

Лекция №4 по дисциплине «ОПЕРАЦИОННЫЕ СИСТЕМЫ»

ПОТОКИ В ОС

Преподаватель: Золотоверх Д.О.

ЗАДАЧИ ОС

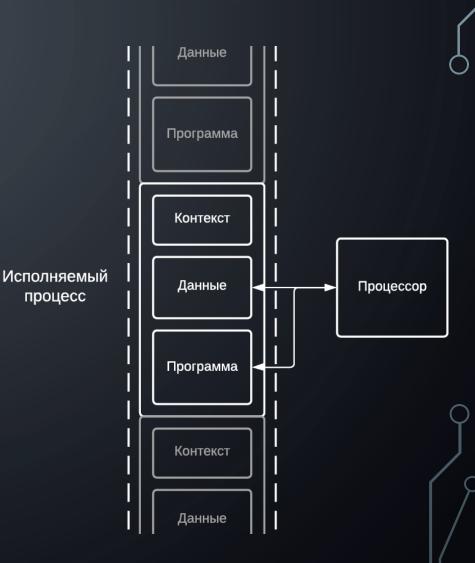
- Управлять запуском нескольких процессов
- Предоставлять необходимые ресурсы процессу
- Защищать ресурсы процесса от других процессов
- Позволять процессам обмениваться информацией
- Позволять процессам синхронизироваться

НЕ ЗАБЫВАЕМ, ЧТО

- Компьютер состоит из нескольких типов ресурсов
- ОС должна предоставлять доступ к ресурсам
- Все ресурсы кроме процессора являются медленными
- Приложения не должны писаться под конкретное «железо»

ПОНЯТИЕ ПРОЦЕССА

- Программа совокупность инструкций
- Процесс непосредственное выполнение этих инструкций
- Процессу могут принадлежать некоторые ресурсы
- Процесс имеет определенное состояние



ПРОЦЕСС С ТОЧКИ ЗРЕНИЯ ОС

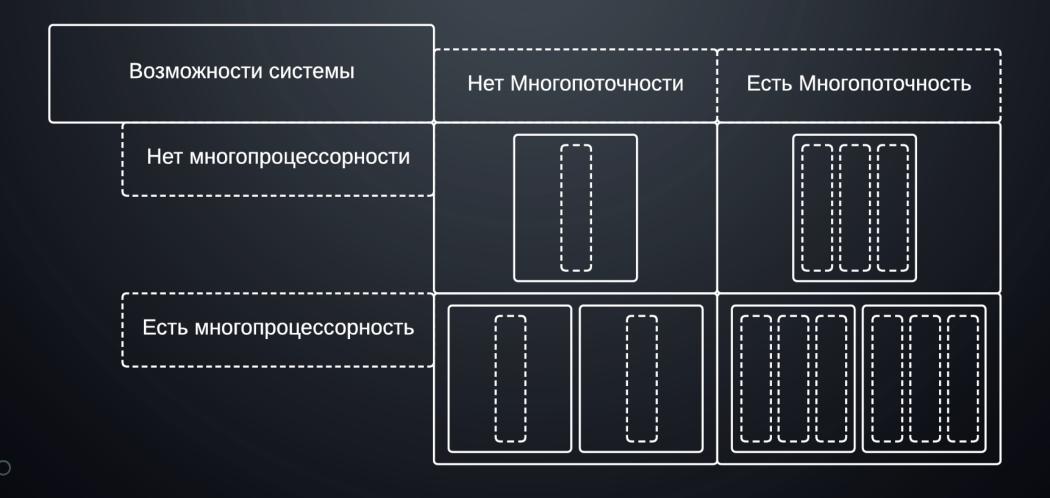
- Ресурсы, над которыми владеет процесс
 - Память
 - Устройства ввода-вывода
- Планирование и исполнение процесса
 - Создание
 - Выполнение
 - Переключение
 - Уничтожение
 - Приоритеты

ПОТОК (ТРЕД, НИТЬ)

- Наименьшая единица обработки для ОС
- Поток существует как часть процесса
- Многопоточность способность ОС, платформы или приложения запускать много потоков в рамках одного процесса.



ВОЗМОЖНОСТИ ОС, ПЛАТФОРМЫ, ПРОГРАММЫ



КОМПОНЕНТЫ ПРОЦЕССА

- Идентификатор
- Контекст
- Состояние
- Приоритет
- Счетчик команд
- Указатель на память

- Информация о статусе устройств
- Другая информация

КОМПОНЕНТЫ ПОТОКА

- Состояние
- Контекст
- Стеки выполнения
- Ресурсы потока
- Ресурсы процесса

ОТЛИЧИЯ ОТ ПРОЦЕССА

- Поток зависим от процесса и является его частью
- Потоки и процессы используют одно адресное пространство
- Потоки более простая для ОС структура данных
- Переключение между потоками происходит быстрее



ОТЛИЧИЯ ОТ ПРОЦЕССА

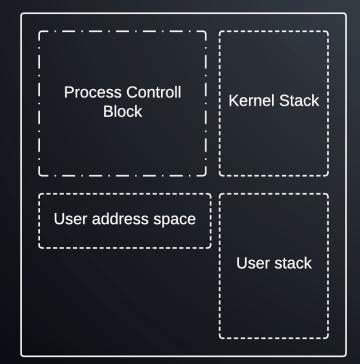
- Решение ОС или пользователя по поводу процесса влияет на поток:
 - Приостановка процесса => приостановка всех потоков
 - Завершение процесса => завершение всех потоков

 Как и процессы, потоки имеют состояния и могут синхронизироваться с другими потоками

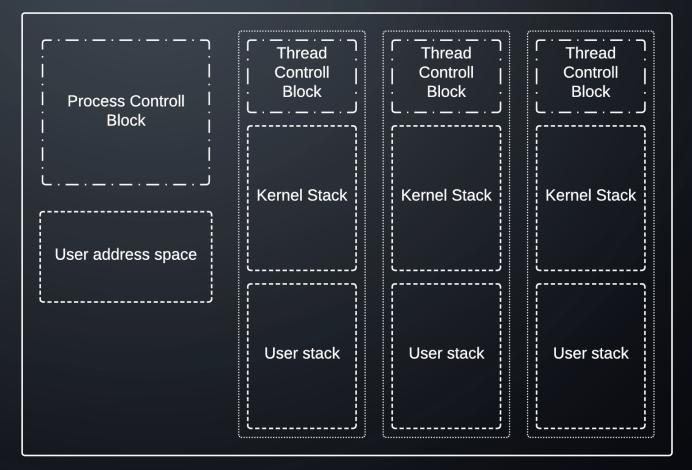


СТРУКТУРА ПРОЦЕССА С ПОТОКАМИ

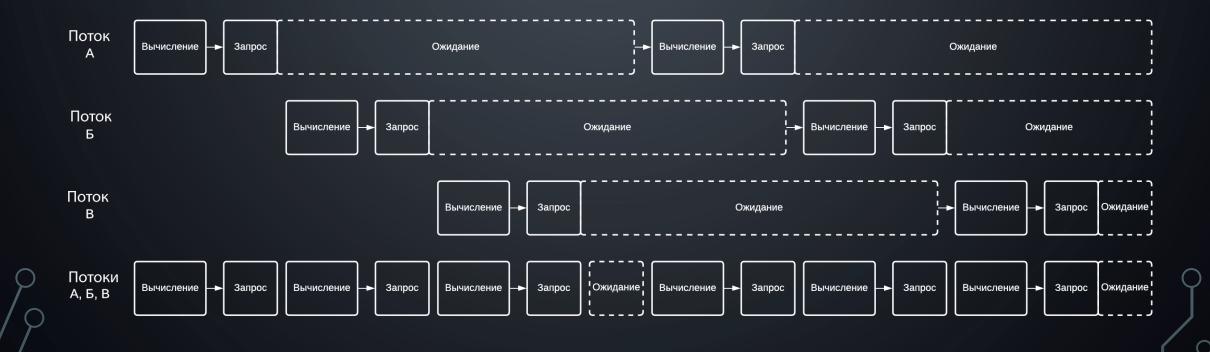
Процесс без многопоточности



Процесс с многопоточностю и несколькими потоками



выполнение потоков



РЕАЛИЗАЦИИ ПОТОКОВ

- На уровне Ядра (Kernel threads)
 - Потоки реализованы на уровне ОС
 - ОС хранит данные о потоках
- На уровне пользователя (User threads)
 - ОС не знает о существовании потока
 - Потоки реализованы на уровне приложения

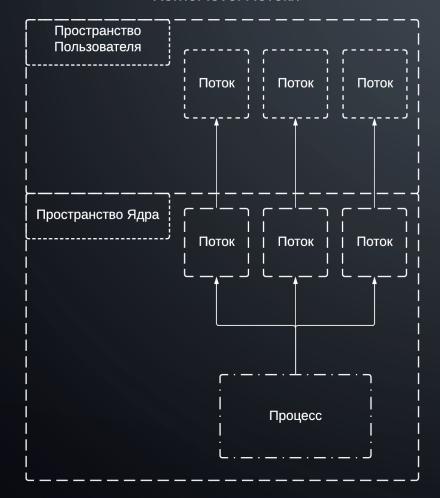


В ЧЕМ РАЗНИЦА?

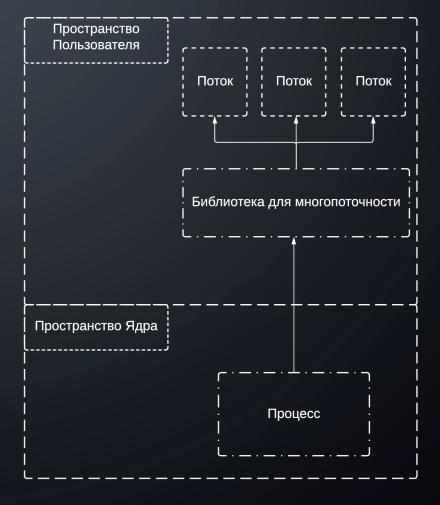
- + ОС знает о потоках и может эффективно ими управлять
- + Хорошо подходит для приложений с частыми блокировками
- + ОС может запланировать выполнение потоков на разных ядрах процессора
- Передача контроля от одного потока к другому требует обращение к ядру ОС
- Реализация на уровне ОС является более медленной
- Дополнительные затраты на управление

РЕАЛИЗАЦИИ ПОТОКОВ

Kernel-level Потоки



User-level Потоки



РЕАЛИЗАЦИЯ ПОТОКОВ

- На уровне Ядра (Kernel threads)
 - Потоки реализованы на уровне ОС
 - ОС хранит данные о потоках
- На уровне пользователя (User threads)
 - ОС не знает о существовании потока
 - Потоки реализованы на уровне приложения



СПАСИБО ЗА ВНИМАНИЕ!

