**Задание 1: Парсер новостных сайтов с использованием горутин и каналов**

**Описание:** Разработайте многопоточный парсер для новостных сайтов, который будет собирать заголовки последних статей с нескольких ресурсов. Программа должна асинхронно обращаться к API новостных сайтов (или использовать фейковые URL для тренировки) и сохранять полученные данные в общий отчет, при этом выводить результаты, как только данные от сайта получены. Это симулирует задачу реального парсинга новостей, которую можно масштабировать для анализа контента или мониторинга новостей.

**Требования:**

* Используйте горутины для параллельного обращения к нескольким сайтам.
* Примените буферизированные каналы для передачи данных между горутинами.
* Добавьте тайм-аут на случай медленных ответов, чтобы избежать долгого ожидания от одного источника.
* Создайте итоговый отчет с заголовками всех сайтов и временной меткой получения данных.

**Задание 2: Анализ логов сервиса с фильтрацией событий**

**Описание:** Создайте приложение для параллельного анализа логов от разных сервисов компании. Логи включают данные о событиях, таких как ошибки, запросы пользователей и системные сообщения. Программа должна асинхронно получать данные от нескольких сервисов и фильтровать их, сохраняя только сообщения об ошибках для дальнейшего анализа.

**Требования:**

* Разработайте функцию, которая принимает список логов и фильтрует их, оставляя только ошибки.
* Используйте горутины для параллельной обработки логов с нескольких сервисов.
* Примените паттерн fan-in для объединения отфильтрованных логов в единый канал для удобного вывода.
* Сохраните итоговый список ошибок в файл или переменную для дальнейшего анализа.

**Задание 3: Пул воркеров для обработки пользовательских транзакций**

**Описание:** Реализуйте пул воркеров для обработки списка транзакций пользователей, таких как денежные переводы. Программа должна параллельно обрабатывать транзакции, учитывая при этом ограничения на число одновременных операций. Это полезно для систем, в которых важно контролировать параллельную нагрузку, предотвращая перегрузку.

**Требования:**

* Создайте фиксированный пул воркеров для обработки списка транзакций.
* Реализуйте мьютексы для контроля доступа к балансу пользователей.
* По завершении всех транзакций выведите итоговый баланс каждого пользователя.
* Используйте группу ожидания (WaitGroup) для синхронизации выполнения всех операций.