TP2 : Processus

Système M1

semaine du 3 octobre

1 Calcul en parallèle

Question 1. Écrire une fonction `find(int *tab, int length, int val)` qui cherche la valeur `val` dans un tableau `tab` de longueur `length`. Cette fonction retournera l'un des indices de `val` si cette valeur est dans le tableau, ou `-1` sinon.

Question 2. Écrire une fonction `find_2p` qui fait la même chose en utilisant deux processus : le père recherche dans la première partie du tableau, le fils dans la seconde.

Question 3. On suspend (avec *Ctrl-Z*) le processus père avant qu'il ne récupère la valeur de son fils. Qu'affiche `ps` ?

2 Indéterminisme lors d'accès concurrents aux fichiers

Question 4. Lancez le programme ci-dessous sur gros fichier (quelques KB), qui contient par exemple "`01234678901234567890123456789`..."

```
int main(int argc, char *argv[]) {
  char c[2];
  int f = open(argv[1], O_RDONLY);

  if (fork()) {
    c[0] = 'a';
    while (read(f, c + 1, 1)) {
      write(STDOUT_FILENO, c, 2);
    }
  }
  else {
    c[0] = 'b';
    while (read(f, c + 1, 1)) {
      write(STDOUT_FILENO, c, 2);
    }
  }

  close(f);

  return 0;
}
```

Question 5. Théoriquement, quels sont tous les résultats qui peuvent apparaître lorsque le fichier contient "`012`" ?

Question 6. Modifiez le programme et imposez une pause aléatoire avant/après chaque lecture ou écriture en utilisant `usleep(rand()% 1000000)`.

Question 7. Où et comment utiliser `srand()` pour qu'aucun des temps de pause ne soient corrélés ?

Question 8. Essayez d'obtenir grâce à ces modifications chacun des résultats théoriquement observables pour le programme original.

Question 9. Pourquoi a-t-on utilisé `write` et pas `printf` ?

3 Plus de processus

Question 10. Écrivez un programme qui lance 26 processus. Chacun sera nommé par une lettre de l'alphabet ('a', ..., 'z') et écrira son nom sur la sortie standard toutes les 3 secondes environ.

Question 11. Lancez votre programme et tuez les processus un par un.

Question 12. Modifiez votre programme pour que l'arbre des 26 processus (`ps f`) forme un peigne.