

Языки прикладного программирования

Лекция 9

```
for object to mirror_mod.mirror_object
```

```
operation == "MIRROR_X":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
@selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.name))  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly one object")
```

```
-- OPERATOR CLASSES --
```

```
types.Operator):  
    X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```

Процессы, потоки...

Из чего состоит компьютер?

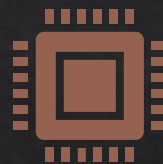
Из чего состоит компьютер?



Оперативная
память (ОЗУ/RAM)



Долговременная
память (ПЗУ)

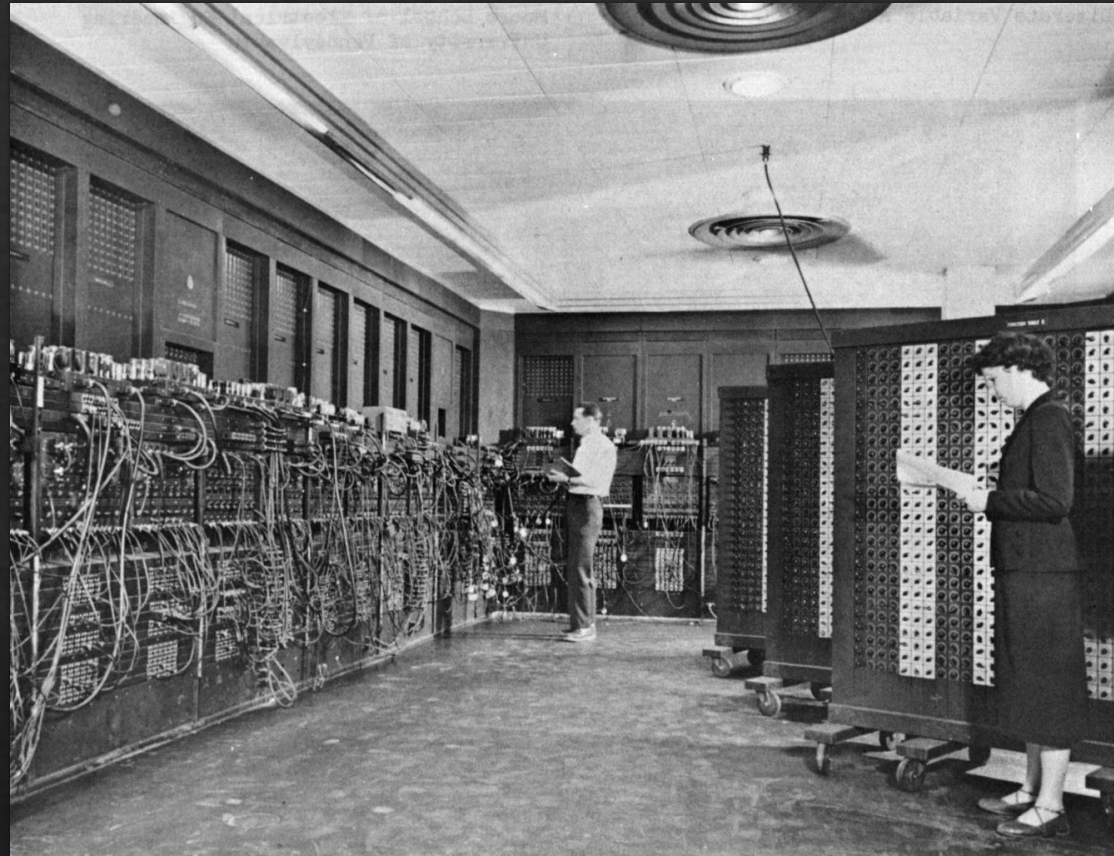


Процессор
(CPU)



Периферия
(ввод/вывод)

Первые ЭВМ



Пакетная обработка

С*10.95 ОПРЕДЕЛЕНИЕ АДРЕСА АКТИВНОЙ РЕАЛИЗАЦИИ ПАРАМЕТРА ЗАДАЧИ

1	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

IX 1985 г. ТУ 57-7-81

ТЕХНОИНФОРМ



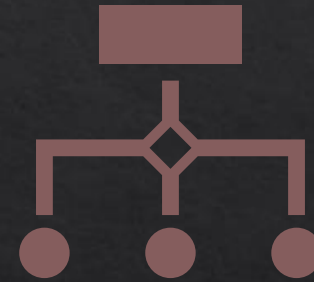
Системы разделения времени



Процессы и потоки

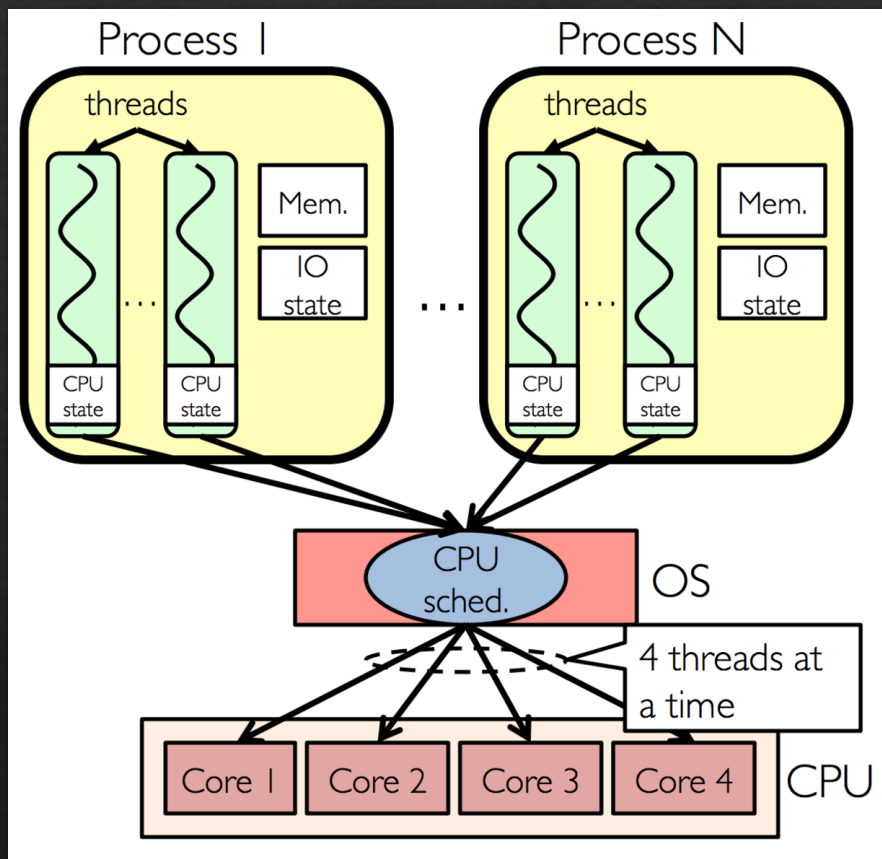


Процесс – программа в стадии исполнения.

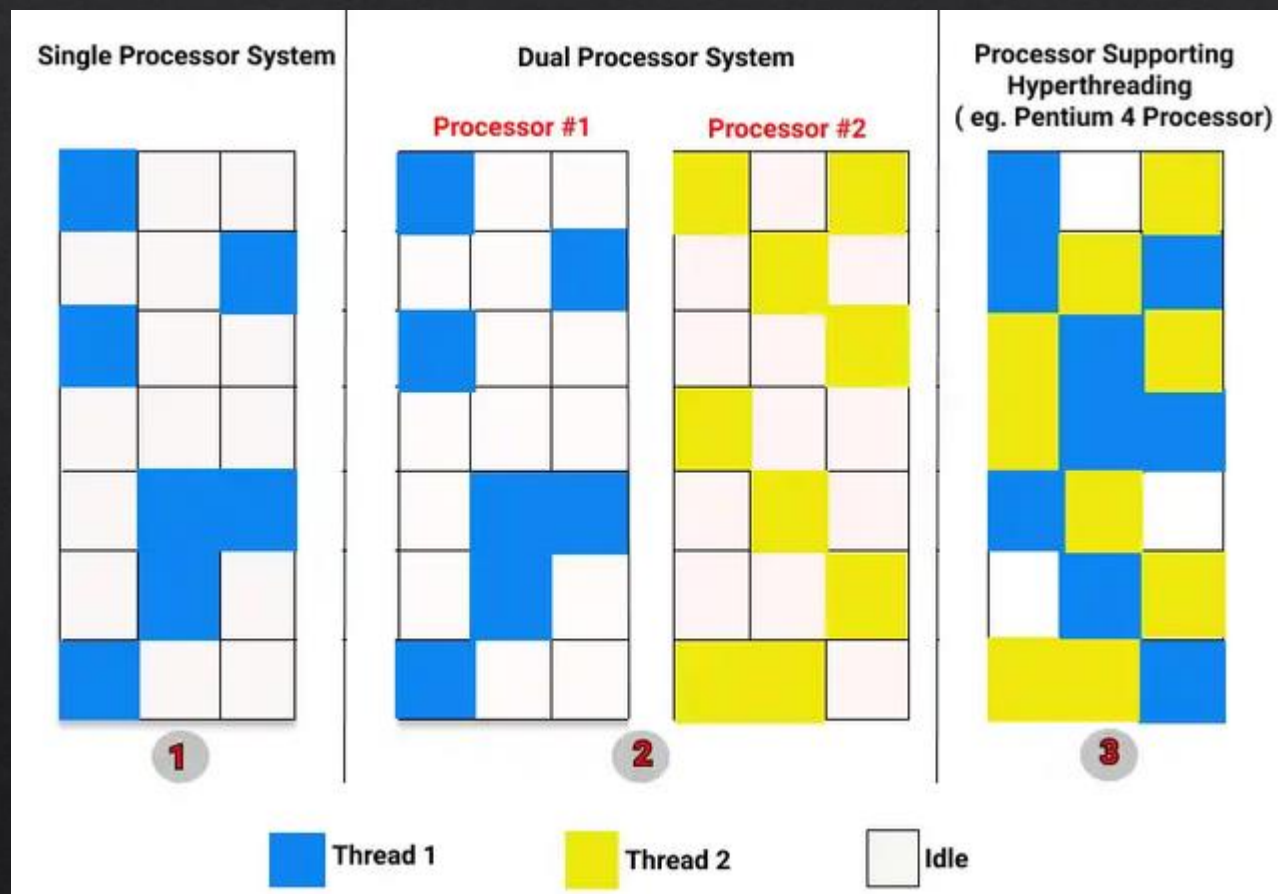


Потоки – ветви кода, выполняющиеся «параллельно», то есть без предписанного порядка во времени.

Реальная параллельность



Виртуализация ядер

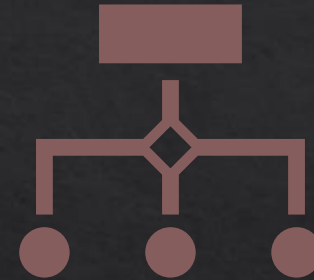


Отличие процесса и потока



Процесс - заявка для ОС на все виды ресурсов (память, файлы и пр.), кроме процессорного времени,

т.е. способ сгруппировать взаимосвязанные данные и ресурсы.



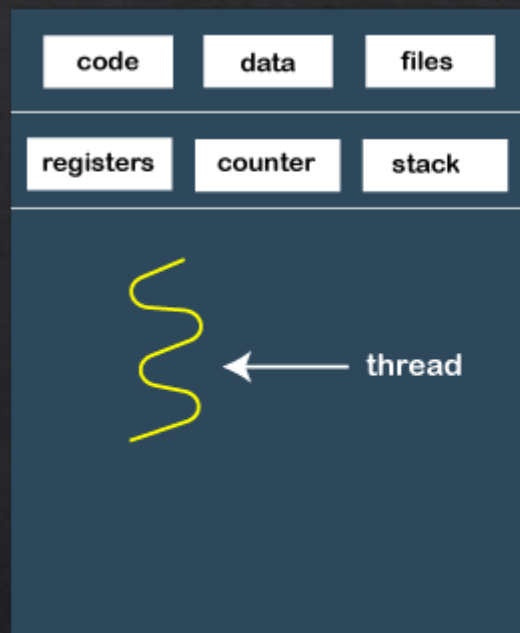
Поток — это заявка на процессорное время,

т.е. единица выполнения, исполняемая на процессоре.

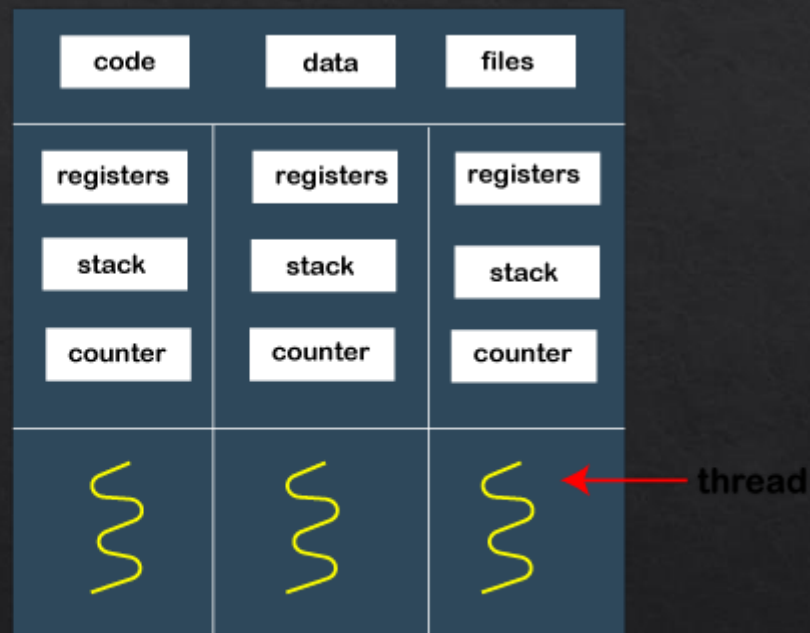
Процессы и потоки

- ◆ Поток – «легче», «дешевле» чем процесс (содержит только регистры, стек, счетчик команд).
- ◆ Потоки могут получить доступ к ресурсам процесса так как используют его адресное пространство
- ◆ Процессы изолированы друг от друга, обладают своим собственным адресным пространством и данными.

Потоки и процессы

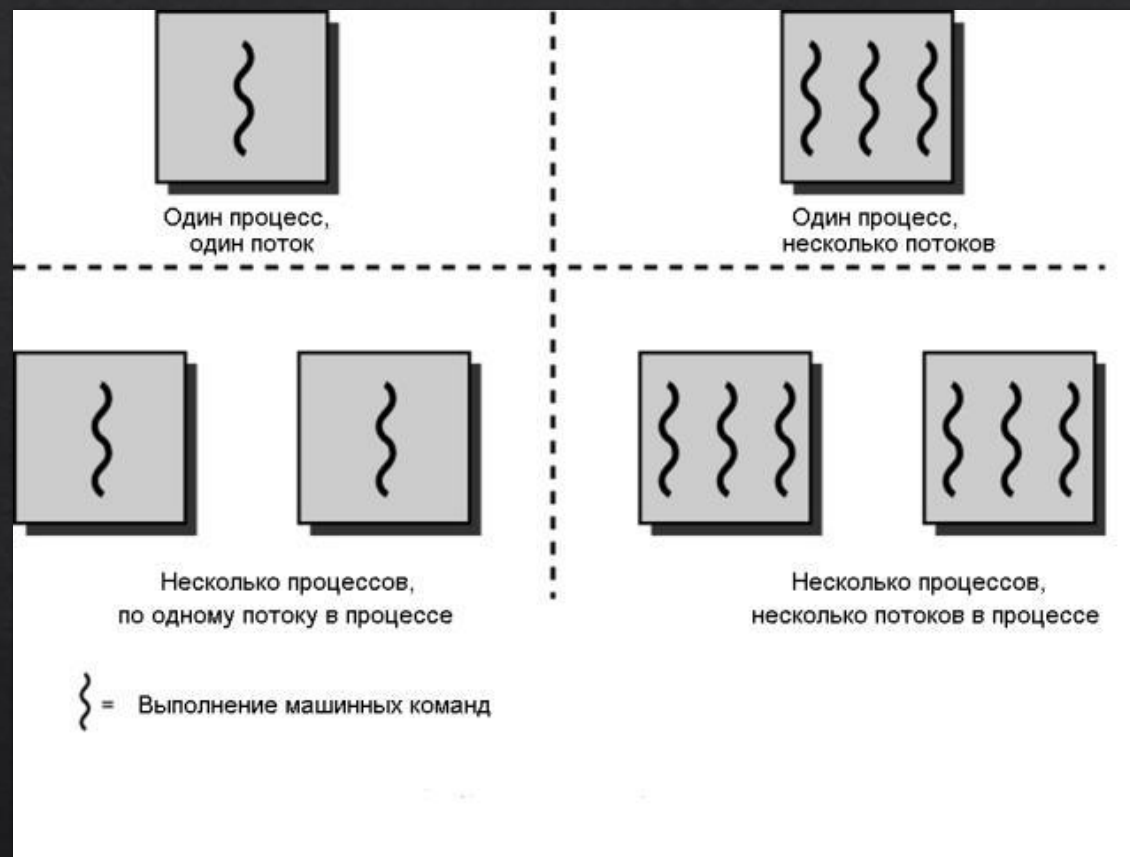


Single-threaded process



Multi-threaded process

Так много вариантов...



Многозадачность и многопоточность



Многозадачность — свойство среды выполнения (ОС) обеспечивать возможность параллельной (или псевдопараллельной) обработки нескольких задач.



Многопоточность — свойство платформы (ОС, виртуальной машины и т. д.) или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из нескольких потоков, выполняющихся «параллельно», то есть без предписанного порядка во времени.

А есть подвох?

Состояние гонки (Race condition)

Thread 1	Thread 2		Integer value
			0
read value		←	0
increase value			0
write back		→	1
	read value	←	1
	increase value		1
	write back	→	2

Состояние гонки (Race condition)

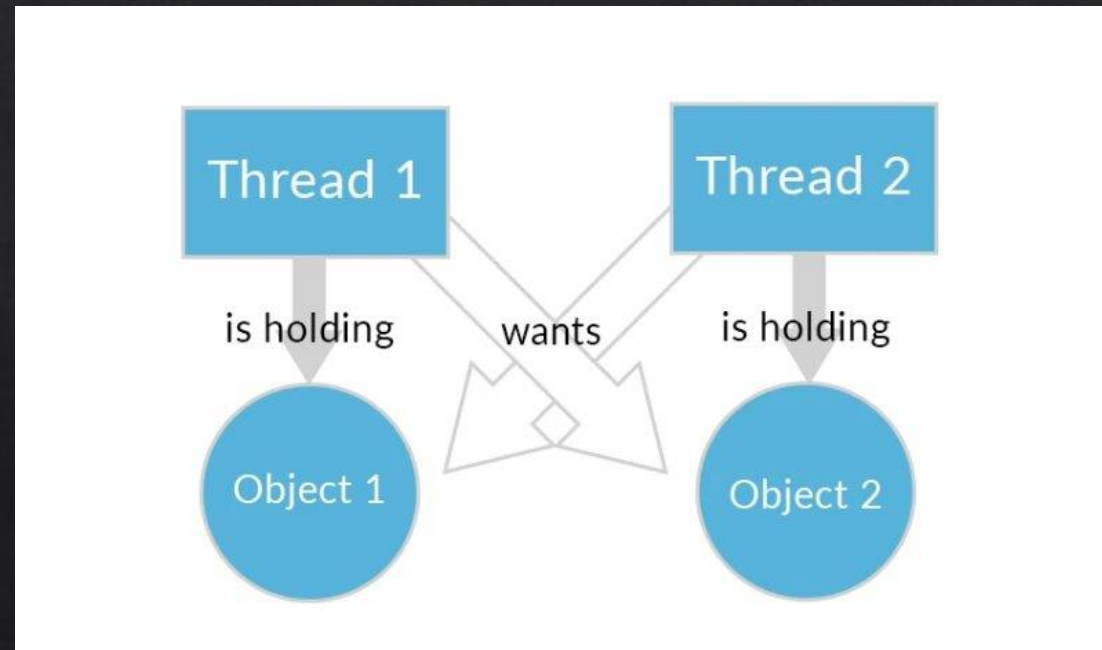
Thread 1	Thread 2		Integer value
			0
read value		←	0
	read value	←	0
increase value			0
	increase value		0
write back		→	1
	write back	→	1

Состояние гонки (Race condition)

- ◇ Решение:
 - ◇ Атомарные операции
 - ◇ Различные примитивы синхронизации
 - ◇ Мьютекс
 - ◇ Критическая секция (lock)
 - ◇ Семафор
 - ◇ Спинлок
 - ◇ Монитор
 - ◇ ...

Deadlock

- ◆ Deadlock — ситуация, при которой несколько потоков находятся в состоянии ожидания ресурсов, занятых друг другом, и ни один из них не может продолжать выполнение.



Livelock (голодание потоков)

- ◆ Голодание потоков — это ситуация, в которой поток не может получить доступ к общим ресурсам, потому что на эти ресурсы всегда претендуют какие-то другие потоки, которым отдаётся предпочтение.

Асинхронность

Что такое асинхронность?

task 3



task 2



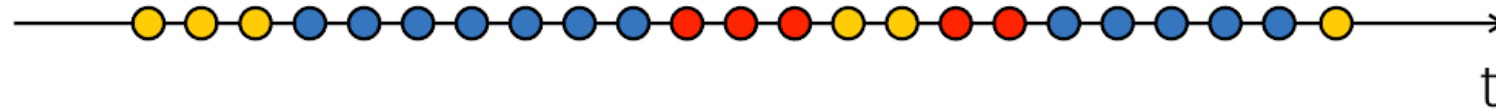
task 1



task 3

task 2

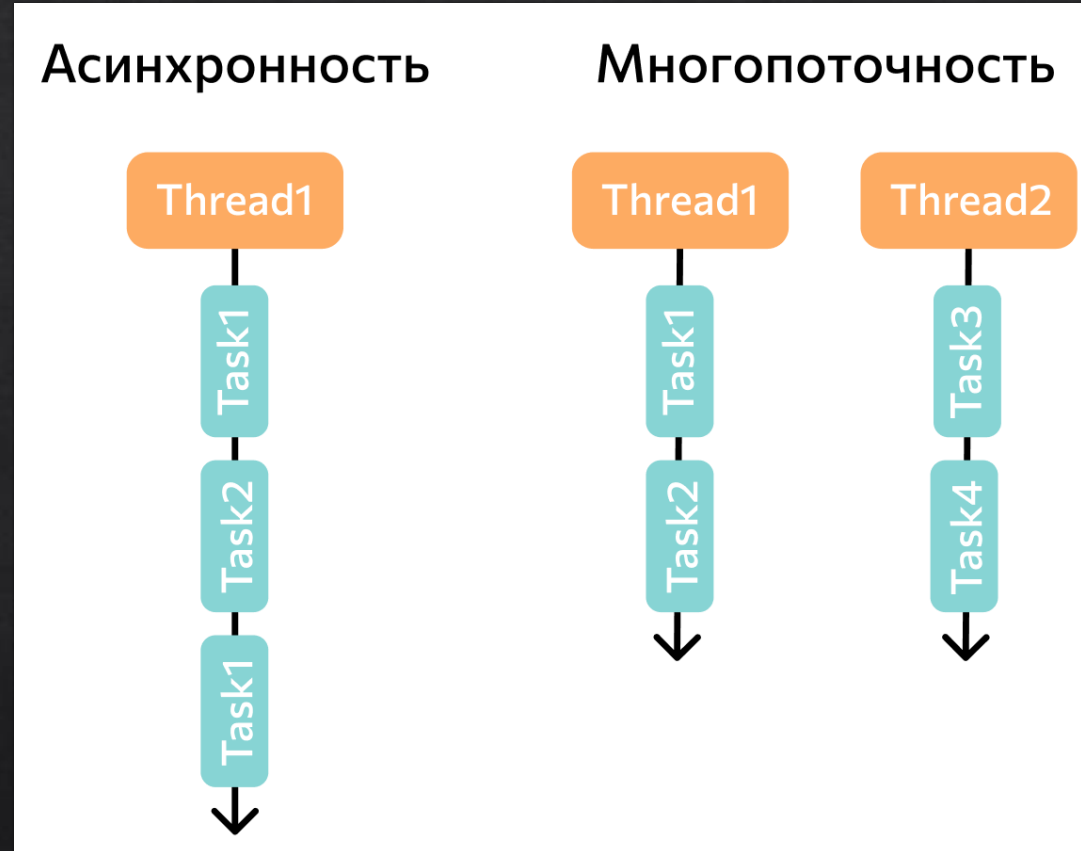
task 1



Что такое асинхронность?

- ◆ Асинхронное программирование — концепция программирования, при которой результат выполнения функции доступен спустя некоторое время в виде асинхронного (нарушающего стандартный порядок выполнения) вызова. Запуск длительных операций происходит без ожидания их завершения и не блокирует дальнейшее выполнение программы
- ◆ В общем случае асинхронность может работать на одном потоке.

Асинхронность и многопоточность



Что почитать?

- ◆ Таненбаум, Бос: Современные операционные системы
- ◆ <https://habr.com/ru/company/otus/blog/549814/>
- ◆ <https://habr.com/ru/post/470830/> - асинхронность в С#, забегание вперед.