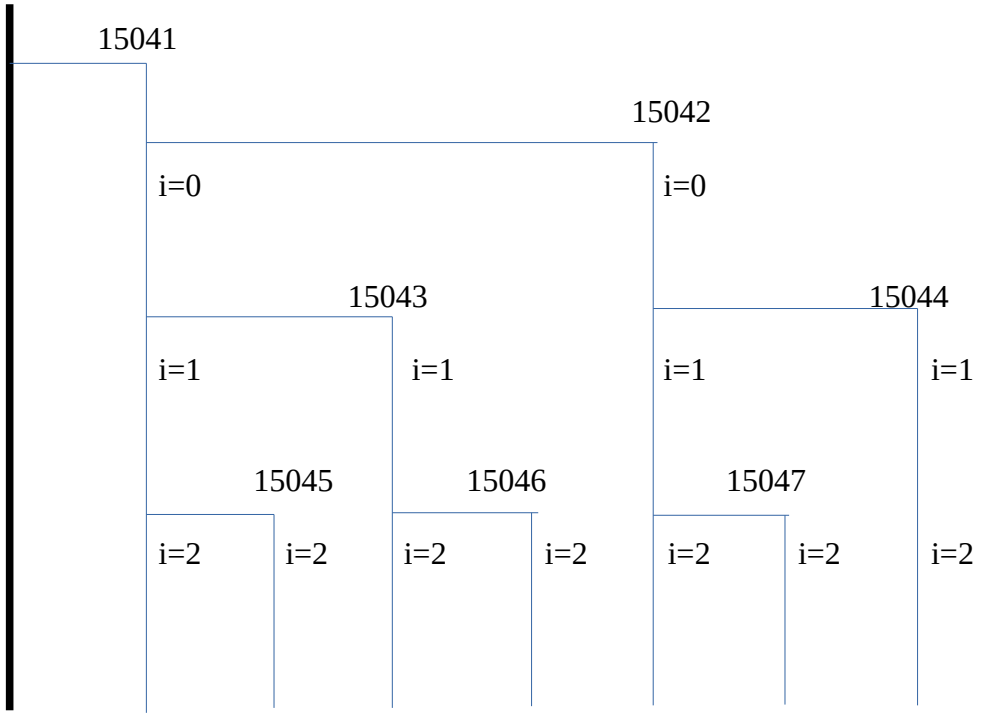
Correction exos 1 et 2 du TP 3

Exercice 1

1) La hiérarchie des processus

shell
pid = 13368



2) 8 processus ont été créés au total

Exercice 2

```
int main(int argc, char** argv) {

      int retourFork = fork();
      int fd = open("toto.txt", O_RDWR | O_CREAT , 0644);
      if (retourFork > 0) {   //père

            char* phrase = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
            affiche(fd, phrase, strlen(phrase));
            int retour;
      } else {   //fils

            char phrase[] = "88888888888888888888888888888888888888888888888";
            affiche(fd, phrase, strlen(phrase));

      }
      return 0;
}
```

Un contenu possible du fichier après exécution :

```
aaaaaaaaa8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8a8
a8a8a88888888888
```

Explication : la table des descripteurs de fichiers est copiée, donc fd pointe vers la même entrée de la "table des fichiers ouverts" (TFO) qui, elle, est commune à tous les processus. Dans l'entrée correspondante de la TFO, il y a, entre autres, le chemin du fichier ouvert, le mode d'ouverture, ainsi que l'"offset" (la tête de lecture). Ainsi le père et le fils ont chacun un descripteur de fichier qui pointe vers le même fichier avec la même tête de lecture. Ainsi, ils n'écrasent jamais ce que l'autre à fait.

Dans mon exécution, le système a donné la main au père au début (d'où la suite de "a"), puis ensuite il y a eu alternance entre le père et le fils (d'où la suite de "a8a8a8"), puis ensuite le fils a eu la main pour terminer (d'où la suite de "8").
Ici, tout aurait pu être possible, à condition que tous les "a" et tous les "8" soient écrits. Exemple :
8a8a8a