

# Unix et Programmation Shell

10 juin 2013

Vous répondrez aux questions en par deux lignes en moyenne.

**Q 1.** Citer deux informaticiens à l'origine de GNU/Linux.

```
grep -rE '([0-9]{1,3}\.){3}[0-9]+'  
--include=r*.*/etc 2>/dev/null
```

**Q 2.** Préciser trois acronymes du monde unix.

Dans la plupart des cas, cette commande écrira au moins une ligne. Pourquoi ?

**Q 3.** La commande `uname` permet d'obtenir des informations sur le système. Lesquelles ?

**Q 13.** Pour supprimer correctement les balises `html` du fichier `index.html`, quelle est la bonne commande ?

**Q 4.** Donner un exemple de système d'exploitation récent basé sur un noyau `linux`.

```
$ sed 's/<[^>]*>/' index.html
```

**Q 5.** Rappeler l'organisation générale du système de fichiers `unix`.

```
$ sed 's/<.*>/' index.html
```

**Q 6.** Quelle est la particularité du répertoire `/proc` ?

```
$ sed 's/<[^>]*>/g' index.html
```

**Q 7.** Décrire avec précision le rôle d'une ligne de `shell` commençant par le caractère `#`.

```
$ sed 's/<.*>/g' index.html
```

**Q 8.** Préciser 7 commandes internes populaires du `shell bash`.

**Q 14.** Commenter

```
[/tmp/ici]$  
echo 'cd $1;pwd'>cd;chmod u+x cd;  
[/tmp/ici]$ mkdir /tmp/labas  
[/tmp/ici]$ ./cd /tmp/labas  
/tmp/labas  
[/tmp/ici]$
```

**Q 9.** Détailler

```
$ ls -il /tmp/fb.sh  
48744 -rwxr-x-x 1 foo bar 35  
6 juin 10:23 /tmp/fb.sh
```

**Q 10.** Commenter avec précision le résultat de la commande ci-dessous :

```
$/usr/bin/time --format="cpu=%U" \  
time -p sleep 2  
real 2.00  
user 0.00  
sys 0.00  
cpu=0.00
```

**Q 11.** Proposer une explication

```
$ alias xp='if ! cat $((x++))  
2>/dev/null;then echo -n $x;fi'  
$ xp;xp;xp;xp  
12trois  
4
```

**Q 12.** Que fait la commande

**Q 15.** Dans l'exemple ci-dessous, sur quel hôte est exécuté la commande `who` ?

```
[]$ ssh pl@192.168.0.100 who  
pl@192.168.0.100's password: ***  
pl tty1 2013-06-06 07:57  
pl pts/1 2013-06-06 13:19  
pl pts/2 2013-06-06 13:05  
pat pts/3 2013-06-06 13:32
```

Rappeler les étapes à suivre pour exécuter des commandes sur une machine distante sans fournir de mot de passe.

**Q 16.** Ecrire un script : `whereis user hosts`, pour déterminer sur quel hôte est connecté un utilisateur donné.

- `user` : utilisateur
- `hosts` : fichier contenant la liste des hôtes accessibles par `ssh`.

Utiliser les commandes : `ping`, `ssh` et `who`.

```

1  execve("/bin/;-)
2  open("/dev/tty", ORDWR|ONONBLOCK)      = 3
3  open("/proc/meminfo", ORDONLY)        = 3
4  read(3, "MemTotal:      1798268 kB\nMemF"... , 1024) = 1024
5  clone(Process 23357 attached
6  child_stack=0, flags=CLONE_CHILD_CLEAR_TID|CLONE_CHILD_SETTID|SIGCHLD, child
7  [pid 23356] clone(Process 23358 attached
8  child_stack=0, flags=CLONE_CHILD_CLEAR_TID|CLONE_CHILD_SETTID|SIGCHLD, child
9  [pid 23357] dup2(4, 1)                  = 1
10 [pid 23358] dup2(3, 0)                  = 0
11 Process 23356 suspended
12 [pid 23357] open("/tmp/src", ORDONLY)   = 3
13 [pid 23357] dup2(3, 0)                  = 0
14 [pid 23357] execve("/bin/cat", ["cat"], [/* 53 vars */]) = 0
15 [pid 23358] open("/tmp/dst", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
16 [pid 23358] dup2(3, 1)                  = 1
17 [pid 23358] execve("/usr/bin/wc", ["wc", "-l"], [/* 53 vars */] <unfinished
18 [pid 23358] <... execve resumed> )      = 0
19 [pid 23357] read(0, "gnu posix Bash sed awk\n", 32768) = 23
20 [pid 23357] write(1, "gnu posix Bash sed awk\n", 23 <unfinished ... >
21 [pid 23357] <... write resumed> )       = 23
22 [pid 23358] <... open resumed> )        = 3
23 [pid 23357] read(0, "", 32768)         = 0
24 Process 23356 resumed
25 Process 23357 detached
26 [pid 23358] read(3, "# Locale name alias data base.\n#"..., 4096) = 2512
27 Process 23356 suspended
28 [pid 23358] read(3, "", 4096)           = 0
29 [pid 23358] read(0, "gnu posix Bash sed awk\n", 16384) = 23
30 [pid 23358] read(0, "", 16384)         = 0
31 [pid 23358] write(1, "1\n", 2)         = 2
32 Process 23356 resumed
33 Process 23358 detached
34 — SIGCHLD (Child exited) @ 0 (0) —

```

FIG. 1 – capture des appels systèmes.

**Q 17.** Le script (2) a été utilisé pour obtenir les traces (1).

- Commenter ce script.
- Quelle commande de débogage a été utilisée ?
- Donner un autre exemple de commande de débogage.
- Quelle commande a été tracée ?
- Représenter schématiquement les processus, fichiers et descripteurs de fichiers mis en jeu par la commande tracée.

```

flags=read , write , open , dup2 , execve , clone
strace -fe trace=$flags bash -c "$1" &> traces.all
grep -vE '(lib|share|ELF|\.so|-1)' traces.all \
  | sed 's/bash.*/;-)/' | cut -c1-75 > traces.txt

```

FIG. 2 – script de traçage.