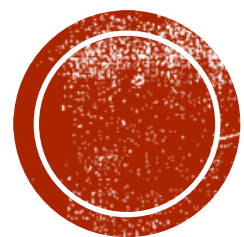


# ЯЗЫКИ ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ

Лекция 8





# ПРОЦЕССЫ, ПОТОКИ...







ИЗ ЧЕГО СОСТОИТ  
КОМПЬЮТЕР?





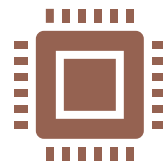
# ИЗ ЧЕГО СОСТОИТ КОМПЬЮТЕР?



Оперативная  
память (ОЗУ/RAM)



Долговременная  
память (ПЗУ)



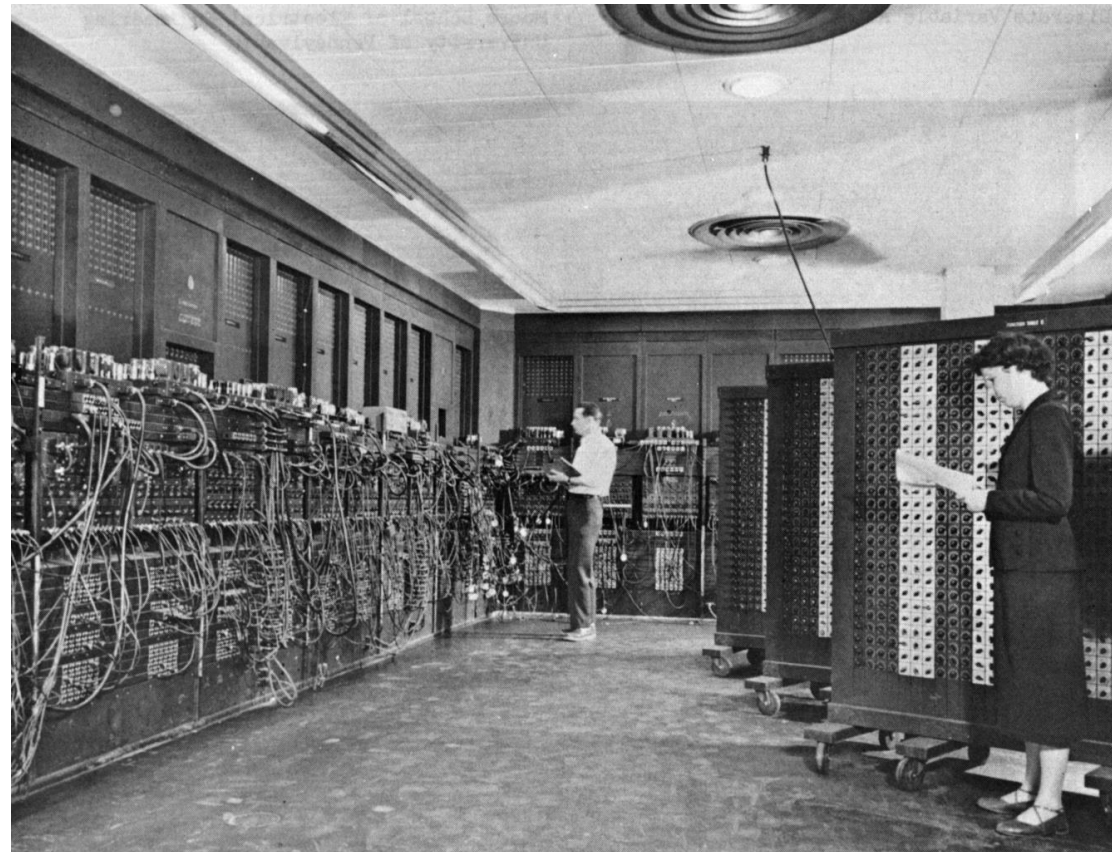
Процессор  
(CPU)



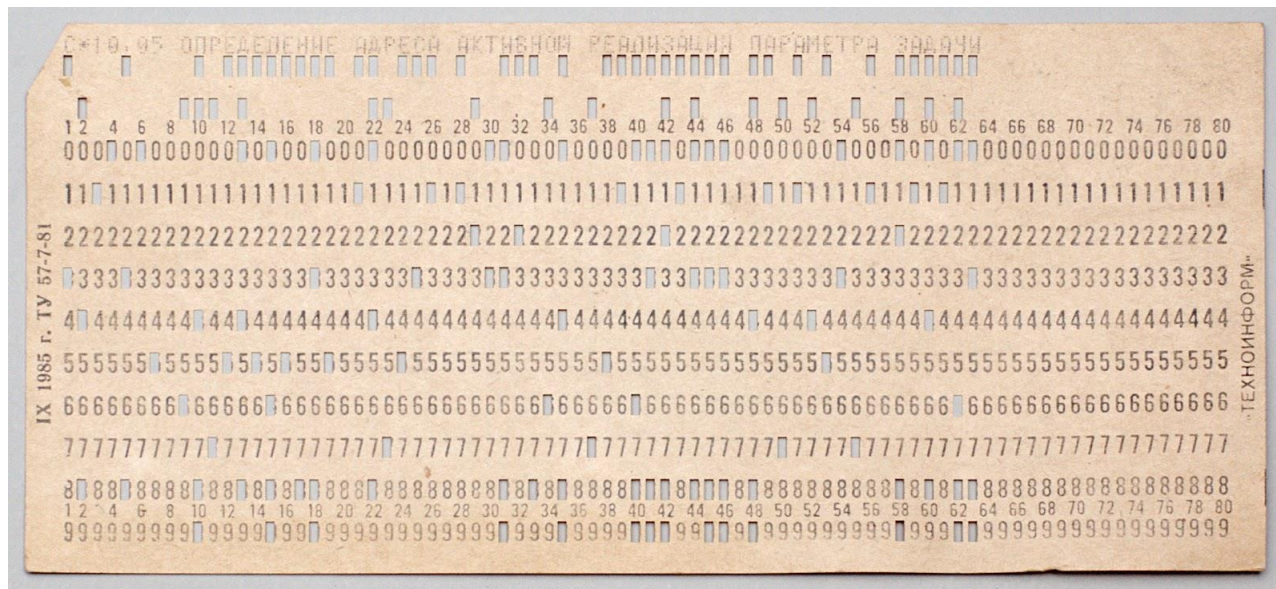
Периферия  
(ввод/вывод)



# ПЕРВЫЕ ЭВМ



# ПАКЕТНАЯ ОБРАБОТКА

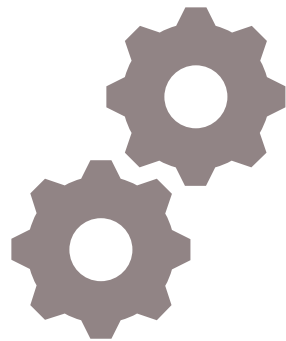


# СИСТЕМЫ РАЗДЕЛЕНИЯ ВРЕМЕНИ

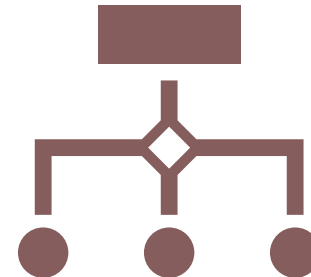


# ПРОЦЕССЫ И ПОТОКИ

---



Процесс – программа в стадии исполнения.

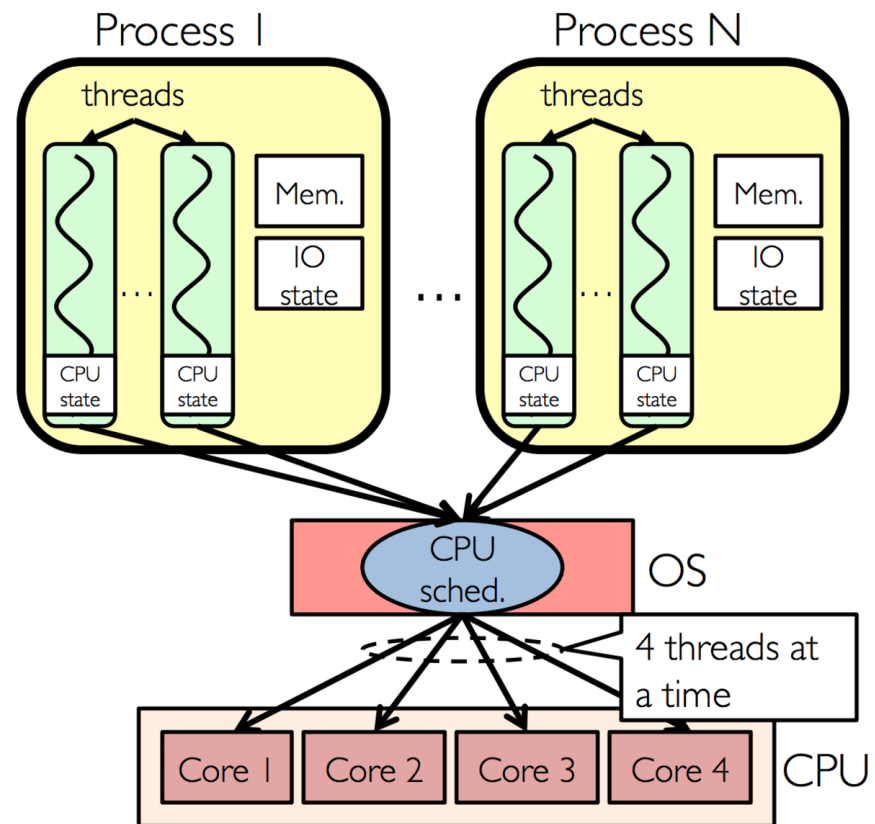


Потоки – ветви кода, выполняющиеся «параллельно», то есть без предписанного порядка во времени.

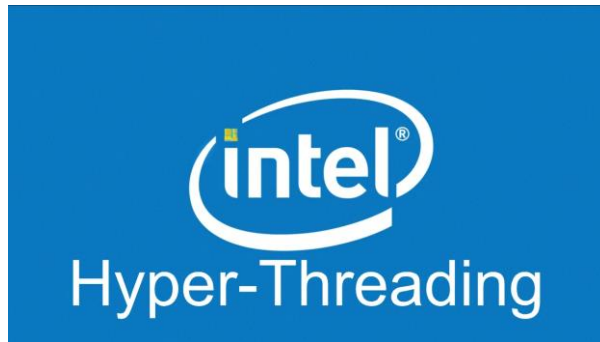




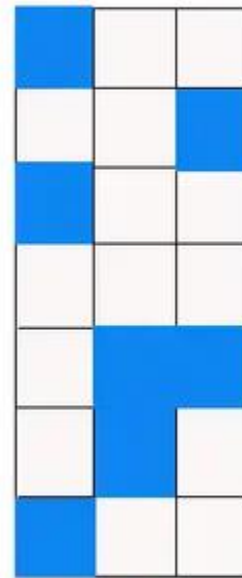
# РЕАЛЬНАЯ ПАРАЛЛЕЛЬНОСТЬ



# ВИРТУАЛИЗАЦИЯ ЯДЕР



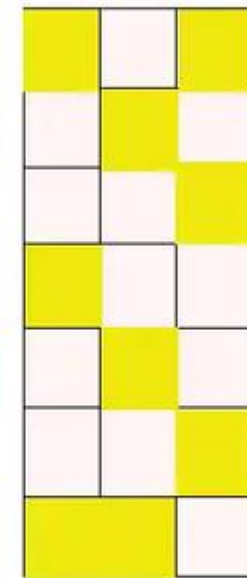
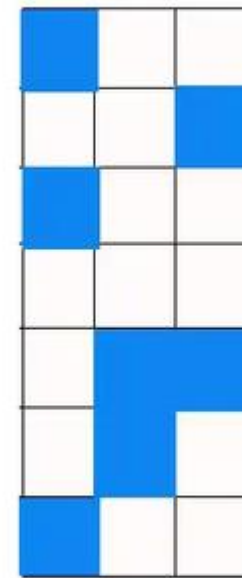
Single Processor System



Dual Processor System

Processor #1

Processor #2



Processor Supporting Hyperthreading  
(eg. Pentium 4 Processor)



1

2

3

Thread 1

Thread 2

Idle







## 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.

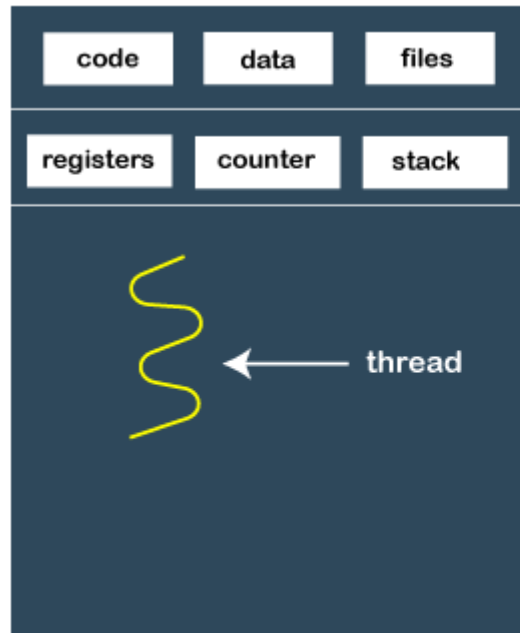
# ПРОЦЕССЫ И ПОТОКИ

- 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.

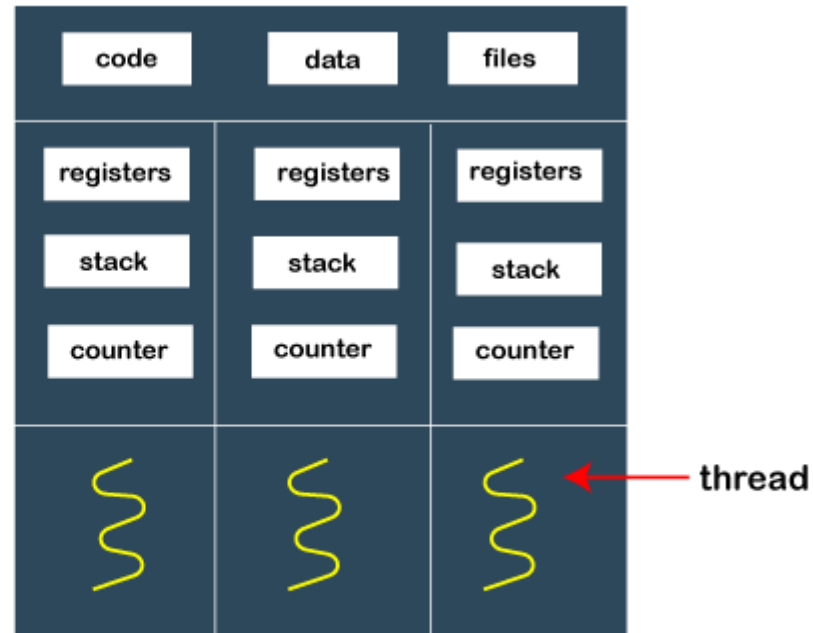




# ПОТОКИ И ПРОЦЕССЫ



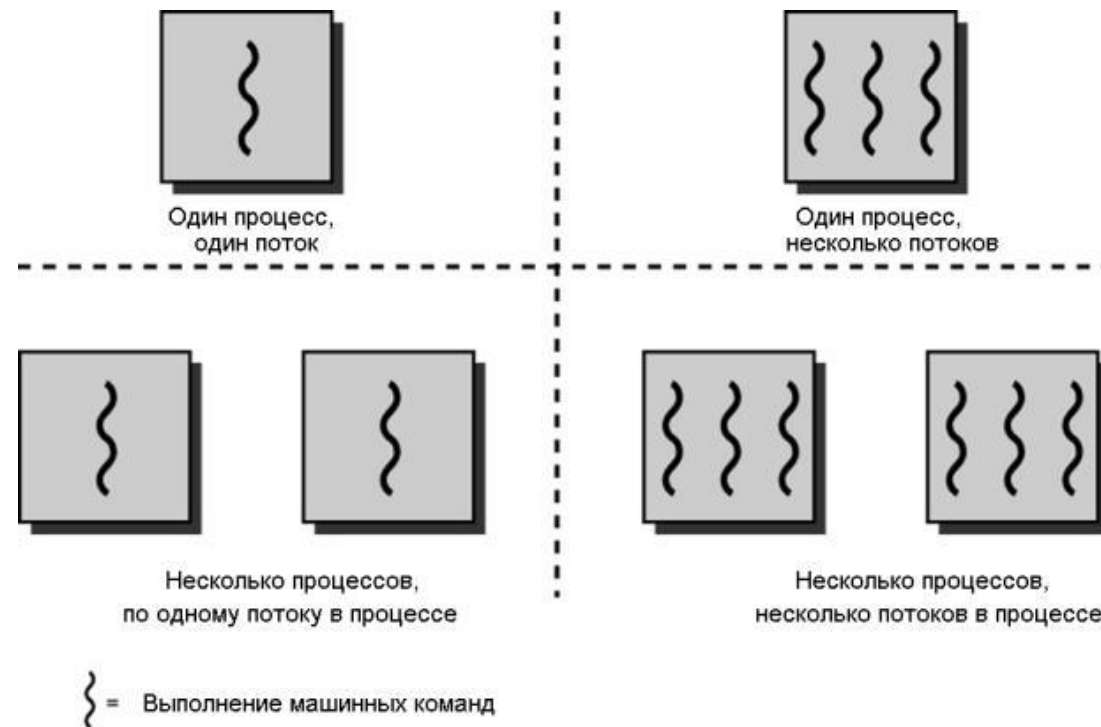
Single-threaded process



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



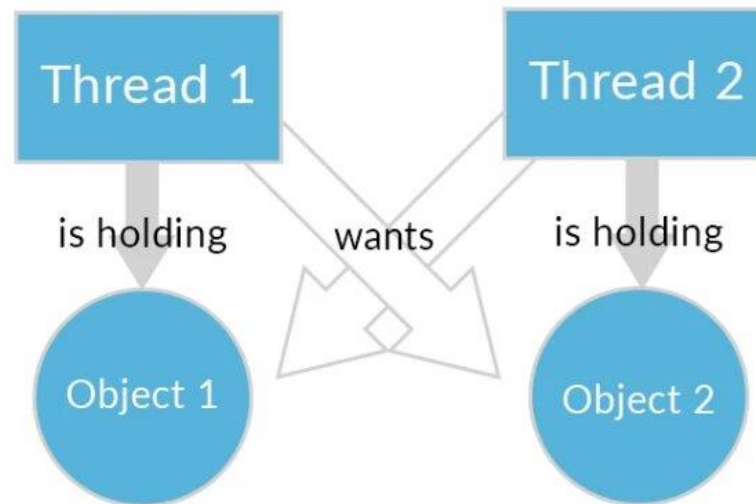
# СОСТОЯНИЕ ГОНКИ (RACE CONDITION)

- 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.

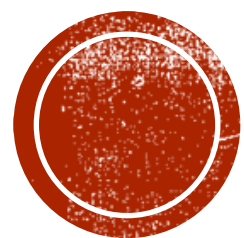




# LIVELOCK (ГОЛОДАНИЕ ПОТОКОВ)

- 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



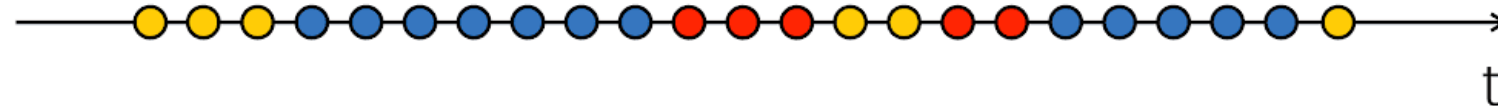
task 1



task 3

task 2

task 1





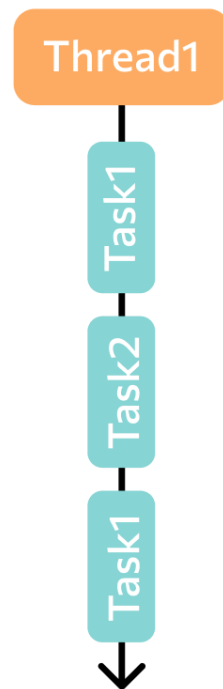
# ЧТО ТАКОЕ АСИНХРОННОСТЬ?

- 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.

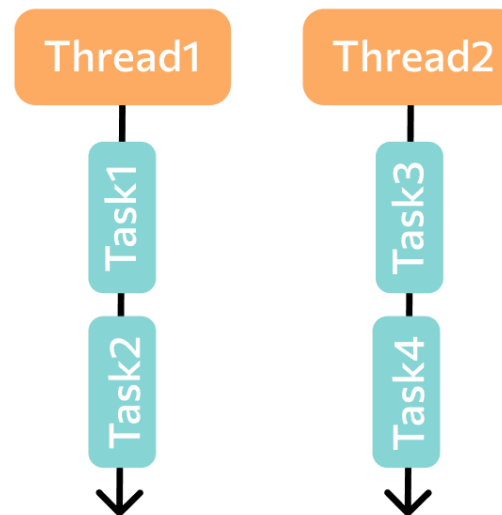


# АСИНХРОННОСТЬ И МНОГОПОТОЧНОСТЬ

Асинхронность



Многопоточность



# ЧТО ПОЧИТАТЬ?

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

