

# Programmation Système

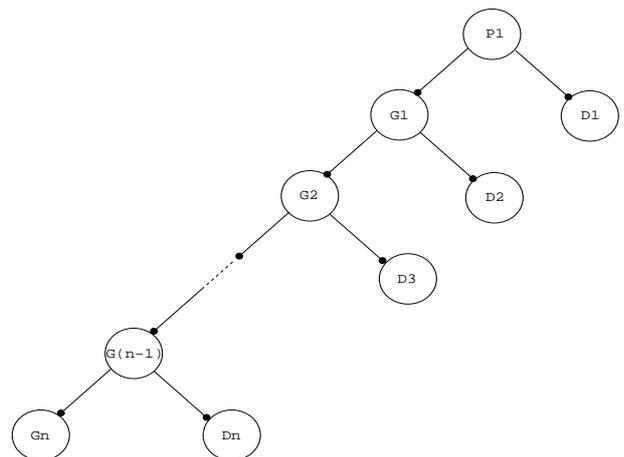
## TP noté (1h15)

### Exercice 1 - Commande wc -c (15')

Écrire un programme C qui se comporte comme la commande `wc -c` : Le programme ouvre un fichier de caractères dont le nom est donné en argument sur la ligne de commande, compte le nombre de caractères du fichier et affiche ce nombre. Les caractères du fichier doivent être lus par paquet de 10.

### Exercice 2 - Arborescence de processus (20')

Écrire un programme C qui prend un entier  $n \geq 1$  en argument sur la ligne de commande et réalise l'arborescence de processus de la figure ci-contre : P1 est le père de G1 (fils gauche) et D1 (fils droit), G1 est le père de G2 et D2, etc. Chaque processus fils doit afficher son pid ainsi que celui de son père (cf. exemple d'exécution). Chaque processus père doit attendre la terminaison de ses fils avant de terminer.



Exemple d'exécution :

```
$ ./a.out 3
Je suis P1, mon pid est 20209
Je suis G1 (fils gauche), mon pid est 20210, celui de mon pere est 20209
Je suis D1 (fils droit), mon pid est 20211, celui de mon pere est 20209
Je suis G2 (fils gauche), mon pid est 20212, celui de mon pere est 20210
Je suis D2 (fils droit), mon pid est 20213, celui de mon pere est 20210
Je suis G3 (fils gauche), mon pid est 20214, celui de mon pere est 20212
Je suis D3 (fils droit), mon pid est 20215, celui de mon pere est 20212
$
```

### Exercice 3 - Communication en triangle (40')

Écrire un programme C qui implémente le schéma de la figure ci-contre : le processus père crée 2 processus fils (fils1 et fils2) et 3 tubes de communication (tube1, tube2 et tube3). Le processus père commence par envoyer l'entier 1 au processus fils1 via le tube1. Le processus fils1 lui ajoute 1 et transmet le résultat (2) au processus fils2 via le tube2. Le processus fils2 lui ajoute 1 et envoie le résultat (3) au processus père via le tube3. Le processus père lui ajoute 1 et envoie le résultat (4) au processus fils1 via le tube1 et ainsi de suite... Le programme se termine lorsque le processus père reçoit un entier  $> 100$ . Le processus père affiche alors cet entier et attend la terminaison de ces fils avant de terminer.

