

Exercice n° 1

Écrire un programme qui lance la commande passée en argument (avec une option). Par exemple, lance `ls -l` exécute `ls -l`. (On suppose par ailleurs que les commandes se situent dans le répertoire `/bin/`.)

Exercice n° 2

Expliquez pas-à-pas comment s'exécute ce programme.

```
#include<stdio.h>
#include<unistd.h>

main ()
{
  int x;
  x=0;
  pid_t pidfils;
  pidfils=fork();
  if (pidfils==0)
    {
      x=1;
    }
  else
    printf('x=%d\n',x);
}
```

Exercice n° 3

Pour chacun des programmes suivants expliquez combien de *hello!* sont affichés à l'écran. Donner également le nombre de processus créés et dessiner l'arbre généalogique.

1.

```
int main(){
  fork();
  printf('hello!\n');
  exit(0);}
```
2.

```
int main (){
  fork();
  fork();
  printf('hello!\n');
  exit(0);}
```

```

3. int main () {
    fork() ;
    fork() ;
    fork() ;
    printf(‘‘hello!\n’’) ;
    exit(0) ;}

```

Exercice n° 4

Écrire un programme qui lance un fils. Le fils doit afficher “ *C’est le fils qui parle* ” puis “ *Mon pid est ...* ” (les points de suspension doivent évidemment être remplacés par le pid du fils), et enfin “ *Le pid de mon père est ...* ” (les points de suspension doivent être remplacés par le pid du père). Par ailleurs, le père, lui, affiche successivement “ *C’est le père qui parle* ”, “ *Le pid de mon fils est ...* ”, “ *Mon pid est ...* ” et enfin “ *Le pid du grand-père de mon fils, donc mon père, est ...* ”.

Exercice n° 5

Écrire un programme qui crée deux fils (fils1 et fils2). Le père doit afficher “ *je suis le père* ”, fils1 doit afficher “ *je suis fils1* ” suivi de son pid, et idem pour fils2.

Exercice n° 6

Combien de *hello!* imprime chacun des programmes? Combien de processus sont créés? Donner l’arbre généalogique.

```

1. int main {
    int i ;
    for (i=0 ;i<2 ;i++)
        fork() ;
    printf(‘‘hello!\n’’) ;
    exit(0) ;}

```

```

2. void process() {
    fork() ;
    fork() ;
    printf(‘‘hello!\n’’) ;}

```

```

int main () {
    process() ;
    printf(‘‘hello!\n’’) ;
    exit(0) ;}

```

```

3. int main () {
    if(fork())
        fork() ;
    printf(‘‘hello!\n’’) ;
    exit(0) ; }

```

```
4. int main () {
    if (fork()==0) {
        if (fork()) {
            printf("hello!\n"); }
        }
    }
```