhclient: bound to 192.168.100.173 -- renewal in
9923 seconds.
Apr 12 10:25:08 servecole kernel: [17179692.860000] eth1: no IPv6 routers
present
abuladmin@servecole:~$ ps
  PID TTY          TIME CMD
 6608 pts/0    00:00:00 su
 6609 pts/0    00:00:00 bash
 6615 pts/0    00:00:00 tail
 6616 pts/0    00:00:00 ps
```

Le "tail" lancé plus haut apparaît effectivement dans la liste. On constate en le lançant qu'il a pour PID (Process ID) 6615. Ce qui correspond bien à celui renvoyé par "ps"

il existe évidemment un grand nombre d'options disponibles avec "ps", par exemple :

ps -e : affichage de tous les processus.

ps -f et ps -l : affichage d'informations, assez complete.

ps -u user: affichage des processus de user (EUID).

Plus d'informations concernant les options disponibles sur le manuel de la commande "ps"

```
man ps
```

Commande pstree

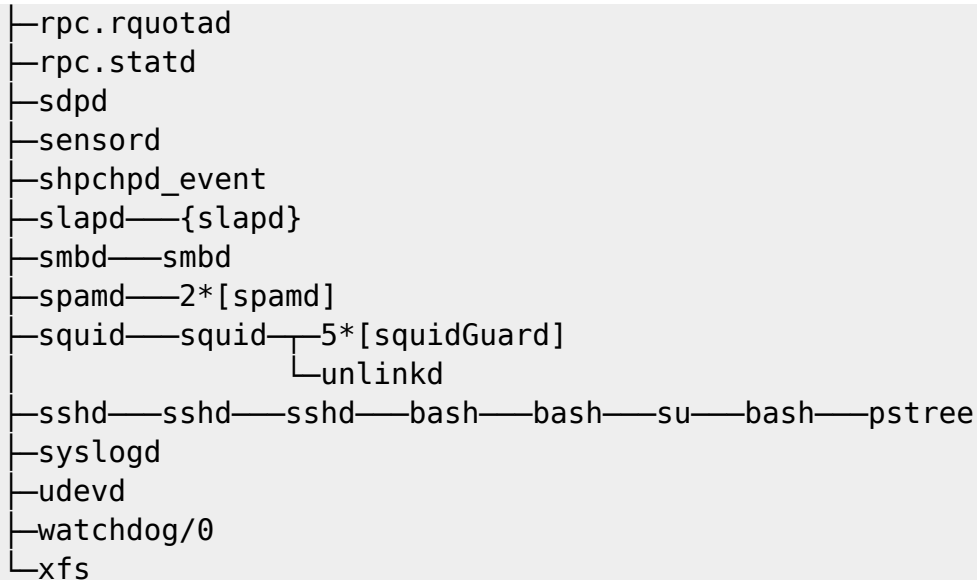
- Les processus sont lancés de manière hiérarchique. On parle donc de processus parents et enfants. La commande pstree affiche l'arborescence des processus actifs sur le serveur.

```
abuladmin@servecole:~$ pstree
init--acpid
    |--and
    |--apache2--5*[apache2]
    |--apache2--apache2
```

```

|—arpwatch
|—atd
|—courierlogger—authdaemon.pla—5*[authdaemon.pla]
|—2*[courierlogger]
|—2*[couriertcpd]
|—cron
|—cupsd
|—dbus-daemon
|—dd
|—dhclient3
|—dhcpcd3
|—events/0
|—6*[getty]
|—hackd—{hackd}
|—hald—hald-runner—hald-addon-acpi
|                                     |—hald-addon-keyb
|                                     |—2*[hald-addon-stor]
|—hcid
|—hpiod—{hpiod}
|—in.tftpd
|—kdm
|—khelper
|—7*[kjournald]
|—klogd
|—krfcommd
|—ksoftirqd/0
|—kswapd0
|—kthread—aio/0
|                                     |—ata/0
|                                     |—ata_hotplug/0
|                                     |—kacpid
|                                     |—kblockd/0
|                                     |—khubd
|                                     |—kseriod
|                                     |—nfsd4
|                                     |—2*[pdflush]
|                                     |—rpciod/0
|                                     |—scsi_eh_0
|                                     |—scsi_eh_1
|—lockd
|—mdadm
|—migration/0
|—mysqld_safe—logger
|                                     |—mysqld—8*[{mysqld}]
|—named—3*[{named}]
|—nasd
|—8*[nfsd]
|—nmbd—nmbd
|—portmap
|—python
|—rpc.mountd

```



il est tout a fait possible de réduire l'arbre à une petite famille de processus en combinant "pstree" avec la commande "grep". par exemple :

```
abuladmin@servecole:~$ pstree | grep sql
|-mysqld_safe-+-logger
|
`-mysqld--8*[{mysqld}]
```

Seul inconvénient, on ne connaîtra pas le niveau au dessus de mysql.

Dans certains cas, il est aussi utile de voir apparaître le PID de chaque processus. Pour ce faire, il existe l'option -pc. Voyons en gardant l'exemple ci-dessus :

```
abuladmin@servecole:~$ pstree -pc | grep sql
|-mysqld_safe(4597)-+-logger(4662)
|
`-mysqld(4661)-+-{mysqld}(4663)
|
|   |-{mysqld}(4664)
|   |-{mysqld}(4665)
|   |-{mysqld}(4666)
|   |-{mysqld}(4668)
|   |-{mysqld}(4669)
|   |-{mysqld}(4670)
|   `-{mysqld}(4673)
```

Commande top

- Les commandes précédentes nous permettaient d'avoir un affichage dit statique de l'ensemble des processus en cours d'exécution. Il est aussi possible de voir dynamiquement le comportement de ces processus, tels que l'utilisation en pourcentage du processeur, la quantité de mémoire utilisée, etc. Il s'agit de la commande "top"

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19031	logcheck	35	10	4292	1756	1068	S	0.7	0.4	0:00.02	logcheck
1	root	16	0	1564	528	460	S	0.0	0.1	0:01.10	init

2	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.11	events/0
6	root	10	-5	0	0	0	S	0.0	0.0	0:00.01	khelper
7	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
9	root	10	-5	0	0	0	S	0.0	0.0	0:00.45	kblockd/0
10	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
123	root	15	0	0	0	0	S	0.0	0.0	0:00.02	pdflush
124	root	15	0	0	0	0	S	0.0	0.0	0:00.18	pdflush
126	root	17	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
125	root	15	0	0	0	0	S	0.0	0.0	0:00.21	kswapd0
714	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kseriod
1831	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	ata/0
1832	root	13	-5	0	0	0	S	0.0	0.0	0:00.00	ata_hotplug/0
1836	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	scsi_eh_0
1837	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	scsi_eh_1
1909	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khubd
2106	root	15	0	0	0	0	S	0.0	0.0	0:00.09	kjournald
2281	root	13	-4	2496	988	368	S	0.0	0.2	0:00.33	udev

On constate cependant que la commande utilisée sans option nous affiche tous les processus en cours pour tous les utilisateurs. Il est tout à fait possible de n'afficher que les processus d'un utilisateur.

```
abuladmin@servecole:~$ top -u abuladmin
```

```
top - 12:05:27 up 1:42, 1 user, load average: 0.00, 0.01, 0.00
Tasks: 126 total, 2 running, 124 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.3% us, 0.0% sy, 0.0% ni, 99.7% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 482152k total, 471284k used, 10868k free, 205416k buffers
Swap: 2000052k total, 16456k used, 1983596k free, 114632k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19743	abuladmi	16	0	2200	1136	856	R	0.3	0.2	0:00.03	top
6608	abuladmi	17	0	3728	1256	916	S	0.0	0.3	0:00.00	su
6609	abuladmi	15	0	4624	2176	1420	S	0.0	0.5	0:00.04	bash

Quelques indications sur l'affichage :

PID : Correspond au Process ID des processus en cours

USER : L'utilisateur auquel appartiennent les processus affichés

PR : La priorité de la tâche

%CPU : Pourcentage de processeur utilisé

%MEM : Pourcentage de mémoire utilisée

NI : Valeur du Nice. Plus le Nice est important, moins la priorité est élevée

VIRT : Total de la mémoire virtuelle utilisée par la tâche en cours

Il est possible de sortir plus d'informations concernant les tâches. Certaines options permettent d'obtenir ces précisions. Pour cela, se référer au manuel de la commande :

```
man top
```

L'affichage étant dynamique, il est rafraîchi par défaut toutes les 5 secondes environ. Il est tout à fait possible d'optimiser la période de rafraîchissement grâce à l'option -d suivi du nombre de secondes entre chaque période :

```
top -d 1 -u abuladmin
# affiche toutes les secondes l'évolution des processus en cours d'exécution
pour l'utilisateur abuladmin
```

Commandes kill/killall

- Il peut arriver que certains programmes entrent dans des boucles sans fin et créent de forts ralentissements de la machine. Ou alors certains processus ne se sont pas terminés correctement et se retrouvent "defunct" ou "zombies". Dans ce genre de situation, une manière radicale pour se débarrasser de ces processus encombrants est la commande kill. Comme son nom (anglais) l'indique, cette commande "tue" le processus choisi en lui envoyant un signal lui ordonnant de mettre fin à son existence.

L'utilisation de cette commande est simple et possible quel que soit l'utilisateur. Il s'agit de taper la commande suivie du PID du processus trouvé à partir d'une des commandes citées plus haut (ps, pstree ou top).

```
abuladmin@servecole:~$ tail -f /var/log/syslog&
[1] 20142
Apr 12 12:00:02 servecole logger[19026]: watcher: mailfile is less than 4
lines, nothing went wrong. i don't send mail.
[...]
abuladmin@servecole:~$ ps
  PID TTY          TIME CMD
 6608 pts/0        00:00:00 su
 6609 pts/0        00:00:00 bash
20142 pts/0        00:00:00 tail
20144 pts/0        00:00:00 ps
abuladmin@servecole:~$ kill 20142
[1]+  Complété                  tail -f /var/log/syslog
abuladmin@servecole:~$ ps
  PID TTY          TIME CMD
 6608 pts/0        00:00:00 su
 6609 pts/0        00:00:00 bash
20145 pts/0        00:00:00 ps
```

Une autre commande permet également de "tuer" une famille complète de processus, que l'on peut au préalable identifier dans l'arborescence des processus (cf pstree). Il s'agit de la commande killall

suivi du nom du processus :

```
abuladmin@servecole:~$ killall mysql
```

Commande free

- Un moyen de connaître précisément l'état de la mémoire vive à un instant t est la commande "free". Elle donne la quantité de mémoire utilisée et la quantité de mémoire libre.

```
abuladmin@servecole:~$ free
              total        used         free       shared    buffers     cached
Mem:          482152      471088       11064           0       213124     112552
-/+ buffers/cache: 145412      336740
Swap:        2000052       16456     1983596
```

Il est aussi possible d'affiner l'affichage afin que ce soit plus simplement lisible. Les options -b -k et -m afficheront respectivement les valeurs en octets, kilooctets et mégaoctets.

```
abuladmin@servecole:~$ free -b
              total        used         free       shared    buffers     cached
Mem:        493723648    482394112    11329536           0   218357760   115273728
-/+ buffers/cache: 148762624    344961024
Swap:      2048053248    16850944   2031202304
abuladmin@servecole:~$ free -k
              total        used         free       shared    buffers     cached
Mem:          482152      471088       11064           0       213264     112548
-/+ buffers/cache: 145276      336876
Swap:        2000052       16456     1983596
abuladmin@servecole:~$ free -m
              total        used         free       shared    buffers     cached
Mem:              470          460           10           0           208          109
-/+ buffers/cache: 141           328
Swap:             1953           16          1937
```

Outils réseau

Dans certains cas, il sera nécessaire d'effectuer certains tests réseau de manière à optimiser la qualité de la connexion, récupérer les informations réseau d'un serveur distant ou encore connaître les clients connectés sur le serveur.

La commande who

- Il est possible de connaître la liste des utilisateurs connectés sur le serveur grâce à la commande who.

```
abuladmin@servecole:~$ who
abecmd pts/0      2007-04-12 10:25 (192.168.100.1)
```

```
abuladmin TXdev.abuledu:0 2007-04-12 15:26 (TXdev.abuledu)
```

Le résultat nous montre qu'il y a un abuladmin connecté via un TX et un abecmd connecté via un poste autonome distant.

La (célèbre) commande ping

- Afin de savoir si un ordinateur distant est bien en ligne et accessible, il existe la commande ping, qui consiste à envoyer des paquets à l'ordinateur distant, et à attendre la réponse :

```
abuladmin@servecole:~$ ping secure.ryxeo.com
PING secure.ryxeo.com (62.212.122.120) 56(84) bytes of data.
64 bytes from ryxeo.pck.nerim.net (62.212.122.120): icmp_seq=1 ttl=64
time=0.426 ms
64 bytes from ryxeo.pck.nerim.net (62.212.122.120): icmp_seq=2 ttl=64
time=0.361 ms
64 bytes from ryxeo.pck.nerim.net (62.212.122.120): icmp_seq=3 ttl=64
time=0.364 ms
64 bytes from ryxeo.pck.nerim.net (62.212.122.120): icmp_seq=4 ttl=64
time=0.426 ms

--- secure.ryxeo.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.361/0.394/0.426/0.034 ms
```

Cette commande nous montre que le serveur secure.ryxeo.com répond bien et est donc en ligne.

```
abuladmin@servecole:~$ ping 123.456.789.1
ping: unknown host 123.456.789.1
```

Le résultat de celle-ci nous montre qu'il ne parvient pas à résoudre l'adresse ip 123.456.789.1. Celle-ci n'existe pas.

```
abuladmin@servecole:~$ ping pop.wanadoo.fr
PING pop.wanadoo.fr (80.12.242.2) 56(84) bytes of data.

--- pop.wanadoo.fr ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5014ms
```

Ici, on voit qu'il y a bien une résolution de l'adresse pop.wanadoo.fr, il existe bien un serveur correspondant à ce nom, mais il n'est pas apte à répondre au ping. Soit il est hors-ligne, soit il est configuré pour ne pas répondre à ce genre de requête.

Commande mtr

- mtr est un traceroute, qui permet de connaître l'ensemble des noeuds se trouvant entre le poste local et l'ordinateur distant, point par point.

```

abuladmin@servecole:~$ mtr google.fr
                                My traceroute  [v0.69]
servecole (0.0.0.0)(tos=0x0 psize=64 bitpattern=0x00)
Thu Apr 12 15:51:32 2007
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                     Packets

Pings
Host                                     Loss%   Snt   Last
Avg  Best  Wrst StDev
 1. 192.168.100.253                       0.0%    17    0.3
0.3  0.3  0.4  0.0
 2. lo9-lns903-tip-courbevoie.nerim.net   0.0%    17   53.6
53.3 52.5 54.5  0.6
 3. gi0-2-3-holger.nerim.net              0.0%    17   54.6
66.6 53.1 212.7 38.7
 4. abovenet.panap.fr                     11.8%    17   54.9
54.2 53.3 54.9  0.5
 5. so-4-0-0.cr1.lhr3.uk.above.net        0.0%    17   61.6
68.5 60.6 107.1 14.7
 6. so-1-0-0.mpr1.lhr2.uk.above.net       0.0%    17   68.0
66.1 60.7 101.2 10.2
 7. 213-152-230-99.google.net             0.0%    17   62.1
62.4 61.5 63.5  0.5
 8. 72.14.238.242                         0.0%    17   63.2
62.7 61.5 64.8  0.8
 9. 66.249.95.107                         0.0%    17   72.5
73.2 72.5 74.1  0.5
10. 72.14.232.241                         0.0%    16   72.9
73.3 72.4 74.2  0.5
    64.233.174.185
11. 216.239.49.126                        0.0%    16   77.0
77.2 75.1 87.5  2.8
    216.239.49.114
12. gv-in-f104.google.com                 0.0%    16   72.6
73.8 72.6 74.8  0.7

```

Commande host

- Il est possible de connaître l'adresse ip d'un poste en fonction de son nom de domaine, et vice versa grâce à la commande host :

```

abuladmin@servecole:~$ host secure.ryxeo.com
secure.ryxeo.com has address 62.212.122.120
secure.ryxeo.com mail is handled by 5 secure.ryxeo.com.

```

```

abuladmin@servecole:~$ host 62.212.122.120
120.122.212.62.in-addr.arpa domain name pointer ryxeo.pck.nerim.net.

```

Commande route

- La commande route permet de connaître la table de routage ip du noyau. On peut savoir ainsi quelle interface réseau est dans quel sous réseau

Attention, cette commande n'est disponible qu'en tant que root ou abecmd

```
abecmd@servecole ~ ♪ route
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref      Use
Iface
192.168.100.0    *                255.255.255.0   U      0      0      0 eth1
localnet         *                255.255.255.0   U      0      0      0 eth0
default          192.168.100.253 0.0.0.0         UG     0      0      0 eth1
```

Commande ifconfig

- ifconfig donne la configuration de la carte réseau de la machine locale. Il est possible de lui préciser, dans le cas d'une machine équipée de plusieurs carte, quelle carte afficher. Mais par défaut, il affiche toutes celles qui sont configurées.

Attention, cette commande est également une commande disponible en tant que root ou abecmd.

```
abecmd@servecole ~ ♪ ifconfig
eth0      Lien encap:Ethernet  HWaddr 00:0A:5E:44:DD:41
          inet adr:192.168.0.1  Bcast:192.168.0.255  Masque:255.255.255.0
          adr inet6: fe80::20a:5eff:fe44:dd41/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:41030 erreurs:0 :0 overruns:0 frame:0
          TX packets:76639 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:4848647 (4.6 MiB) Octets transmis:33339181 (31.7 MiB)
          Interruption:209 Adresse de base:0x2c00

eth1      Lien encap:Ethernet  HWaddr 00:11:2F:AE:18:85
          inet adr:192.168.100.173  Bcast:192.168.100.255
          Masque:255.255.255.0
          adr inet6: fe80::211:2fff:feae:1885/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:42384 erreurs:0 :0 overruns:0 frame:0
          TX packets:18996 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:46319960 (44.1 MiB) Octets transmis:1667629 (1.5 MiB)
          Interruption:193 Adresse de base:0x8c00

lo        Lien encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
```

```
UP LOOPBACK RUNNING MTU:16436 Metric:1
Packets reçus:571 erreurs:0 :0 overruns:0 frame:0
TX packets:571 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:0
Octets reçus:49845 (48.6 KiB) Octets transmis:49845 (48.6 KiB)
```

```
abecmd@servecole ~ ♪ ifconfig eth0
eth0      Lien encap:Ethernet HWaddr 00:0A:5E:44:DD:41
          inet adr:192.168.0.1 Bcast:192.168.0.255 Masque:255.255.255.0
          adr inet6: fe80::20a:5eff:fe44:dd41/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          Packets reçus:42159 erreurs:0 :0 overruns:0 frame:0
          TX packets:78447 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:4966127 (4.7 MiB) Octets transmis:33587823 (32.0 MiB)
          Interruption:209 Adresse de base:0x2c00
```

From: <https://docs.abuledu.org/> - La documentation d'AbulÉdu

Permanent link: https://docs.abuledu.org/abuledu/administrateur/commandes_d_administration_avancee?rev=1176402063

Last update: 2007/04/12 20:21

