

Таким образом, разработанная охранная система позволяет оборудовать транспортные средства дешевой альтернативой существующим сигнализациям и обеспечить удобное управление с помощью мобильного приложения, предоставляющего пользователю удобный инструмент для круглосуточного мониторинга состояния транспортного средства и оперативного реагирования на внештатные ситуации независимо от местоположения владельца.

#### Литература

1. Голубцов, М. С. Микроконтроллеры AVR: от простого к сложному / М. С. Голубцов. – Санкт-Петербург : БХВ-Петербург, 2020. – 352 с.
2. Кострин, Д. К. Датчики в электронных устройствах / Д. К. Кострин, А. А. Ухов. – Санкт-Петербург : ЛЭТИ, 2018. – 240 с.
3. Белов, А. В. Микроконтроллеры в устройствах на базе датчиков / А. В. Белов. – Санкт-Петербург : Наука и техника, 2021. – 368 с.

### РАСПРЕДЕЛЕННОЕ ПРИЛОЖЕНИЕ ИЗВЛЕЧЕНИЯ РЕЧИ ИЗ БОЛЬШИХ АУДИОФАЙЛОВ

Т. А. Зинчук

Учреждение образования «Гомельский государственный технический  
университет имени П. О. Сухого», Республика Беларусь

Научный руководитель Е. Д. Гуменников

*Рассмотрено создание распределенного приложения для извлечения речи из больших аудиофайлов, ориентированное на обработку длительных и ресурсоемких аудиоданных в условиях ограниченных вычислительных ресурсов отдельных узлов. Описана архитектура системы с разделением ролей менеджера и воркеров, а также механизм взаимодействия компонентов через TCP и веб-интерфейс на базе ASP.NET. Уделено внимание используемым примитивам синхронизации. Показаны основные этапы пайплайна: загрузка файла, сегментация, извлечение речевых фрагментов с использованием SileroVAD, транскрибация моделью Whisper и агрегация результатов. Отдельное внимание уделено отказоустойчивости, масштабируемости и особенностям развертывания в Windows в формате self-contained исполняемых файлов.*

**Ключевые слова:** распределенная обработка, аудио, SileroVAD, Whisper, отказоустойчивость, Windows.

Распределенное приложение извлечения речи из больших аудиофайлов представляет собой программно-аппаратный комплекс, ориентированный на обработку длительных и ресурсоемких аудиоданных в условиях ограниченных вычислительных ресурсов отдельных узлов. Основной задачей разработки является обеспечение устойчивой, масштабируемой и отказоустойчивой архитектуры, способной эффективно функционировать в операционной системе Windows без необходимости сложной установки и конфигурации.

Архитектура системы строится по принципу разделения ролей на менеджер (координирующий узел) и набор воркеров (исполняющих узлов). Менеджер отвечает за прием задач, их декомпозицию, распределение между воркерами и агрегацию результатов. Воркеры настраивают задачи и с помощью stdin/stdout-интерфейса выполняют распределение задач по процессам-обработчикам, выполняющим непосредственную обработку аудио с использованием операций линейной алгебры и моделей машинного обучения, выполняющих извлечение и транскрибацию речи. Для извлечения используется довольно распространенная модель *SileroVAD*, а для транскри-

бации используется модель *Whisper*. Процессы-обработчики написаны в виде Python-скриптов, а сетевые узлы системы на .NET 8.0.

Эффективность дистрибуции файлов достигается за счет использования различных примитивов синхронизации: от использования lock-переменных до concurrent-версий переменных для хранения менеджером информации о списке обрабатываемых файлах и состояния воркеров менеджером и состояния Python-обработчиков воркерами. Также активно используется класс Task и CancellationToken для одновременного выполнения множества рутинных задач.

Система ориентирована на работу с файловой моделью, где центральным объектом является аудиофайл, подлежащий транскрибации. Менеджер хранит метаданные задач, временные файлы сегментов и результирующие текстовые данные. Воркеры работают с локальными копиями сегментов, что позволяет минимизировать сетевые задержки и повысить производительность. В названиях временных файлов содержатся временные метки, которые используются для определения расположения сегмента в изначальном файле. Используются префиксы названий файлов в виде SessionId, что позволяет системе работать с несколькими пользователями одновременно.

Для взаимодействия пользователя и менеджера используется связка ASP.NET Web-Application, который обеспечивает интерфейс в браузере и HTTP API. При передаче файлов используется multipart form-data, что позволяет без ограничений передавать в систему файлы больших размеров. Кроме того, реализация клиентского приложения на базе ASP.NET позволяет использовать сервис через обычный веб-браузер – без необходимости установки дополнительного ПО. Сервер веб-страницы, при необходимости, конвертирует файл для соответствия требуемым параметрам аудио для обработки (WAV, 8/16 кГц, моно, 16-бит).

Воркеры с менеджером взаимодействуют с использованием протокола TCP, который обеспечивает надежность и прозрачность. При подключении к менеджеру воркеры отправляют сообщение о доступных обработчиках, которые менеджер учитывает при распределении задач. Также менеджер ведет учет HeartBeat-сообщений от воркеров, и в случае обнаружения задержки, превышающей допустимый предел менеджер перераспределяет задачи на работающие воркеры. Воркер в свою очередь, при обнаружении разрыва соединения с менеджером, сбрасывает все обработчики, после чего заново пытается установить соединение.

Пайплайн обработки аудиофайла включает несколько этапов. На первом этапе происходит загрузка и регистрация файла в системе и выдача SessionId пользователю. Далее следует этап сегментации, на котором аудиофайл разбивается на более мелкие фрагменты, удобные для параллельной обработки. На следующем этапе происходит извлечение речи, который позволяет уменьшить объем обрабатываемых и передаваемых по сети данных, выделяя только участки в аудиофайлах в которых присутствует речь. Затем выполняется транскрибация, включающая распознавание речи. На завершающем этапе происходит агрегация результатов и формирование итогового документа, который отсылается пользователю.

Так же при запуске Менеджер и Воркер используют файлы Configuration.json, с помощью которых можно конфигурировать сетевые настройки узлов и параметры начального разбиения аудиофайлов, в том числе и количество запускаемых Python-процессов, что позволяет настроить систему под различное аппаратное обеспечение.

Дистрибуция системы реализована в виде self-contained исполняемых файлов для Windows, что позволяет запускать компоненты без предварительной установки зависимостей, и требует наличия обычной флешки для развертывания системы. Ка-

ждому узлу соответствует отдельная директория, содержащая все необходимые библиотеки, конфигурационные файлы и исполняемый модуль.

Для разрешения зависимостей Python-обработчиков требуется скачать необходимые папки с зависимостями, включая клонирование репозитория Whisper и SileroVAD с GitHub. При скачивании зависимостей так же необходимо скачать torch 2.1 и torchaudio 2.1, так как это позволяет разрешить проблемы зависимостей бэкенда обработки аудио у SileroVAD. При запуске процессов обработчиков воркер обязан установить переменные среды на те, что находятся в папке зависимостей с помощью свойства `ProcessStartInfo.EnvironmentVariables`.

Таким образом, предложенная система демонстрирует эффективный подход к организации распределенной обработки аудиоданных в среде Windows. Она сочетает в себе простоту развертывания, гибкость взаимодействия компонентов и высокую отказоустойчивость, что делает ее перспективным решением для задач транскрибации больших аудиофайлов.

#### Литература

1. Silero Team. Silero VAD: pre-trained enterprise-grade Voice Activity Detector. – URL: <https://github.com/snakers4/silero-vad>.
2. Radford A. et al. Robust Speech Recognition via Large-Scale Weak Supervision. – URL: <https://github.com/openai/whisper>.
3. Microsoft. System.Collections.Concurrent Namespace. – URL: <https://learn.microsoft.com/en-us/dotnet/api/system.collections.concurrent>.

## ИНТЕРАКТИВНЫЙ ПРОГРАММНЫЙ ТРЕНАЖЕР ПО ТЕМЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

А. А. Готченя

Учреждение образования «Гомельский государственный технический  
университет имени П. О. Сухого», Республика Беларусь

Научный руководитель О. Д. Асенчик

*Разработан интерактивный программный тренажер по информационной безопасности на основе больших языковых моделей (LLM). Система выступает в роли виртуального экзаменатора: генерирует вопросы из загруженной базы нормативных документов, анализирует развернутые ответы обучаемых на естественном языке, при неверном ответе формирует наводящий вопрос – не раскрывая правильного решения – и по завершении сессии выставляет итоговую оценку с развернутым анализом. Применение технологии дополненной генерации (RAG) исключает галлюцинации и привязывает обучение к актуальной внутренней документации организации.*

**Ключевые слова:** большие языковые модели, информационная безопасность, дополненная генерация (RAG), интерактивный тренажер.

Информационная безопасность (ИБ) является одной из наиболее динамично развивающихся областей современных информационных технологий. Рост числа кибератак, утечек данных и инцидентов, связанных с действиями сотрудников, формирует устойчивый запрос на системы обучения, способные не просто информировать пользователей, но и формировать у них устойчивые практические навыки.

Традиционные методы подготовки – статичные тесты с выбором правильного варианта – демонстрируют принципиальное ограничение: пользователи запоминают