

# IO bound операции в коорутинах

Посмотрим на выигрыш в производительности от короутин с io-операциями:

```
import asyncio
import time

SUB_PROGS_COUNT = 100

async def worker():
    await asyncio.sleep(1)

async def main():
    begin = time.time()
    await asyncio.gather(
        *[worker() for _ in range(SUB_PROGS_COUNT)]
    )
    print(time.time() - begin)

asyncio.run(main())
```

1.008833885192871

**В 100 раз быстрее**, чем то же число последовательных вызовов.

Программа отработала чуть больше, чем за секунду. Не тратила время на ожидание, а передавала управление другим короутинам. Все это отработало в одном потоке и одном процессе. Мы не тратили вычислительные мощности на переключения и выделения ресурсов подпрограммам.