

Классная работа №5
30.10.2019

1. Напишите программу, которая считает количество простых чисел в диапазоне от 1 до N наивным алгоритмом (написать функцию `int isprime(int n)`). Необходимо разделить всю работу между всеми имеющимися ядрами (используйте функцию `int get_nprocs()`) и вычислить ответ в параллельном режиме. Ответы необходимо собрать через `pipe` (дочки пишут свои ответы через `write(fd[1], &answer, sizeof(int))`, а родитель читает числа).

Запуск	<code>./prog 1000000000</code>
Вывод	50847534

2. Скомпилировать все программы *.c в указанной директории. Используйте

```
DIR *opendir(const char *name);
struct dirent *readdir(DIR *dirp);
int closedir(DIR *dirp);
```

3. Скомпилировать все программы *.c в указанной директории в параллельном режиме (по количеству доступных процессов). То есть если имеется 2 ядра и 3 программы A.c B.c C.c, то сначала необходимо дождаться завершения первых двух дочерних процессов, а только потом запускать третий процесс. В этом нам поможет семафор из `pipe`. Чтобы поднять семафор — пишем в него 1 байт, чтобы опустить семафор — читаем из него 1 байт. Изначально семафор равен количеству доступных ядер.

- Родитель порождает все дочерние процессы.
- Дочерний процесс проверяет семафор. Как только он свободен, то порождает внучку.
- . Дочерний процесс ждет завершения внучки. Как только она завершает увеличивает семафор.

4. По нажатию `ctrl + c` увеличивать счётчик до заданного числа, при достижении указанного числа завершить программу.

Запуск	<code>./prog 3</code>
Вывод	1 2 3

5. Напишите программу, которая считает количество простых чисел в диапазоне от 1 до N наивным алгоритмом (написать функцию `int isprime(int n)`). По нажатию `ctrl + z` выводить сколько сейчас проверено чисел. После вывода продолжить счёт с места остановки. Можно использовать глобальную переменную.

Запуск	<code>./prog 1000</code>
Вывод	10 23 100 25

6. Запустить другую программу. По нажатию `ctrl + c` завершить запущенную программу и вывести слово `end`.

Запуск	<code>./prog another</code>
Вывод	end