

Unix et Programmation Shell

Philippe Langevin

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

Automne 2013

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/term.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

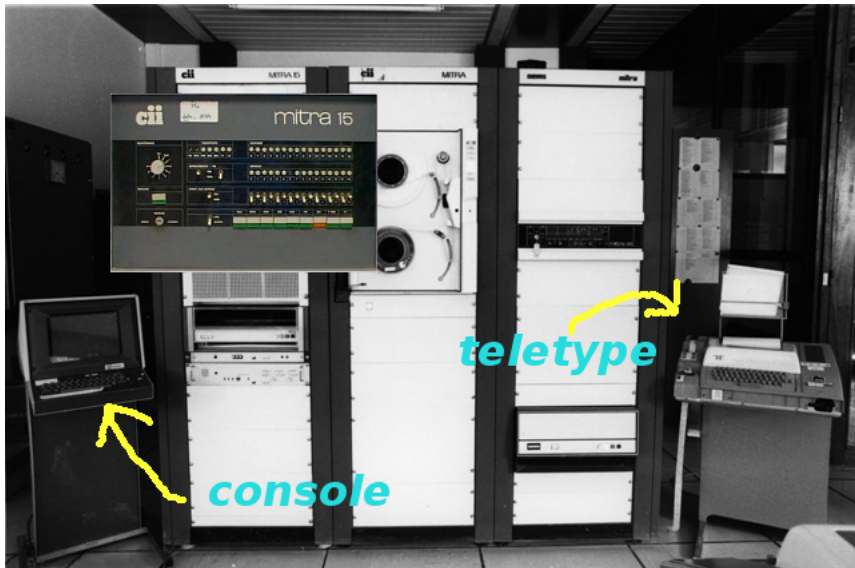
3 - login

- console
- login
- user, root
- connexion à distance
- tour de l'hôte
- mot de passe

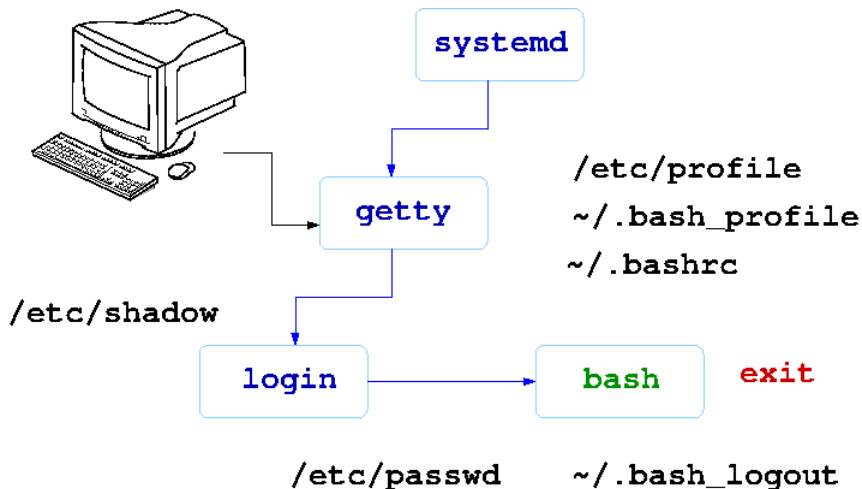
origine



MITRA 15



console réelle



login

`login` initialise les variables :

- HOME, SHELL (/etc/passwd)
- USER, LOGNAME (/etc/passwd)
- PATH (/usr/local/bin :/bin :/usr/bin)
- MAIL (répertoire)

avant l'action du **shell**.

pseudo terminal

```
~> tty
/dev/pts/2

~> fuser -av /dev/pts/2
UTIL. PID ACCES COMMANDE
/dev/pts/2: pl 1567 F.... xfce4-terminal
pl 3041 F.... bash

~> ps -oppid ,comm 3041
PPID COMMAND
1567 bash

~> ps -oppid ,comm 1567
PPID COMMAND
1 xfce4-terminal

~> ps -oppid ,comm 1
PPID COMMAND
0 systemd

~> ps -oppid ,comm 0
error: process ID out of range
```

root, user

L'administrateur **root** est habilité à créer les comptes utilisateurs **user** à partir de :

- nom de login
- nom complet

Un jeu de commande d'administration permet à **root** de paramétrer les comptes utilisateurs.

- unicité du super-utilisateur pour **unix**.
- **sudo** : permet de lancer une commande **as root**.

création d'un compte

```
# useradd toto -c 'Mr Toto' -g novice
# ls -al /home/toto/
total 28
drwx-----. 4 toto novice 4096
drwxr-xr-x. 9 root root 4096
-rw-r--r--. 1 toto novice 18 .bash_logout
-rw-r--r--. 1 toto novice 193 .bash_profile
-rw-r--r--. 1 toto novice 124 .bashrc
drwxr-xr-x. 2 toto novice 4096 .gnome2
```

- `.bash{,_logout, _profile, rc}` configuration du **shell**

VOIR AUSSI

`passwd`, `crypt(3)`, `userdel(8)`, `usermod(8)`.

.bash_profile

```
1 # .bash_profile
2 # Get the aliases and functions
3 if [ -f ~/.bashrc ]; then
4   ~/.bashrc
5 fi
6 # User specific environment and startup programs
7 export PATH=$PATH:$HOME/bin:$HOME/script
8 export TEXINPUTS=$TEXTINPUTS:$HOME/texinput
9 export HISTFILESIZE=2048
10 export HISTSIZE=512
11 export HISTFILE=~/.compil_history
12 export PS1="\t pl@\ W> "
```

.bashrc

```
# .bashrc
# Source global definitions
if [ -f /etc/bashrc ]; then
. /etc/bashrc
fi
set -o vi
alias sshnab='ssh langevin@naboo.univ-tln.fr '
alias sshgpu='ssh pl@imath-gpu.univ-tln.fr '
alias sshm1='ssh pl@serveur-imath01.univ-tln.fr '
alias sshm2='ssh pl@serveur-imath02.univ-tln.fr '
# etc ...
```

/etc/passwd

```
↪ ls -ald /home/t*
drwx-----. 4 titi novice /home/titi
drwx-----. 4 toto novice /home/toto
↪ tail -2 /etc/passwd
toto:x:1002:1002:Mr Toto:/home/toto:/bin/bash
titi:x:1003:1002:Mr Titi:/home/titi:/bin/bash
# passwd toto
# mail toto@ouest.fr -s compte << -FIN
> salut toto ,
> votre compte est active !
> bye
> -FIN
```

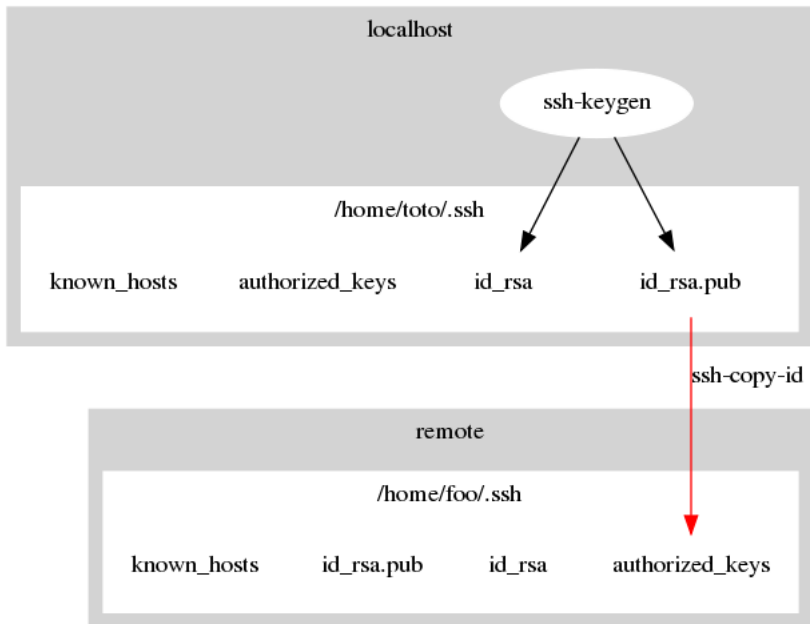
configuration ssh

```
[toto@ouest ~]$ ls .ssh/  
authorized_keys  known_hosts
```

```
[toto@ouest ~]$ ssh-keygen
```

```
[toto@ouest ~]$ ls .ssh/  
authorized_keys  id_rsa  id_rsa.pub  known_hosts
```

```
[toto@ouest ~]$ ssh-copy-id turing.euphoria.fr
```



configuration ssh

```
[toto@ouest ~]$ ssh turing.euphoria.nil  
[toto@turing ~]$ pstree -aulpG  
init,1  
!-- sshd,4197  
!    !  
!    |-- sshd,27781  
!        !  
!        |-- sshd,27783,toto  
!            !  
!            |-- bash,27784  
!                !  
!                |-- pstree,6718 -aulpG
```

commande distante

```
↪ ssh toto@turing.euphoria.nil
...
[toto@turing ~]$ pstree -aulpG
init,1
  !-- sshd,4197
  !
  !   !-- sshd,27781
  !       !
  !           !-- sshd,27783,toto
  !               !
  !                   !-- bash,27784
  !                       !
  !                           !-- pstree,6718 -aulpG
```

console virtuelle

```
~> ps -C mingetty -opid ,cmd
PID CMD
1291 /sbin/mingetty /dev/tty2
1295 /sbin/mingetty /dev/tty4
2108 /sbin/mingetty /dev/tty3
```

ALT-CTRL-F3, ouvre une **console**.

```
login: pl passwd: ——
[pl] ps -C mingetty
  PID TTY          TIME CMD
 1291 tty2        00:00:00 mingetty
 1295 tty4        00:00:00 mingetty

[pl] tty
/dev/tty3
```

shell de login

```
[pl] bash -c 'ps -f; sleep 5'
UID  PID  PPID  TTY  CMD
pl   2257  2158  tty3  -bash
pl   2317  2257  tty3  bash -c ps -f; sleep 5
pl   2318  2317  tty3  ps -f
```

- le '-' indique un **shell** de login.

getty

```
[pl] pstree -ap | grep -A7 -e '|-login' > tty.txt  
[pl] exit
```

ALT-CTRL-F1, ferme la **console**.

```
~> cat tty.txt  
|-login ,2108  
|   '-bash ,2143  
|       |-grep ,2176 -A7 -e |-login  
|           '-pstree ,2175 -ap  
|-mingetty ,1291 /dev/tty2  
|-mingetty ,1295 /dev/tty4
```

découverte de l'hôte

```
[toto@turing ~]$ uname -a  
Linux turing.euphoria.fr 2.6.18-53.1.4.el5 ...  
i386 GNU/Linux
```

```
[toto@turing ~]$ grep -c proc /proc/cpuinfo  
4
```

```
[toto@turing ~]$ lscpu  
Architecture:      i686  
CPU op-mode(s):    32-bit  
CPU(s):             2 Thread(s) par coeur : 2
```

découverte de l'hôte

~> uptime

```
18:00:55 up 14 days, 22:21, 1 user,
          load average: 1.37, 0.61, 0.33
```

~> free -g

```
      total    used  free  shared  buffers  cached
Mem:    47      30   16       0         0      29
-/+ buffers/cache:    1    46
Swap:      72       0    72
```

accès au réseau

```
[toto@turing ~]$ cat /etc/resolv.conf
search euphoria.fr
nameserver 10.1.65.1
[toto@turing ~]$ /sbin/ifconfig eth0
eth0 Link Ethernet HWaddr 00:1A:64:34:C3:FA
inet adr:10.1.65.72 Bcast:10.1.255.255
           Masque:255.255.0.0
adr inet6:fe80::21a:64ff:fe34:c3fa/64 ...
[toto@turing ~]$ /sbin/route
Table de routage IP du noyau
Destination  Passerelle  Genmask          I  face
10.1.0.0      *           255.255.0.0     U  eth0
default      blackjack   0.0.0.0         UG eth0
```


stockage

```
[toto@turing ~]$ df -h
Sys. de fich.  Tail. Occ. Disp. %Occ. Monte sur
/dev/mapper/rootvg-rootvol
              7,7G  6,6G 715M   91% /
/dev/mapper/lun_systemp1
              99M   28M   66M  30% /boot
tmpfs         1014M    0 1014M   0% /dev/shm
/dev/sda2     63G  1,2G   58G   2% /var/log
/dev/mapper/bd_vg-bd_lv
              764M  502M  263M 66% /bd
/dev/mapper/webdata_vg-webdata_lv
              90G   46G   45G 51% /webdata
/dev/mapper/web_vg-web_lv
              2,8G  1,8G 1015M 64% /www
```

who

```
[toto@turing ~]$ whoami
toto
[toto@turing ~]$ wc -l /etc/passwd
141 /etc/passwd
[toto@turing ~]$ w
 11:41:41 up 23 days, 16:02,  1 user,
           load average: 0,18, 0,23, 0,30
USER  TTY      FROM          LOGIN@      WHAT
toto  pts/0    sol83——    11:23      sshd: toto
[toto@turing ~]$ last -n3
toto    pts/0    euphoria Sep 10 11:23    still
michko pts/0    sol83——    Sep 10 11:03 - 11:15
nicolas pts/0    pc-nico  Sep  6 16:26 - 16:28
```

machine pour le calcul

```
[toto@imath ~]$ grep -c proc /proc/cpuinfo
24
[toto@imath ~]$ top
21:53:53 up 102 days, load average: 2.00, 2.00, 2.00
Tasks: 431 total, 3 running, 428 sleeping, 0 zombie
Cpu(s): 8.3%us, 0.0%sy, 0.0%ni, 91.7%id, 0.0%wa
Mem: 98986172k total, 73171416k used, 25814756k free
Swap: 76218360k total, 100900k used, 76117460k free

  PID USER   PR   NI  S %CPU %MEM    TIME+  COMMAND
14855 pv      25    0   R 100.0  0.0   10276:06 kpc
14856 pv      25    0   R  99.8  0.0   10263:15 kpc
   1 root    15    0   S  0.0  0.0    41:40.06 init
   2 root    RT   -5   S  0.0  0.0     0:00.24
   3 root    34   19   S  0.0  0.0     0:00.10
```

sécurisation des password

```
[toto@turing ~]$ who  
toto pts/0          2012-09-01 17:54 (ouest.net)
```

```
[toto@turing ~]$ tail -2 /etc/passwd  
toto:x:1002:1002:Mr Toto:/home/toto:/bin/bash  
titi:x:1003:1002:Mr Titi:/home/titi:/bin/bash
```

```
[toto@turing ~]$ cat /etc/shadow  
cat: /etc/shadow: Permission non accordée
```

/etc/passwd et /etc/shadow

```
~> ls -alt /etc/passwd
```

```
-rw-r--r--. 1 root root ... 17:13 /etc/passwd
```

```
~> ls -alt /etc/shadow
```

```
-----. 1 root root ... 17:13 /etc/shadow
```

```
[root@localhost ~]# tail -2 /etc/shadow
```

```
toto:$6$6gznwvTA$6cM.B8GJvIBYN0QRDnYRU6Px/QQCFoy  
bKhbl3Vr4q7XmycL.LYzldXu9eBtaBNTiOGzkBmwDXkwfAj9  
K4jaTT0:15584:0:99999:7: : :
```

```
titi:$6$QMF37bGZ$NfVulaWrmlCi/cuNaeM.UXHyO9ZS1yw  
Biel1LXaxOhAURnW.BwtAGcy6HEpuF93yQt.mxoJd3Dq23Zu  
KMkReA0:15584:0:99999:7: : :
```

permission

```
↪ ls -alt /etc/passwd  
-rw-r--r--. 1 root root ... 17:13 /etc/passwd
```

```
↪ ls -alt /etc/shadow  
-----. 1 root root ... 17:13 /etc/shadow
```

```
[root@localhost ~]# tail -2 /etc/shadow  
toto:$6$6gznwvTA$6cM.B8GJvIBYN0QRDnYRU6Px/QQCFoy  
bKhbl3Vr4q7XmycL.LYzldXu9eBtaBNTiOGzkBmwDXkwfAj9  
K4jaTT0:15584:0:99999:7: : :
```

```
titi:$6$QMF37bGZ$NfVulaWrmlCi/cuNaeM.UXHyO9ZS1yw  
Biel1LXaxOhAURnW.BwtAGcy6HEpuF93yQt.mxoJd3Dq23Zu  
KMkReA0:15584:0:99999:7: : :
```

man crypt

CRYPT(3) Manuel du programmeur Linux

NOM crypt – Chiffrage de données ou de mot de passe

SYNOPSIS

```
#define _XOPEN_SOURCE      #include <unistd.h>
char *crypt(const char *key, const char *salt);
édition des liens avec l'option -lcrypt.
```

DESCRIPTION

La fonction crypt est celle utilisée pour le cryptage des mots de passe.

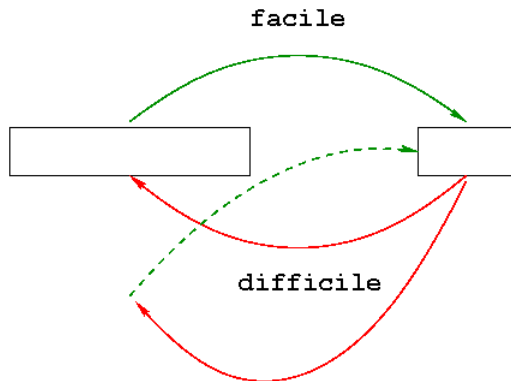
NOTES

un cryptogramme est de la forme

$\$id\$salt\$encrypted$:

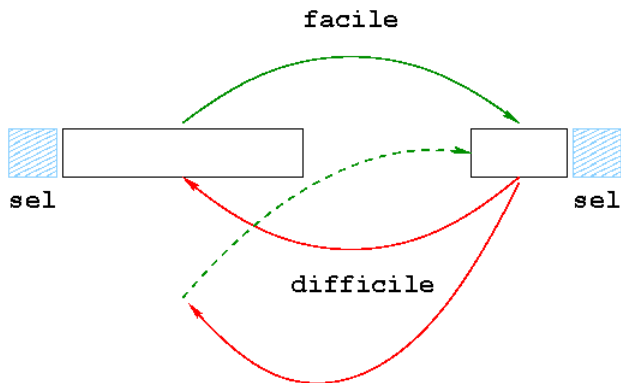
1 : MD5 ... 5 : SHA-256 , 6 : SHA-512

fonction à sens unique



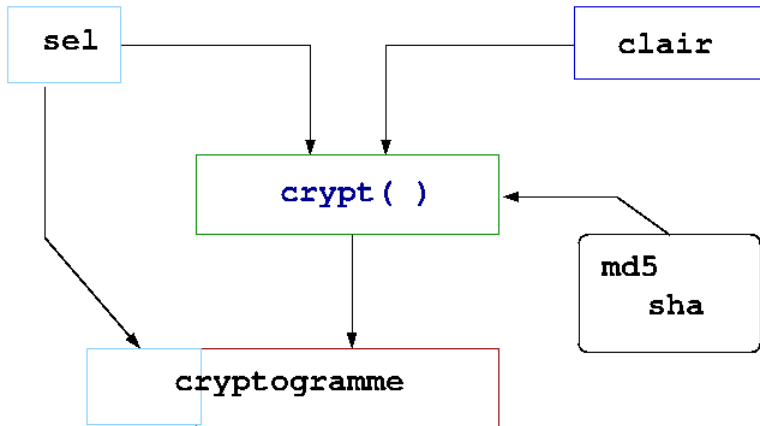
- Le système conserve les cryptogrammes (empreintes)
- L'utilisateur doit fournir le clair.

salage



- Le salage aléatoire renforce la sécurité du système.
- Le même passe conduit à des empreintes différentes.

hachage



décryptage

```
1 #define _XOPEN_SOURCE      /* See feature_test_macros(7)
   */
2 #include <unistd.h>
3 #include <stdio.h>
4 #include <string.h>
5 #include <stdlib.h>
6 int main (int argc, char* argv[] )
7 {
8     char *res;
9     res = crypt( argv[1], argv[2] ) ;
10    if ( strcmp( argv[2], res ))
11        return 1;
12
13    return 0;
14 }
```

décryptage

```
:: gcc -Wall pwd.c
/tmp/ccgsfZqg.o: In function 'main':
pwd.c:(.text+0x21): undefined reference to '
crypt '
collect2: ld a retourne 1 code d'etat d'
execution
:: gcc -Wall pwd.c -lcrypt
:: ./a.out toto '$6$6gznwvTA$6cM.
B8GJvlBYN0QRDnYRU
6Px/QQCFoybKhbl3Vr4q7XmycL.
LYzldXu9eBtaBNTiOGzkBmw
DXkwfAj9K4jaTT0 '

:: echo $?
0
```

cassage, charte

Une fois connu le mot de passe crypté, il reste possible de casser les mots de passe **faibles** par la force brute ...

- **/etc/shadow** est un anti-cracker !

[[John the Ripper](#)]

déontologie : Un utilisateur **unix** signe une charte qui stipule qu'un mot de passe est strictement personnel.