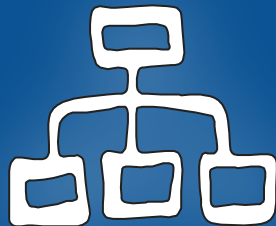


# Спец. семинар «Корректность программ и операционные системы»

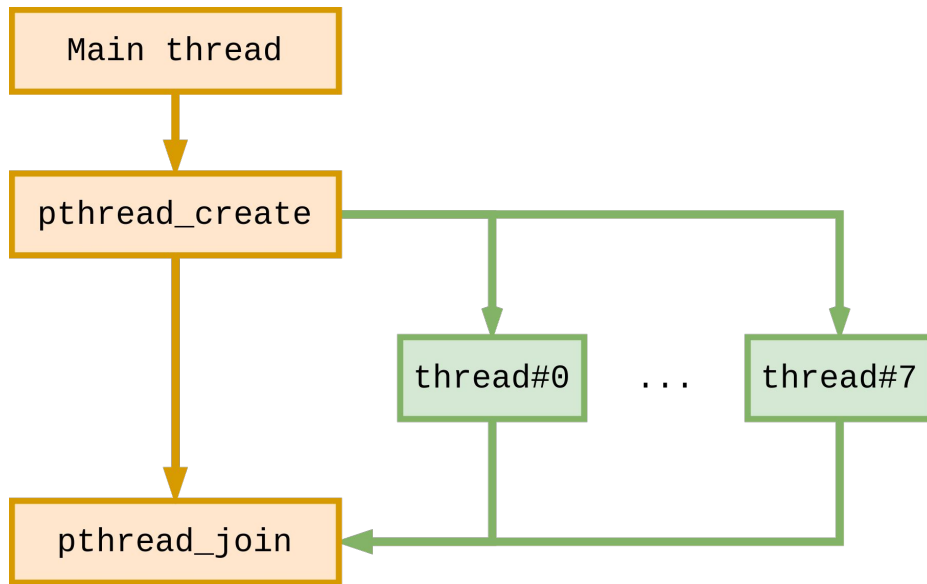
Семинар #1:

1. Организация многопоточности в POSIX и ОС Linux.
2. Зоопарк примитивов синхронизации:  
атомарная переменная, spinlock, мьютекс и семафор.
3. Сравнение производительности примитивов.

# Организация многопоточности в POSIX и ОС Linux



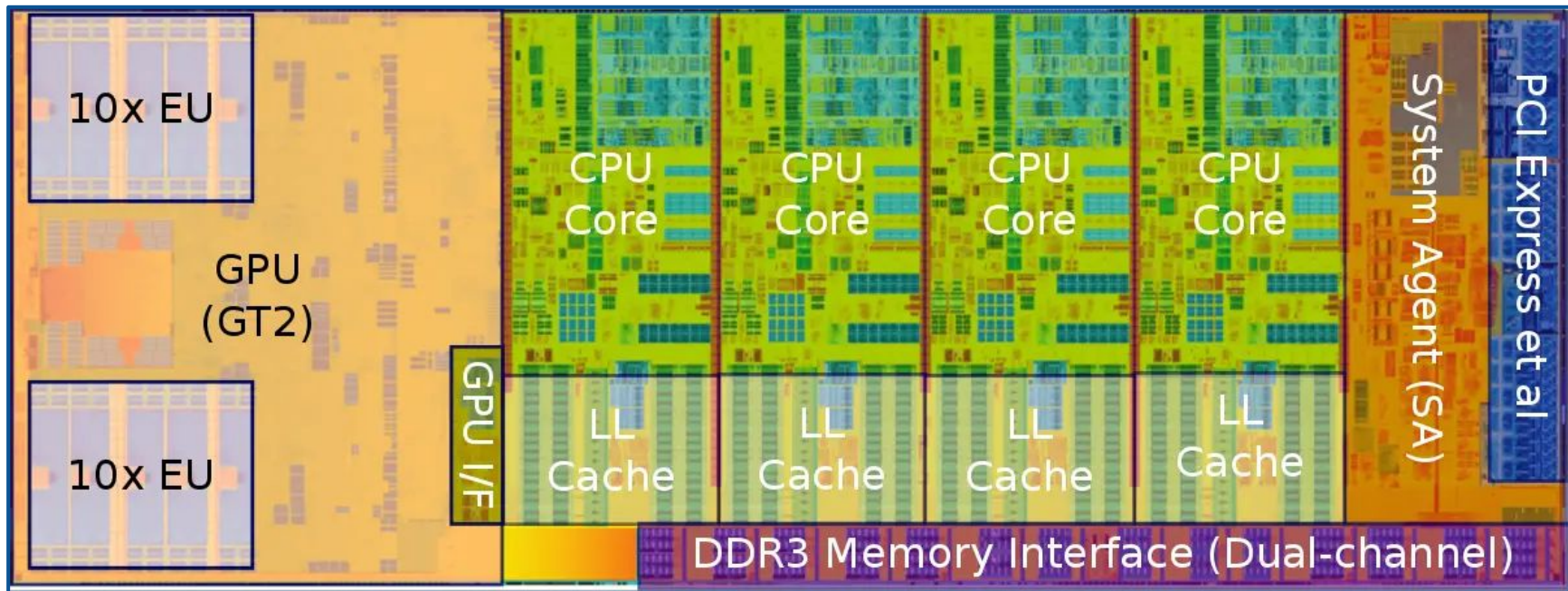
# Многопоточность в pthreads



**Реализация – библиотека NPTL в glibc (тяжеловесные потоки).**

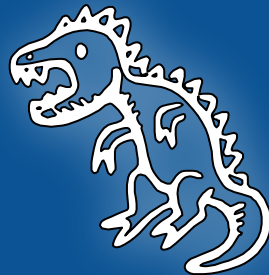
Атрибуты в `pthread_create`: аффинность потока, алгоритм планировщика потоков, адрес стека потока.

# Воздействие аффинности потоков



Распределение потоков по ядрам процессора влияет на эффективность их синхронизации.

# Зоопарк примитивов синхронизации



# Синхронизация потоков

```
mov     eax, dword [var] ; race.c:44 var++; ; 0x402c
add     eax, 1
mov     dword [var], eax ; 0x402c
```

## Подходы к синхронизации потоков:

- Без синхронизации ☠
- Атомарные переменные (lock-free)
- Spinlock-и
- Блокировка на уровне планировщика ОС
- Гибридные подходы

# Производительность синхронизации

Бенчмарк примитивов синхронизации: [01\\_pthreads\\_sync](#)

Примитив синхронизации	Время на $10^7 N$ сложений	Особенности реализации
Без синхронизации	0.2 с	Отсутствует корректность
Атомарные переменные	2.0 с	Частный случай синхронизации
Мьютекс pthreads	8 с	Применение <code>futex()</code>
Спинлоки	15 с	TAS lock
Семафоры pthreads	20 с	Чуть другие spinlock-и
Семафоры SYS V IPC	110 с (для $10^6 N$ )	Реализация в ядре, транзакционность

**Спасибо за внимание!**  
**Вопросы?**

