

# Многопоточное программирование

## Практика

Юрий Литвинов  
y.litvinov@spbu.ru

21.09.2022

## По домашке про матрицы

- ▶ Замерять производительность, запустив код один раз — неправильно
  - ▶ Масса случайных факторов, от планировщика до занятости ОС другими задачами
- ▶ Все замеры времени **всегда** требуют матстата!
  - ▶ Время работы — случайная величина с нормальным (как правило) распределением
  - ▶ Замеряя время, мы строим выборку этой величины
  - ▶ По выборке можно оценить матожидание и дисперсию
  - ▶ Именно матожидание и дисперсия есть результаты замера!

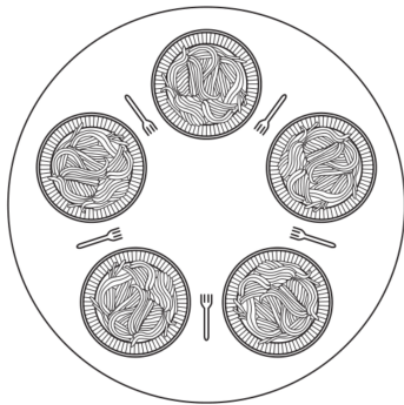
## Напоминание про условия взаимной блокировки

1. имеется разделяемый ресурс, к которому потоки хотят получить доступ, но пользоваться им может только один поток
2. таких ресурсов несколько и поток, захватив один, хочет получить доступ к другим, которые в этот момент захвачены другими потоками
3. нельзя отнять захваченный ресурс у потока
4. потоки ждут друг друга «по кругу»

Блокировка возможна, только если выполнены сразу все эти условия.

# Задача, “Обедающие философы”

- ▶ Есть  $N$  тарелок спагетти,  $N$  вилок и  $N$  философов
- ▶ Философ может думать и есть
- ▶ Чтобы есть, философу нужны две вилки
- ▶ Пример — транзакция, переводящая деньги со счёта на счёт



© A. Tanenbaum, Modern Operating Systems

## Что надо сделать

- ▶ Смоделировать ситуацию обедающих философов
  - ▶ Придумать красивую объектно-ориентированную модель
- ▶ Выводить на экран состояния философов
- ▶ Считаем, что философы думают и едят случайное, но небольшое количество времени
- ▶ Реализация должна гарантировать отсутствие взаимоблокировок
- ▶ Нужно уметь корректно останавливать процесс и распускать философов по домам