The "Santa's Helper" Task Manager

Задание:

Необходимо создать консольное (CLI) приложение, имитирующее диспетчер задач для Санты, который будет эффективно распределять три типа работы:

1. **Проверка Списков Подарков** (I/O-bound, **Асинхронность**).
2. **Упаковка Подарков** (CPU-bound, **Многопроцессность**).
3. **Кормление Оленей** (**Многопоточность**).

1: Проверка Списков Подарков

Цель: чтение и обработка множества файлов с пожеланиями

1. Создать асинхронную функцию check_wishlist(child_name, delay), которая:
    - Принимает имя ребенка и имитирует задержку чтения (например, от 0.5 до 2 секунд) с помощью await asyncio.sleep(delay).
    - Выводит в консоль сообщение о начале и завершении проверки списка для этого ребенка.
2. Написать основную асинхронную функцию manage_wishlists(), которая:
    - Создает список задач (tasks) для 10-15 "детей" с разными случайными задержками.
    - Запускает эти задачи конкурентно с помощью asyncio.gather() и дожидается их выполнения.

2: Упаковка Подарко

Цель: Имитировать сложную, вычислительно интенсивную работу по упаковке .

1. Создать функцию pack_gift(gift_name, complexity), которая:

- Принимает название подарка и его "сложность" (число для имитации работы).
- Имитирует длительную вычислительную работу, например, выполняя цикл с большим количеством итераций (например, 10^7 или 10^8 операций), чтобы максимально загрузить одно ядро.
- Выводит в консоль сообщение о начале и завершении упаковки.

2. Написать функцию start_packing_station(gift_list, num_workers), которая:
    - Использует модуль multiprocessing.Pool для создания пула процессов.
    - Распределяет список из 5-7 "сложных" подарков между процессами.
    - Дожидается завершения всех процессов и сообщает об этом.

3: Кормление Оленей

Цель: Имитировать несколько одновременных действий, которые могут быть **блокирующими**

1. Создать функцию feed_reindeer(reindeer_name, food_amount, time_to_eat), которая:
    - Принимает имя оленя, количество еды и время, необходимое для "кормления".
    - Использует time.sleep(time_to_eat) для имитации блокирующего ожидания.
    - Выводит в консоль сообщения о начале и завершении кормления.
2. Написать функцию start_feeding_station(reindeer_list), которая:
    - Создает несколько экземпляров threading.Thread (около 4-5 оленей).
    - Запускает потоки и дожидается их завершения с помощью .join().

4: Главный Диспетчер

Цель: Объединить все части и измерить общее время.

1. Создать главную функцию main(), которая:
    - Фиксирует время начала работы.
    - Последовательно вызывает manage_wishlists(), start_packing_station(), и start_feeding_station().
    - Фиксирует время окончания.
    - Выводит общее время, затраченное на все новогодние задачи.
2. Попробовать запустить все три станции **одновременно** (используя потоки для вызова асинхронной части и процессов/потоков для остальных).