Contents

1 Exercice n°1

```
1 echo $...
```

- $$: Process ID (PID) du script actuel
- $BASHPID : PID du bash actuel
- $PPID : PID du processus parent
- ! : (process ID) du dernier processus en background
- OLDPWD : chemin du dossier précédent
- PWD : chemin du dossier actuel
- HOME : chemin vers notre dossier home
- PATH : ensemble des chemins que le bash va regarder automatiquement
- ? : status de retour de la commande, fonction, ou script lui même
- PIPESTATUS : tableau conteanant les status de retour of the last executed foreground pipe.
- RANDOM : nbr entier positif aléatoire
- SRANDOM : pareil mais ne peut pas être reseemed
- IFS : champ séparateur
- PS1 : main prompt, du command-line.

- SHELL : le chemin du bash (/bin/bash)
- COLUMNS : Colonne du terminal
- LINES : lignes du terminal

```
1 echo "PID:" $$ "BASHPID:" $BASHPID "PPID:" $PPID ==> 5916 5916 3302
```

```
1 (echo "PID:" $$ "BASHPID" $BASHPID "PPID" $PPID)
2 PID: 5916 BASHPID 7034 PPID 3032
```

```
1 {_____ echo "PID:" $$ "BASHPID" $BASHPID "PPID" $PPID; _____}
2 PID: 5916 BASHPID 5916 PPID 3032
```

Pour voir le manuel de read est type, il faut regarder le man de builtins

1.1 voir ~/scripts/[readin/fctToFile/fctInsFile]

```
1  function fct() \n
2 {... ; ... ; done}
```

Puis type fct
Enfin pour tout mettre dans un fichier type fct | tail -n +2 > file
>;< ... = fichier

= commande

2 Exercice n°2

2.1 Question 1

```
1 printf "Je␣m'appelle␣%s.\n" $USER
```

Affiche notre nom d'utilisateur

2.2 Question 2

```
1   strace -- printf "Je␣m'appelle␣%s.\n" $USER
2
3
4 execve("/usr/bin/printf", ["printf", "Je␣m'appelle␣%s.\\n", "cliquot"],
      0x7ffdc488dc68 /* 53 vars */) = 0
5 brk(NULL)                                 = 0x55b5742ae000
6 arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd11cf63c0) = -1 EINVAL (Argument
      invalide)
7 access("/etc/ld.so.preload", R_OK)        = -1 ENOENT (Aucun fichier ou
     dossier de ce type)
8 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 5
9 fstat(5, {st_mode=S_IFREG|0644, st_size=106506, ...}) = 0
10 mmap(NULL, 106506, PROT_READ, MAP_PRIVATE, 5, 0) = 0x7f58afc1f000
11 close(5)                                 = 0
12 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC)
      = 5
```

```
read(5, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360A
    \2\0\0\0\0\0"..., 832) = 832
pread64(5, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@
    \0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(5, "\4\0\0\0\20\0\0\0\5\0\0\0GNU
    \0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(5, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\237\333t\347\262\27\3201
    \223\27*\202C\370T\177"..., 68, 880) = 68
fstat(5, {st_mode=S_IFREG|0755, st_size=2029560, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
    0) = 0x7f58afc1d000
pread64(5, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@
    \0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(5, "\4\0\0\0\20\0\0\0\5\0\0\0GNU
    \0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(5, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\237\333t\347\262\27\3201
    \223\27*\202C\370T\177"..., 68, 880) = 68
mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 5, 0) = 0
    x7f58afa2b000
mmap(0x7f58afa4d000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|
    MAP_FIXED|MAP_DENYWRITE, 5, 0x22000) = 0x7f58afa4d000
mmap(0x7f58afbc5000, 319488, PROT_READ, MAP_PRIVATE|MAP_FIXED|
    MAP_DENYWRITE, 5, 0x19a000) = 0x7f58afbc5000
mmap(0x7f58afc13000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED
    |MAP_DENYWRITE, 5, 0x1e7000) = 0x7f58afc13000
mmap(0x7f58afc19000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED
    |MAP_ANONYMOUS, -1, 0) = 0x7f58afc19000
close(5)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f58afc1e580) = 0
mprotect(0x7f58afc13000, 16384, PROT_READ) = 0
mprotect(0x55b57358a000, 4096, PROT_READ) = 0
mprotect(0x7f58afc67000, 4096, PROT_READ) = 0
munmap(0x7f58afc1f000, 106506)          = 0
brk(NULL)                               = 0x55b5742ae000
brk(0x55b5742cf000)                     = 0x55b5742cf000
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC)
    = 5
fstat(5, {st_mode=S_IFREG|0644, st_size=8308144, ...}) = 0
mmap(NULL, 8308144, PROT_READ, MAP_PRIVATE, 5, 0) = 0x7f58af23e000
close(5)                                = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
write(1, "Je␣m'appelle␣cliquot.\n", 22Je m'appelle␣cliquot.
)␣=␣22
close(1)␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣=␣0
close(2)␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣=␣0
exit_group(0)␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣=␣?
+++␣exited␣with␣0␣+++
```

On constate que les mots détectés sont : ["printf", "Je m'appelle %s.\", "cliquot"]

2.3 Question 3

avoir un | pour le stderr (== 2>) command 2>&1 ... | ...

L'appel systéme pour ouvrir des fichiers est : openat

On l'appelle 3 fois

```
command 2>&1 strace -- printf "Je m'appelle %s.\n" $USER | grep openat
```

```
  openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 5
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC)
    = 5
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC)
   = 5
```

D'après man openat, le file descriptor est le return de la fonction (-1 errno) Ici 5
Les fichiers ouverts sont :

- /etc/ld.so.cache
- /lib/x86$_{64}$-linux-gnu/libc.so.6
- /usr/lib/locale/locale-archive

Le mode d'ouverture est read-only (+ un flag que je comprends pas)

2.4 Question 4

```
command 2>&1 strace -- printf "Je m'appelle %s.\n" $USER | grep write
```

Un seul write

2.5 Question 5

Strace était sur le descripteur de fichier stderr
tee : command pour split ce qu'il reçoit en standard input en standard output

2.5.1 stderr + stdin

```
echo $(strace -- printf "Je m'appelle %s.\n" $USER)
```

2.5.2 eliminer sortie standard

```
echo $(strace -- printf "Je m'appelle %s.\n" $USER >/dev/null)
```

2.5.3 redirection stderr > stdout

```
echo -e "$(strace -- printf "Je m'appelle %s.\n" $USER 2>&1)"
```

2.6 Question 6

```
strace -e %file -- printf "Je m'appelle %s.\n" $USER
```

On obtient :

```
  execve("/usr/bin/printf", ["printf", "Je␣m'appelle␣%s.\\n", "cliquot"
   ], 0x7fff347673c8 /* 53 vars */) = 0
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Aucun fichier ou
   dossier de ce type)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 5
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC)
    = 5
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC)
   = 5
Je m'appelle␣cliquot.
```

Avec :

```
strace -e %file -- bash -c "cat␣./temp␣2>dev/null/"
```

On obtient en plus stat, et access:

```
  execve("/usr/bin/bash", ["bash", "-c", "cat␣./temp␣2>dev/null/"], 0
   x7ffe25b5fc48 /* 53 vars */) = 0
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Aucun fichier ou
   dossier de ce type)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 5
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libtinfo.so.6", O_RDONLY|
   O_CLOEXEC) = 5
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libdl.so.2", O_RDONLY|O_CLOEXEC
   ) = 5
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC)
    = 5
openat(AT_FDCWD, "/dev/tty", O_RDWR|O_NONBLOCK) = 5
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC)
   = 5
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache",
    O_RDONLY) = 5
stat("/home/cliquot/ZZ1/syExp/tp1", {st_mode=S_IFDIR|0775, st_size
   =4096, ...}) = 0
stat(".", {st_mode=S_IFDIR|0775, st_size=4096, ...}) = 0
stat("/home", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
stat("/home/cliquot", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
stat("/home/cliquot/ZZ1", {st_mode=S_IFDIR|0775, st_size=4096, ...}) =
   0
stat("/home/cliquot/ZZ1/syExp", {st_mode=S_IFDIR|0775, st_size=4096,
   ...}) = 0
stat("/home/cliquot/ZZ1/syExp/tp1", {st_mode=S_IFDIR|0775, st_size
   =4096, ...}) = 0
stat("/home/cliquot/scripts", {st_mode=S_IFDIR|0775, st_size=4096,
   ...}) = 0
stat(".", {st_mode=S_IFDIR|0775, st_size=4096, ...}) = 0
stat("/home/cliquot/scripts/bash", 0x7ffd36655a30) = -1 ENOENT (Aucun
   fichier ou dossier de ce type)
stat("/home/cliquot/scripts/bash", 0x7ffd36655a30) = -1 ENOENT (Aucun
   fichier ou dossier de ce type)
stat("/home/cliquot/scripts/bash", 0x7ffd36655a30) = -1 ENOENT (Aucun
   fichier ou dossier de ce type)
stat("/home/cliquot/scripts/bash", 0x7ffd36655a30) = -1 ENOENT (Aucun
   fichier ou dossier de ce type)
```

```
23 stat("/home/cliquot/scripts/bash", 0x7ffd36655a30) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
24 stat("/home/cliquot/scripts/bash", 0x7ffd36655a30) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
25 stat("/home/cliquot/scripts/bash", 0x7ffd36655a30) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
26 stat("/home/cliquot/.local/bin/bash", 0x7ffd36655a30) = -1 ENOENT (
      Aucun fichier ou dossier de ce type)
27 stat("/usr/local/sbin/bash", 0x7ffd36655a30) = -1 ENOENT (Aucun fichier
       ou dossier de ce type)
28 stat("/usr/local/bin/bash", 0x7ffd36655a30) = -1 ENOENT (Aucun fichier
      ou dossier de ce type)
29 stat("/usr/sbin/bash", 0x7ffd36655a30)  = -1 ENOENT (Aucun fichier ou
      dossier de ce type)
30 stat("/usr/bin/bash", {st_mode=S_IFREG|0755, st_size=1183448, ...}) = 0
31 stat("/usr/bin/bash", {st_mode=S_IFREG|0755, st_size=1183448, ...}) = 0
32 access("/usr/bin/bash", X_OK)           = 0
33 stat("/usr/bin/bash", {st_mode=S_IFREG|0755, st_size=1183448, ...}) = 0
34 access("/usr/bin/bash", R_OK)           = 0
35 stat("/usr/bin/bash", {st_mode=S_IFREG|0755, st_size=1183448, ...}) = 0
36 stat("/usr/bin/bash", {st_mode=S_IFREG|0755, st_size=1183448, ...}) = 0
37 access("/usr/bin/bash", X_OK)           = 0
38 stat("/usr/bin/bash", {st_mode=S_IFREG|0755, st_size=1183448, ...}) = 0
39 access("/usr/bin/bash", R_OK)           = 0
40 stat(".", {st_mode=S_IFDIR|0775, st_size=4096, ...}) = 0
41 stat("/home/cliquot/scripts/cat", 0x7ffd36655920) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
42 stat("/home/cliquot/scripts/cat", 0x7ffd36655920) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
43 stat("/home/cliquot/scripts/cat", 0x7ffd36655920) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
44 stat("/home/cliquot/scripts/cat", 0x7ffd36655920) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
45 stat("/home/cliquot/scripts/cat", 0x7ffd36655920) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
46 stat("/home/cliquot/scripts/cat", 0x7ffd36655920) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
47 stat("/home/cliquot/scripts/cat", 0x7ffd36655920) = -1 ENOENT (Aucun
      fichier ou dossier de ce type)
48 stat("/home/cliquot/.local/bin/cat", 0x7ffd36655920) = -1 ENOENT (Aucun
       fichier ou dossier de ce type)
49 stat("/usr/local/sbin/cat", 0x7ffd36655920) = -1 ENOENT (Aucun fichier
      ou dossier de ce type)
50 stat("/usr/local/bin/cat", 0x7ffd36655920) = -1 ENOENT (Aucun fichier
      ou dossier de ce type)
51 stat("/usr/sbin/cat", 0x7ffd36655920)   = -1 ENOENT (Aucun fichier ou
      dossier de ce type)
52 stat("/usr/bin/cat", {st_mode=S_IFREG|0755, st_size=43416, ...}) = 0
53 stat("/usr/bin/cat", {st_mode=S_IFREG|0755, st_size=43416, ...}) = 0
54 access("/usr/bin/cat", X_OK)            = 0
55 stat("/usr/bin/cat", {st_mode=S_IFREG|0755, st_size=43416, ...}) = 0
56 access("/usr/bin/cat", R_OK)            = 0
57 stat("/usr/bin/cat", {st_mode=S_IFREG|0755, st_size=43416, ...}) = 0
```

```
stat("/usr/bin/cat", {st_mode=S_IFREG|0755, st_size=43416, ...}) = 0
access("/usr/bin/cat", X_OK)            = 0
stat("/usr/bin/cat", {st_mode=S_IFREG|0755, st_size=43416, ...}) = 0
access("/usr/bin/cat", R_OK)            = 0
bash: dev/null/: Aucun fichier ou dossier de ce type
stat("/home/cliquot/.terminfo", 0x55dc92267550) = -1 ENOENT (Aucun
   fichier ou dossier de ce type)
stat("/etc/terminfo", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
stat("/lib/terminfo", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
stat("/usr/share/terminfo", {st_mode=S_IFDIR|0755, st_size=4096, ...})
   = 0
access("/etc/terminfo/x/xterm-256color", R_OK) = -1 ENOENT (Aucun
   fichier ou dossier de ce type)
access("/lib/terminfo/x/xterm-256color", R_OK) = 0
openat(AT_FDCWD, "/lib/terminfo/x/xterm-256color", O_RDONLY) = 5
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=38458, si_uid
   =1000, si_status=1, si_utime=0, si_stime=0} ---
+++ exited with 1 +++
```

3 Exercice n°3

3.1 Question 1

L'option qui va nous permettre de tracer les appels systèmes est : -e %process

Pour tracer les processus fils va être l'option -f

On va chercher à tracer les commandes fork() et exec()

Exemple :

```
(strace -e %process -f -- emacs &) 2>&1 | grep 'fork\|exec'
```

```
  xecve("/usr/bin/emacs", ["emacs"], 0x7ffe1095f6d0 /* 53 vars */) = 0
[pid 39504] vfork(strace: Process 39509 attached
[pid 39509] execve("/usr/bin/gpg", ["/usr/bin/gpg", "--with-colons", "
   --list-config"], 0x7ffc05bd9090 /* 53 vars */ <unfinished ...>
[pid 39504] <... vfork resumed>)        = 39509
[pid 39509] <... execve resumed>)       = 0
[pid 39504] vfork(strace: Process 39512 attached
[pid 39512] execve("/usr/bin/emacsclient.emacs", ["/usr/bin/emacsclient
   .emacs", "--version"], 0x7ffc05bd7540 /* 53 vars */ <unfinished ...>
[pid 39504] <... vfork resumed>)        = 39512
[pid 39512] <... execve resumed>)       = 0
[pid 39504] vfork(strace: Process 39513 attached
[pid 39513] execve("/usr/bin/git", ["/usr/bin/git", "config", "--get-
   all", "credential.helper"], 0x7ffc05bd6c20 /* 54 vars */ <unfinished
    ...>
[pid 39504] <... vfork resumed>)        = 39513
[pid 39513] <... execve resumed>)       = 0
[pid 39504] vfork(strace: Process 39514 attached
[pid 39514] execve("/usr/bin/git", ["/usr/bin/git", "rev-parse", "--is-
   inside-work-tree"], 0x7ffc05bd7da0 /* 53 vars */ <unfinished ...>
[pid 39504] <... vfork resumed>)        = 39514
[pid 39514] <... execve resumed>)       = 0
[pid 39504] vfork(strace: Process 39515 attached
```

```
19 [pid 39515] execve("/usr/bin/git", ["/usr/bin/git", "--no-pager", "--
     literal-pathspecs", "-c", "core.preloadindex=true", "-c", "log.
     showSignature=false", "-c", "color.ui=false", "-c", "color.diff=
     false", "rev-parse", "--is-inside-work-tree"], 0x7ffc05bd79f0 /* 54
     vars */ <unfinished ...>
20 [pid 39504] <... vfork resumed>)        = 39515
21 [pid 39515] <... execve resumed>)       = 0
22 [pid 39504] vfork(strace: Process 39516 attached
23 [pid 39516] execve("/usr/bin/git", ["/usr/bin/git", "rev-parse", "--is-
     inside-work-tree"], 0x7ffc05bd74c0 /* 53 vars */ <unfinished ...>
24 [pid 39504] <... vfork resumed>)        = 39516
25 [pid 39516] <... execve resumed>)       = 0
26 [pid 39504] vfork(strace: Process 39517 attached
27 [pid 39517] execve("/usr/bin/git", ["/usr/bin/git", "--no-pager", "--
     literal-pathspecs", "-c", "core.preloadindex=true", "-c", "log.
     showSignature=false", "-c", "color.ui=false", "-c", "color.diff=
     false", "rev-parse", "--is-inside-work-tree"], 0x7ffc05bd7110 /* 54
     vars */ <unfinished ...>
28 [pid 39504] <... vfork resumed>)        = 39517
29 [pid 39517] <... execve resumed>)       = 0
30 [pid 39504] vfork(strace: Process 39518 attached
31 [pid 39518] execve("/usr/bin/git", ["/usr/bin/git", "version"], 0
     x7ffc05bd9f10 /* 54 vars */ <unfinished ...>
32 [pid 39504] <... vfork resumed>)        = 39518
33 [pid 39518] <... execve resumed>)       = 0
34 [pid 39504] vfork(strace: Process 39519 attached
35 [pid 39519] execve("/usr/bin/hunspell", ["/usr/bin/hunspell", "-vv"], 0
     x7ffc05bd9900 /* 53 vars */ <unfinished ...>
36 [pid 39504] <... vfork resumed>)        = 39519
37 [pid 39519] <... execve resumed>)       = 0
38 [pid 39504] vfork(strace: Process 39520 attached
39 [pid 39520] execve("/usr/bin/hunspell", ["/usr/bin/hunspell", "-vv"], 0
     x7ffc05bd99e0 /* 53 vars */ <unfinished ...>
40 [pid 39504] <... vfork resumed>)        = 39520
41 [pid 39520] <... execve resumed>)       = 0
42 [pid 39504] vfork(strace: Process 39521 attached
43 [pid 39521] execve("/usr/bin/hunspell", ["/usr/bin/hunspell", "-D", "-a
     ", "/dev/null"], 0x7ffc05bd9930 /* 53 vars */ <unfinished ...>
44 [pid 39504] <... vfork resumed>)        = 39521
45 [pid 39521] <... execve resumed>)       = 0
46 [pid 39504] vfork(strace: Process 39522 attached
47 [pid 39522] execve("/usr/bin/hunspell", ["/usr/bin/hunspell", "-vv"], 0
     x7ffc05bd9400 /* 53 vars */ <unfinished ...>
48 [pid 39504] <... vfork resumed>)        = 39522
49 [pid 39522] <... execve resumed>)       = 0
50 [pid 39504] vfork(strace: Process 39523 attached
51 [pid 39523] execve("/usr/bin/hunspell", ["/usr/bin/hunspell", "-a", "",
      "-d", "fr_FR", "-i", "UTF-8"], 0x7ffc05be9a50 /* 53 vars */ <
     unfinished ...>
52 [pid 39504] <... vfork resumed>)        = 39523
53 [pid 39523] <... execve resumed>)       = 0
54 [pid 39504] vfork(strace: Process 39524 attached
55 [pid 39524] execve("/usr/bin/git", ["/usr/bin/git", "--no-pager", "--
```

```
   literal-pathspecs", "-c", "core.preloadindex=true", "-c", "log.
   showSignature=false", "-c", "color.ui=false", "-c", "color.diff=
   false", "rev-parse", "--show-toplevel"], 0x7ffc05bd82f0 /* 54 vars
   */ <unfinished ...>
[pid 39504] <... vfork resumed>)        = 39524
[pid 39524] <... execve resumed>)       = 0
[pid 39504] vfork(strace: Process 39525 attached
[pid 39525] execve("/usr/bin/git", ["/usr/bin/git", "--no-pager", "--
   literal-pathspecs", "-c", "core.preloadindex=true", "-c", "log.
   showSignature=false", "-c", "color.ui=false", "-c", "color.diff=
   false", "rev-parse", "--git-dir"], 0x7ffc05bd82f0 /* 54 vars */ <
   unfinished ...>
[pid 39504] <... vfork resumed>)        = 39525
[pid 39525] <... execve resumed>)       = 0
[pid 39504] vfork(strace: Process 39526 attached
[pid 39526] execve("/usr/bin/git", ["/usr/bin/git", "rev-parse", "--is-
   inside-work-tree"], 0x7ffc05bd8790 /* 53 vars */ <unfinished ...>
[pid 39504] <... vfork resumed>)        = 39526
[pid 39526] <... execve resumed>)       = 0
[pid 39504] vfork(strace: Process 39527 attached
[pid 39527] execve("/usr/bin/git", ["/usr/bin/git", "--no-pager", "--
   literal-pathspecs", "-c", "core.preloadindex=true", "-c", "log.
   showSignature=false", "-c", "color.ui=false", "-c", "color.diff=
   false", "rev-parse", "--is-inside-work-tree"], 0x7ffc05bd83e0 /* 54
   vars */ <unfinished ...>
[pid 39504] <... vfork resumed>)        = 39527
[pid 39527] <... execve resumed>)       = 0
```

4 Exercice 4

4.0.1 redirection fichier

- com > fic redirige la sortie standard de com dans le fichier fic,
- com 2> fic redirige la sortie des erreurs de com dans le fichier fic,
- com 2>&1 redirige la sortie des erreurs de com vers la sortie standard de com,
- com < fic redirige l'entrée standard de com dans le fichier fic,
- com1 | com2 redirige la sortie standard de la commande com1 vers l'entrée standard de com2.
- com1 |& com2 branche ("connecte" selon le manuel bash) la sortie standard et la sortie d'erreur de com1 sur l'entrée de com2
- Les simples quotes délimitent une chaîne de caractères. Même si cette chaîne
- contient des commandes ou des variables shell, celles-ci ne seront pas
- interprétées.
- Les doubles quotes délimitent une chaîne de caractères, mais les noms de
- variable sont interprétés par le shell.
- Bash considère que les Back-quotes délimitent une commande à exécuter. Les
- noms de variable et les commandes sont donc interprétés.

4.0.2 comparateur

- n1 -eq n2, vrai si n1 et n2 sont égaux (equal) ;
- n1 -ne n2, vrai si n1 et n2 sont différents (non equal);
- n1 -lt n2, vrai si n1 est strictement inférieur à n2 (lower than);
- n1 -le n2, vrai si n1 est inférieur ou égal à n2 (lower or equal);
- n1 -gt n2, vrai si n1 est strictement supérieur à n2 (greater than) ;
- n1 -ge n2, vrai si n1 est supérieur ou égal à n2 (greater or equal).
- ! e, vrai si e est fausse ;
- e1 -a e2, vrai si e1 et e2 sont vraies ;
- e1 -o e2, vrai si e1 ou e2 est vraie.
- 0 est un succès, le reste est un échec.

[expression] == test expression

4.0.3 Boucle et structure conditionnelles

```
if condition1
    then instruction(s)
elif condition2
    then instruction(s)
elif condition3
    ...
fi
```

```
  case valeur_testee in
      valeur1) instruction(s);;
      valeur2) instruction(s);;
      valeur3) instruction(s);;
      ) instruction_else(s);;
esac
```

```
for variable in liste_valeurs
  do instruction(s)
done
```

```
for i in "$@"
do
    echo "$i"
done
```

```
while condition
do
    instruction(s)
done
```

```
a=1
a=$(($a + 1))
echo $a

a=1
let "a=$a␣+␣1"
echo $a

a=1
a=$(echo "$a+1" |bc )
echo $a

declare -i name=...
```

5 Exercice 5 (Coprocesses)

The standard output of command is connected via a pipe to a file descriptor the executing shell, and that file descriptor is assigned to NAME[0]. The standard input of command is connected via a pipe to a file descriptor in the executing shell, and that file descriptor is assigned to NAME[1]. (man Bash coprocesses)

- < lecture
- <> lect et ecriture
- » append
- > écriture

>name = nom de fichier >&num = descripteur de fichier num>- ferme le descripteur
ARRAY : $+TAB$ = *premirecase*+{TAB[n]} = contenu de la case n° n. +${TAB[@]} = tout le contenu.

```
echo "bonjour" >&${COPROC[1]}
echo <&${COPROC[0]}
```

Pour créer le signal, il n'exite pas de symbole EOF, mais on peut fermer le descripteur de fichier pour simuler le EOF.

```
eval exec ${COPROC[1]}">&-"
```

On a: exec qui va permettre de considérer la redirection comme une commande. eval qui va remplacer les variables AVANT d'exécuter. le >&- entre guillemet pour éviter que shell le remplace