

Ressource R3.05 - TD 5

Tubes, redirections

1. Écrire un programme qui fait la même chose que le bash avec
`ls | wc >> output.txt`
2. Expliquez précisément le fonctionnement du programme suivant :

```
1  // include
2  void pere(int fdin){
3      char c;
4      while(1){
5          ssize_t n=read(fdin,&c,1);
6          if (n<=0) break;
7          if (islower(c)) c-=32;
8          write(STDOUT_FILENO,&c,1);
9      }
10     close(fdin);
11 }
12 void fils(int fdout){
13     char c;
14     while(1){
15         ssize_t n=read(STDIN_FILENO,&c,1);
16         if (n<=0) break;
17         write(fdout,&c,1);
18     }
19     close(fdout);
20     exit(0);
21 }
22 int main (int argc,char * argv[]){
23     pid_t pid;
24     int p[2];
25     assert(pipe(p) == 0);
26
27     pid=fork();
28     assert(pid >=0);
29
30     if (pid == 0) {
31         close(p[0]);
32         fils(p[1]);
```

```

33     }else {
34         close(p[1]);
35         pere(p[0]);
36     }
37     wait(NULL);
38     exit(0);
39 }

```

3. On se propose de calculer les nombres premiers inférieurs ou égaux à une valeur N , passée à la ligne de commande. Pour ce faire, on utilise le crible d'Ératosthène. Il s'agit de mettre en place la situation suivante :

```

          -----
2,...,N -> ( )      ( ) --> P
          -----

```

P reçoit 2, qui est donc premier, et crée $F2$

```

          -----          -----
2,...,N -> ( )      ( ) --> F2 --> ( )      ( ) -> P
          -----          -----

```

$F2$ filtre tous les multiples de 2.

P reçoit 3, qui est donc premier, et crée $F3$

```

          -----          -----          -----
2,...,N -> ( )      ( ) --> F2 --> ( )      ( ) -> F3 -> ( )      ( ) -> P
          -----          -----          -----

```

Ainsi de suite

- (a) Le processus initiale P crée un fils relié par un tube qui y écrit tous les entiers de $2, \dots, N$.
- (b) Quand un entier parvient jusqu'à P , il est premier. P crée un nouveau filtre.