Unix et Programmation Shell

Philippe Langevin

département d'informatique
UFR sciences et technique
université du sud Toulon Var

Automne 2013

document

brouillon en révision

- site du cours :
  http://langevin.univ-tln.fr/cours/UPS/upsh.html
- localisation du fichier :
  http://langevin.univ-tln.fr/cours/UPS/doc/file.pdf

dernières modifications

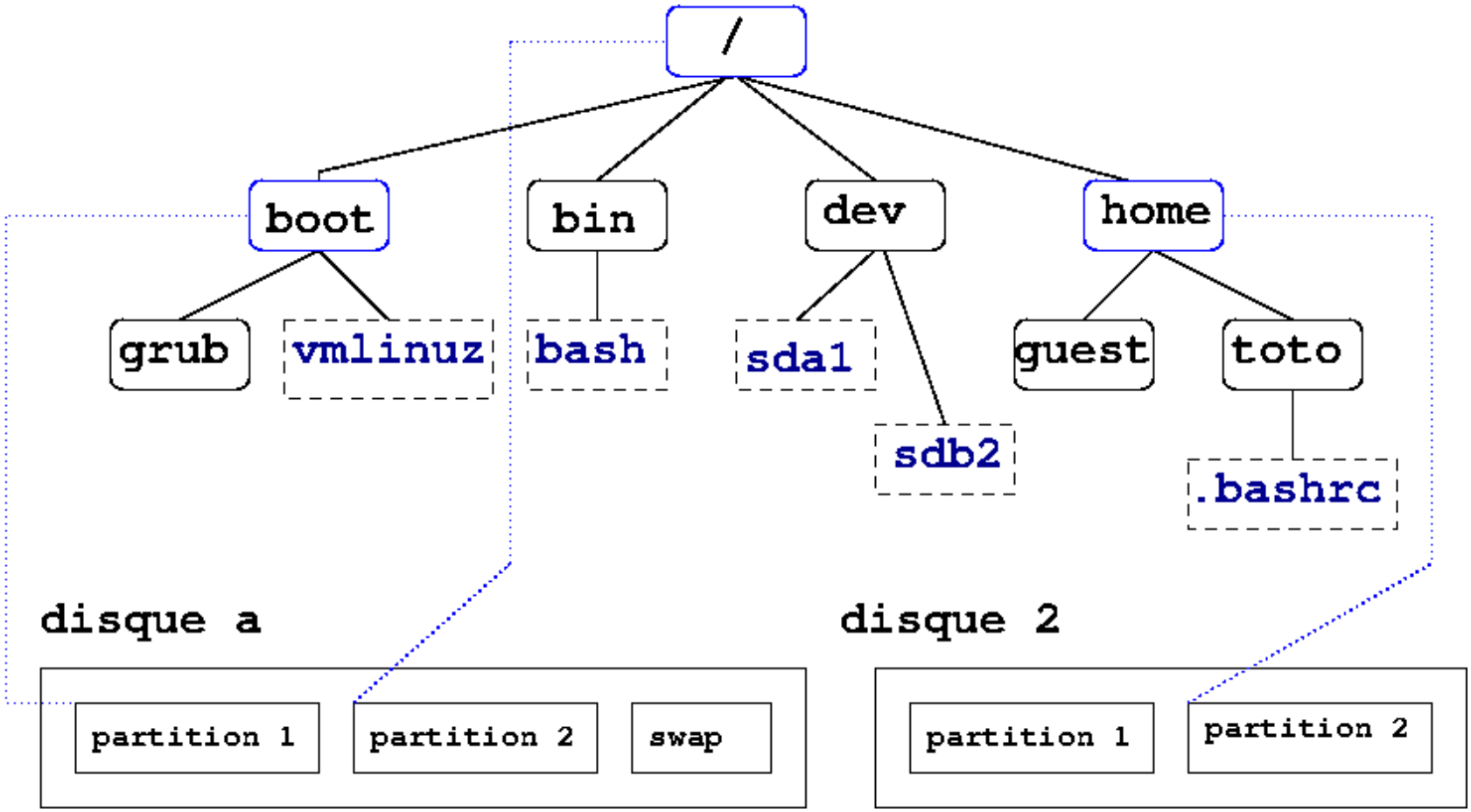
```
      man.tex 2017-09-07 12:27:47.738251920 +0200
     perm.tex 2016-09-30 09:41:54.766553521 +0200
     file.tex 2016-09-30 09:19:02.810595120 +0200
     bash.tex 2016-09-15 12:09:09.887948313 +0200
     term.tex 2016-09-14 18:50:05.124091515 +0200
     upsh.tex 2015-10-25 18:09:36.027434338 +0100
     proc.tex 2015-10-20 22:09:35.450391618 +0200
    shell.tex 2015-09-10 19:31:04.581529236 +0200
prologue.tex 2015-09-07 09:06:31.773157847 +0200
    tools.tex 2015-07-11 09:04:38.890915266 +0200
     pipe.tex 2014-10-02 19:10:22.426127326 +0200
   direct.tex 2014-10-02 07:49:17.162784238 +0200
  syntaxe.tex 2014-10-01 23:52:29.859357485 +0200
     part.tex 2014-10-01 23:52:29.372363438 +0200
```

4 - fichier

- structure
- l-noeud
- commande fichier
- répertoire
- tube nommé
- netcat
- commande réseau

structure générale

L'ensemble des systèmes de fichiers d'un système `unix` est un arbre enraciné dans répertoire / .



Systèmes de fichiers

- `mount` greffe des systèmes de fichiers, ces noeuds sont des points de montage.

```
FILESYSTEMS(5)  Linux Programmer's Manual
NAME
 filesystems - Linux file-system types: minix,
 ext, ext2, ext3, ext4, Reiserfs, XFS, JFS, xia,
 msdos, umsdos, vfat, ntfs, proc, nfs, iso9660,
 hpfs, sysv, smb, ncpfs
DESCRIPTION
 In order to use a file system, you have to mount
    it;
 proc         pseudo file system
 iso9660      CD-ROM file system
 nfs          network file system
```

Quelques points

```
⇝ mount
/dev/sda1 on /
    type ext4 (rw,errors=remount-ro)
proc       on /proc
    type proc (rw,noexec,nosuid,nodev)
tmpfs      on /media
    type tmpfs (rw,nosuid,nodev,noexec,relatime,
   seclabel,mode=755)
sinfo1:/home/perso on /home/perso
    type nfs (rw,soft,intr,sloppy,addr
   =10.9.185.1)
curlftpfs#ftp://freebox:----@mafreebox.freebox.
   fr/Disque%20dur/
       on /media/freebox
    type fuse (rw,nosuid,nodev,noexec,relatime
```

Organisation

bin	exécutables fondamentaux [binaries]
boot	noyau vmlinuz et fichiers de démarrage
dev	périphériques [device]
etc	fichiers de configuration des services
home	données personnelles des utilisateurs
lib	bibliothèques [libraries]
mnt	point de montage temporaire
media	point de montage automatique
proc	pseudo-fichiers des processus en cours
root	répertoire du super-utilisateur
sbin	[(system|superuser) binaries]
tmp	fichiers temporaires
usr	accessibles à tous les utilisateurs
var	fichiers variables, souvent modifiés

- FHS système de fichiers

structure d'un système de fichier

Un système de fichiers comprend trois zone :

- superbloc
- table de noeuds
- fichiers et répertoire

```
↝ df -i
Sys. fich.      Inodes IUtil. ILibre IUti% Monte
   sur
rootfs         1122304 160548 961756    15% /
tmpfs            63474      1  63473     1% /media
/dev/sda1       128016    349 127667     1% /boot
```

super-bloc Il contient des informations globales sur le système de fichier : taille, nom, type, nom de volume, verrous, tables de gestions...

I-noeud

Les noeuds du système de fichiers sont indexés par des entiers qui les caractérisent : *I-noeud* (inode) :

```
↝ ls -l /etc/passwd
-rw-r--r--.
    1 root root 1797  1 sept. 17:13 /etc/passwd

↝ ls -il /home/toto/.bashrc
2883586 -rw-r--r--.
    1 toto novice 124 24 nov. 2011 /home/toto/.
   bashrc
```

attribut

La commande `stat` détaille les attributs des fichiers

```
↝ stat /home/toto/.bashrc
  File: /home/toto/.bashrc
  Size: 124 Blocks: 8 IO Block: 4096 fichier
Device: fd02h/64770d Inode: 2883586 Links: 1
Access: (0644/-rw-r--r--)
                    Uid: (1002/toto) Gid: (1002/
   novice)
Context: unconfined_u:object_r:user_home_t:s0
Access: 2012-09-01 17:35:07.718610665 +0200
Modify: 2011-11-24 23:00:34.000000000 +0100
Change: 2012-09-01 17:05:20.838474864 +0200
 Birth: -
```

taille, bloc

Dans un terminal, lancer les commandes :

```
touch /tmp/x
watch stat /tmp/x
```

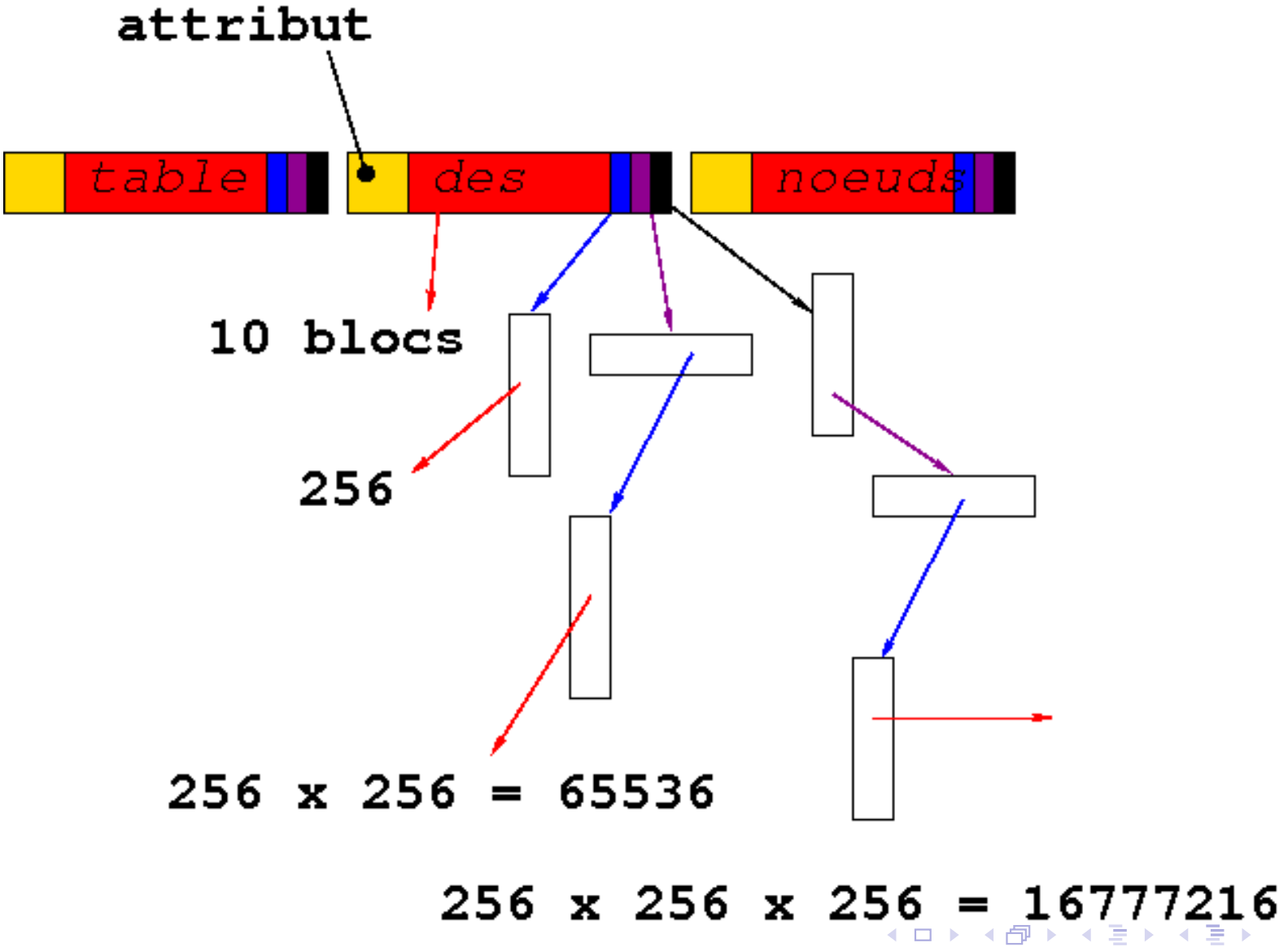
Dans un autre terminal

```
echo hello > /tmp/x
```

Observer le premier terminal. Continuer avec :

```
for((i=0; i<12; i++)) ; do
    cat /tmp/x /tmp/x > /tmp/y ;
    mv /tmp/y /tmp/x; sleep 3;
done
```

gestion des blocs (ext)



Limite

nombre de fichiers = nombre d'inoeuds.

```
↝ b=256
↝ let t=10+b+b*b+b*b*b
↝ echo $t
16843018
↝ echo "l($t)" | bc -l
16.63944676687317765852
```

```
$ bc -l << THIS
> b=256
> t=10+b+b^2+b^3
> l(t)/l(2)
> THIS
24.00564733370382901169
```

Les types de noeud

type	english	création	destruction	
fichier	file		`rm`	
répertoire	directory	`mkdir`	`rm -r`	
lien symbolique	link	`ln -s`	`rm`	
prise	socket	`socket`	`close`	IPC
tube nommé	pipe	`mkfifo`	`rm`	IPC
périphérique	caractére			
périphérique	bloc			

Quelques noeuds

```
⇝ find /dev -type b -name 's*'
/dev/sda3
⇝ tty
/dev/pts/1
⇝ ls -l $(tty)
crw--w----. 1 drmichko tty ...  /dev/pts/1
⇝ ls -l /bin/awk
lrwxrwxrwx. 1 root root 4  ... /bin/awk -> gawk
⇝ ls -ld /*
drwxr-xr-x.  19 root root  3480  ...  /dev
drwxr-xr-x.   6 root root  4096  ...  /home
dr-xr-xr-x.  19 root root 12288  ...  /lib
drwxr-xr-x.   2 root root  4096  ...  /mnt
dr-xr-xr-x. 146 root root     0  ...  /proc
```

commande fichier

- `ls`, `find`, `locate`
- `mkdir`, `cd`, `popd`, `pushd`, `dir`
- `basename`, `dirname`, `pwd`, `which`
- `touch`, `cp`, `mv`
- `tar` : archivage
- `rsync`, `unison` : sauvegarde, `git`

des statistiques :

- `df`,`du` : stat. systèmes de fichiers, disque.
- `lsof` : liste des fichiers ouverts
- `fuser` : usage des fichiers

administratives :

- `ln` : lien
- `mount` : montage
- `fdisk`, `parted` : manipuler les partitions

commande fichier

- file, type
- cmp, comm, diff
- cat, tac, tee
- more, less
- tail, head
- nl, wc,
- grep, strings
- od, hexdump
- sort, uniq
- split, csplit
- join, paste, fold
- col, tr, iconv, cut, awk, sed
- gedit, vi, emacs , aspell

compilation

- `indent` :
- `make` : maintenance
- `gcc` : compilateur gnu
- `grof` : profileur
- `gcov` : couverture
- `gdb` : debugguer
- `valgrind` : fuite mémoire
- `flex` : analyseur lexical
- `bison` : compilateur de compilateur

[abc du c]

Les répertoires

Les noms de fichiers sont stockés dans les répertoires :

inode	taille	nom
3804070	12	.
3538945	12	..
3804071	20	hello.txt
3804072	4052	Hello.txt

```
↝ mkdir tmp
↝ echo 'hello world' > tmp/hello.txt
↝ echo 'hello world' > tmp/Hello.txt
♯ echo -e 'ls /drmichko/tmp\nquit'
      | debugfs /dev/mapper/VolGroup-lv_home
debugfs:  ls /drmichko/tmp
 3804070  (12) .    3538945  (12) ..    3804071
   (20) hello.txt
```

ajout, suppression des entrées

On ajoute un fichier :

```
↝ echo 'hello world' > tmp/nouveau.txt



# echo -e 'ls /drmichko/tmp\nquit'
      | debugfs /dev/mapper/VolGroup-lv_home
debugfs: ls /drmichko/tmp
 3804070 (12) .    3538945 (12) ..    3804071
   (20) hello.txt
 3804072 (20) Hello.txt    3804067 (4032)
   nouveau.txt
```

On efface un autre :

```
↝ rm tmp/hello.txt
```

src/dir.c

```
1  #include <string.h>
2  int main()
3  {   int count;
4      struct dirent *fic ;
5      DIR *dir;
6      char *path, *tok;
7      path=getenv( "PATH" );
8      tok = strtok(path, ":");
9      do {
10         dir = opendir( tok );
11         count=0;
12         while ((fic = readdir(dir)) )
13             if ( DT_REG == fic->d_type) count++;
14         printf ("%s -> %d files\n", tok, count);
15         closedir(dir );
```

src/dir.c

```
/usr/kerberos/sbin -> 5 files
/usr/kerberos/bin -> 14 files
/usr/local/bin -> 4 files
/usr/bin -> 2273 files
/bin -> 115 files
/usr/local/sbin -> 0 files
/usr/sbin -> 362 files
/sbin -> 207 files
/home/pl/bin -> 53 files
/home/pl/script -> 36 files
```

dir.sh

```
1 #!/bin/bash
2 echo $PATH | tr ':' '\n' | while read path
3       do
4     count=$( ls -l $path | grep -cE '^-' )
5     echo $path '->' $count files
6       done
```

- `man 3 readdir`
- `man tr`, `grep`
- `help echo`, `help while`
- 2 solutions parmi d'autres !

[comparaisons] : `python`, `perl`, `bash`, `C`.

Les tubes nommés

```
↝ whoami
michko
↝ mkfifo /tmp/canal
↝ ls -l /tmp/canal
prw-rw-r--. 1 michko michko 0  3 sept. 18:55 /
    tmp/canal
↝ cat > /tmp/canal
```

michko écrit dans le tube...

```
↝ whoami
toto
↝ cat < /tmp/canal
```

toto lit dans le tube.

netcat

La commande `nc`, le couteau suisse du **net**, permet d'écouter sur une *prise* réseau :

```
↝ nc -l 31415
```

ou bien d'envoyer des données vers une *socket* distante :

```
↝ nc turing.euphoria.fr 31415
```

dig, nc

```
⇝ nc -lu 31415 | hexdump -C
```

```
⇝ dig -p 31415 www.euphoria.fr @localhost
```

dig, nc

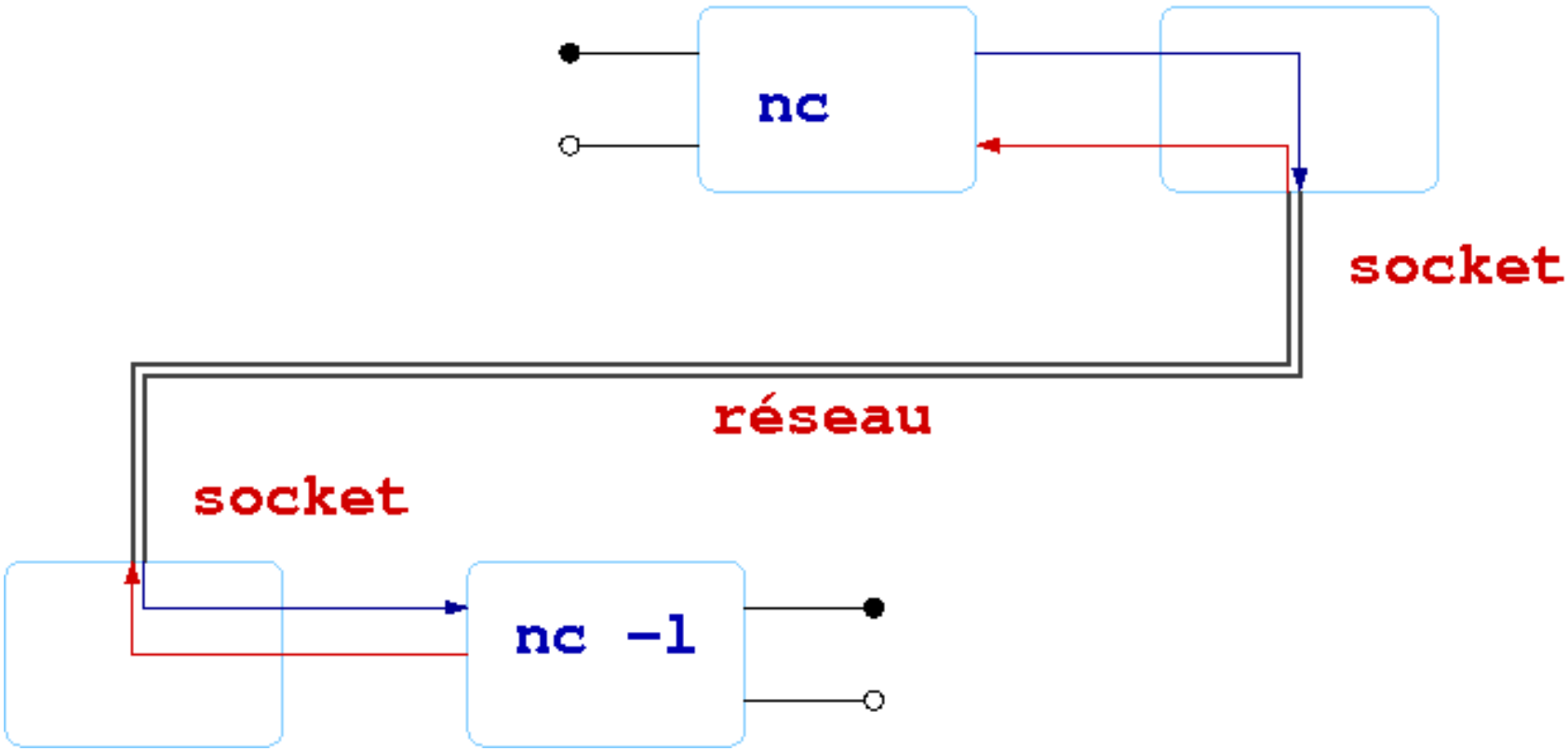
```
⇝ nc -lu 31415 | hexdump -C
```

```
⇝ dig -p 31415 www.euphoria.fr @localhost
```

```
00000000  6a 48 01 00 00 01 00 00  |jH..........|
          00 00 00 00 03 77 77 77  |.....www|
00000010  08 65 75 70 68 6f 72 69  |.euphori|
          61 02 66 72 00 00 01 00  |a.fr....|
```

```
; <<>> DiG 9.7.3-P1-RedHat-9.7.3-2.P1.fc13
  <<>> -p31415 www.euphoria.fr @localhost
;; global options: +cmd
;; connection timed out; no servers could be
   reached
```

client/serveur



gnuchess

```
⇝ gnuchess
GNU Chess 5.08
Adjusting HashSize to 1024 slots
Transposition table:  Entries=1K Size=40K
Pawn hash table: Entries=0K Size=28K
White (1) : e2e4
1. e2e4

black  KQkq  e3
r n b q k b n r
p p p p p p p p
. . . . . . . .
. . . . . . . .
. . . . P . . .
```

gnu chess service

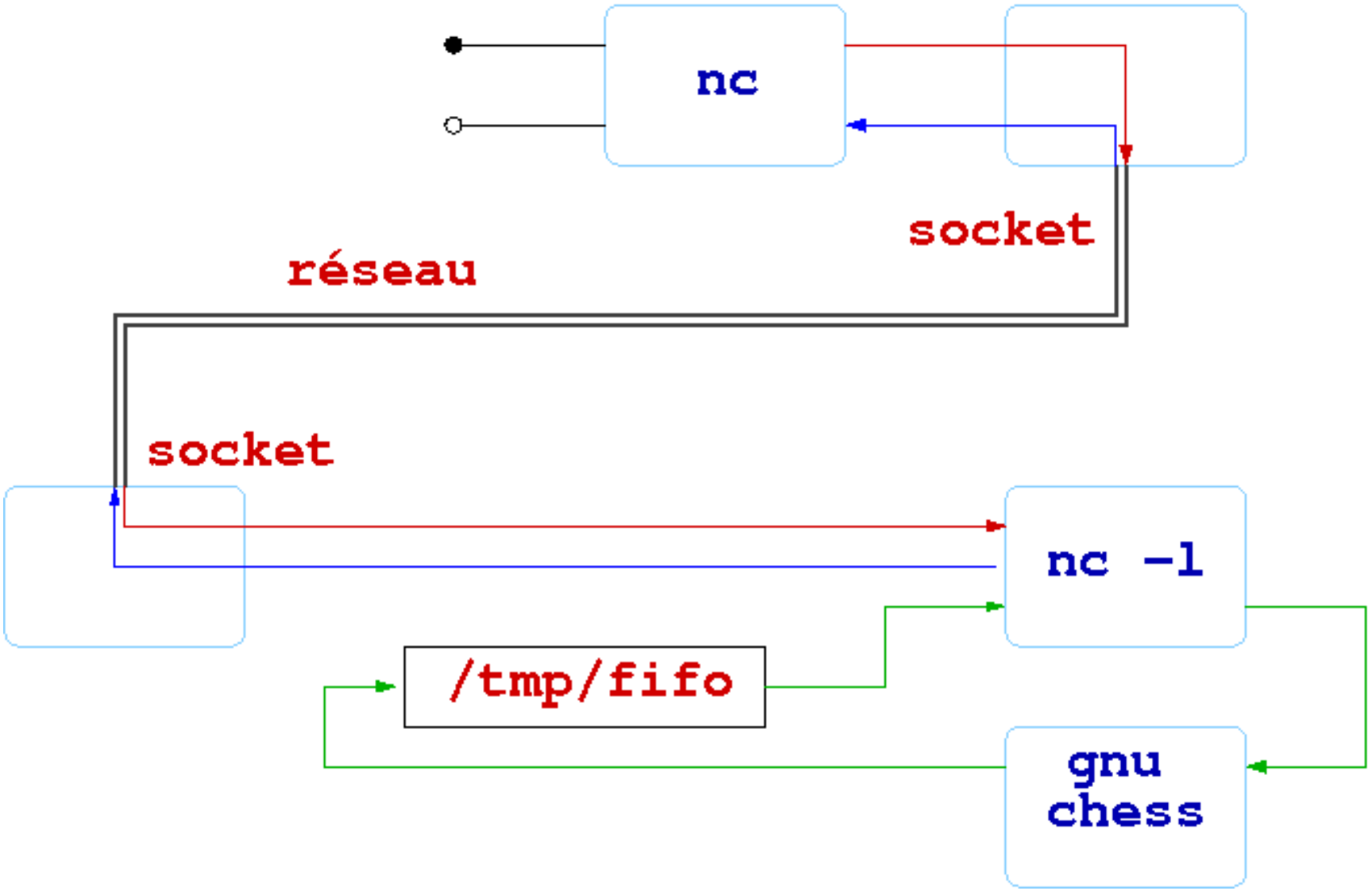
Installation du serveur :

```
↝ mkfifo /tmp/fifo
↝ nc -l 31415 < /tmp/fifo
                | gnuchess > /tmp/fifo
```

connexion d'un client :

```
↝ nc 31415
```

gnu chess service



client

```
↝ nc turing.euphoria.fr  31415
GNU Chess 5.08
Adjusting HashSize to 1024 slots
Transposition table:  Entries=1K Size=40K
Pawn hash table: Entries=0K Size=28K
White (1) : e2e4
1. e2e4

black  KQkq  e3
r n b q k b n r
p p p p p p p p
. . . . . . . .
. . . . . . . .
. . . . P . . .
```

commande réseau

- `ftp`, `sftp`, `scp`
- `ssh`
- `telnet`
- `wget`, `curl`
- `lynx`, `links`

diagnostic :

- `ping`, `host`, `dig`
- `route`, `ifconfig`,`ip`
- `traceroute`, `tracepath`
- `netstat`