

## Третье задание по практикуму для самостоятельного программирования дома

Напишите программу, строящую дерево процессов, эквивалентное дереву каталогов, рассматриваемому начиная от текущей директории. Так, например, если имеется следующее дерево каталогов:

```
.  
./a  
./b  
./a/c  
./a/d  
./a/e
```

то основной процесс, соответствующий текущей директории, должен породить двух потомков, соответствующих директориям `./a` и `./b`, первый из которых, в свою очередь, должен породить трёх потомков, соответствующих директориям `./a/c`, `./a/d` и `./a/e`.

После построения дерева процессов каждому процессу (обозначим PID очередного процесса через  $i$ ) приписывается значение  $a_{i,0}$ , равное количеству элементов в каталоге, соответствующем этому процессу (при подсчёте числа элементов в директории текущую и родительскую директорию учитывать не нужно; так, например, для только что созданной директории  $a_{i,0} = 0$ ). Далее это значение *синхронно* изменяется у всех процессов по следующему правилу:

$$a_{i,j} = \frac{\sum_{k \in R(i)} c_{i,k} a_{k,j-1}}{\sum_{k \in R(i)} c_{i,k}}, \quad j = 1, 2, \dots,$$

где в множество  $R(i)$  входит число  $i$ , а также входят PID всех ближайших процессов-родственников процесса с PID, равным  $i$  (т.е. родителя и сыновей), а числа  $c_{i,k}$  получены с помощью генератора псевдослучайных чисел и все *положительны*.

Произведите достаточно большое число итераций по  $j$ , выведите получившиеся значения  $a_{i,j}$  на экран и сделайте предположение, к чему сходятся числа  $a_{i,j}$  при  $j \rightarrow \infty$  (и всегда ли сходятся?).

Требования:

1. Формат вывода: каждый процесс в отдельной строке должен вывести следующую информацию: свой PID, PID родителя, список PID потомков, результирующее значение  $a_{i,j}$ . Пример (для структуры каталогов, которая описана выше):

```
1408 1407 [1409 1410] 0.5  
1409 1408 [1411 1412 1413] 0.6  
1410 1408 [] 0.7  
1411 1409 [] 0.5  
1412 1409 [] 0.6  
1413 1409 [] 0.5
```

Порядок строк значения не имеет. Процесс вывода для предотвращения смешивания выводимой информации также нужно синхронизировать.

2. После того как произведено необходимое количество итераций, каждый родитель должен дожидаться завершения работы всех своих потомков.
3. Для обмена информацией между процессами должны использоваться неименованные каналы, созданные с помощью вызовов `pipe`.
4. Считать, что число элементов в *каждой* из директорий ограничено 100.
5. Вычисление следующего значения  $a_{i,j}$  по приведённой выше формуле должно производиться *только* в узле с PID, равным  $i$ .

Дополнительное задание на «+»: сделать останов не по числу итераций, а когда *для всех*  $i$  будут одновременно выполняться неравенства  $|a_{i,j} - a_{i,j-1}| < \varepsilon$ , где  $\varepsilon > 0$  — заданное как аргумент командной строки действительное число.

**Мягкий deadline: 7.12.2019.**

**Жёсткий deadline: 23:59, 13.12.2019.**