

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**ИНТЕГРАЦИЯ ИНСТРУМЕНТОВ ДЛЯ СОЗДАНИЯ СИСТЕМЫ
ВИДЕОТРАНСЛЯЦИЙ СОРЕВНОВАНИЙ ПО
ПРОГРАММИРОВАНИЮ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Мещерякова Максима Александровича

Научный руководитель

к. ф.-м. н.

С. В. Миронов

Заведующий кафедрой

к. ф.-м. н.

С. В. Миронов

Саратов 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Обзор используемых инструментов	5
1.1 WebRTC Grabber	5
1.2 ICPC Live V3	5
1.3 OBS Studio	5
2 Возможности и настройки инструментов	7
2.1 WebRTC Grabber	7
2.1.1 Подсистема Grabber	7
2.1.2 Подсистема Signaling	7
2.2 ICPC Live V3	8
2.2.1 Конфигурационный файл events.properties	8
2.2.2 Конфигурационный файл advanced.json	8
3 Интеграция инструментов и тестирование системы	9
ЗАКЛЮЧЕНИЕ	13

ВВЕДЕНИЕ

ICPC (International Collegiate Programming Contest) — это ежегодный международный конкурс по программированию, который проводится для студентов колледжей и университетов. Он был организован в Техасском сельскохозяйственном и машиностроительном университете в 1977 году и с тех пор стал одним из самых престижных соревнований по программированию в мире. Кроме того ICPC является важным событием для студентов и программистов со всего мира. Он позволяет им продемонстрировать свои навыки и знания в области программирования, а также получить ценный опыт участия в международном соревновании. В последние годы в отборочных турах ICPC участвуют уже более 60 000 студентов-программистов из более 3000 университетов из 115 стран. По масштабу соревнования могут поспорить с крупнейшими спортивными турнирами.

ICPC состоит из нескольких этапов: отборочного, полуфинального и финального. На каждом из этапов команды соревнуются между собой в решении задач на программирование, которые были разработаны организаторами чемпионата, а их сложность зависит от этапа соревнования.

Важным элементом соревнований по программированию для дальнейшего их развития и привлечения действующих или будущих студентов является широкомасштабное освещение. Одним из главных способов являются онлайн видеотрансляции самих соревнований.

Трансляция финального этапа ICPC — это уникальное событие, которое привлекает внимание большого количества людей во всем мире и позволяет им увидеть, как талантливые программисты соревнуются за звание лучших. Трансляции финального этапа ICPC обычно проводятся на интернет-платформах, таких как YouTube и Twitch. Во время трансляции зрители могут послушать комментарии опытных участников сообщества олимпиадного программирования, понаблюдать за ходом соревнований, а также за тем, как сами участники работают над решением задач.

Саратовский государственный университет провел уже 25 четвертьфинальных этапов ICPC (Чемпионат Юга и Поволжья России). На текущий момент во время соревнования освещается только турнирная таблица, и система видеотрансляций до сих пор не была интегрирована в процесс соревнования. Поэтому целью данной дипломной работы является интеграция инструментов

для создания системы видеотрансляций для проведения последующих соревнований по программированию, проводимых Саратовским государственным университетом. В частности, в рамках данной работы были поставлены следующие задачи:

- анализ существующих инструментов для реализации системы видеотрансляций соревнований по программированию, на примере финального этапа ICPC;
- ознакомление с возможностями инструментов;
- интеграция инструментов для системы видеотрансляций.

1 Обзор используемых инструментов

1.1 WebRTC Grabber

WebRTC Grabber — это приложение, которое позволяет пользователям захватывать видео рабочего стола, видео и аудио с веб-камеры или микрофона. Оно работает на основе технологии WebRTC, которая обеспечивает прямое соединение между браузерами и позволяет легко создавать видео- и аудиоконференции. WebRTC Grabber имеет открытый исходный код и поддерживается командой разработчиков ICPC Live, которые ответственны за трансляции финального этапа ICPC и чемпионата Северной Евразии (полуфинального этап ICPC). Кроме того, WebRTC Grabber поддерживает различные платформы, включая Windows, Mac OS X и Linux. Это делает его доступным для пользователей на разных операционных системах.

WebRTC Grabber состоит из двух подсистем:

1. Grabber — это автономное приложение для фоновой захвата и передачи изображения экрана и веб-камеры. Grabber работает в фоновом режиме и прослушивает входящие вызовы от Signaling сервера;
2. Signaling — это веб-сервер, основными функциями которого являются: регистрация участников конференции, обмен информацией о присутствии, управление соединением, обработка ошибок.

1.2 ICPC Live V3

ICPC Live V3 — это приложение для создания оверлея для трансляций соревнований по спортивному программированию. ICPC Live V3 имеет открытый исходный код и поддерживается командой разработчиков ICPC Live, которая ответственна за трансляцию финального этапа ICPC и чемпионата Северной Евразии (полуфинального этап ICPC).

1.3 OBS Studio

OBS (Open Broadcaster Software) — это бесплатное программное обеспечение с открытым исходным кодом для записи, создания видео и трансляции на различные сервисы, поддерживающие протокол Real Time Messaging Protocol (RTMP), такие как Twitch, YouTube, Facebook Live и другие. OBS позволяет записывать видео с экрана, записывать аудио, добавлять эффекты и фильтры, а также транслировать видео.

OBS Studio имеет простой и понятный интерфейс, который позволяет быстро настроить программу под свои нужды. Одной из главных особенностей OBS является его гибкость и настраиваемость. Пользователи могут создавать уникальные конфигурации, адаптированные к их потребностям и предпочтениям. Кроме того, OBS поддерживает множество плагинов и расширений, которые позволяют добавлять новые функции и возможности. В целом, OBS является мощным и гибким инструментом для создания качественного видеоконтента, который можно использовать как для личного использования, так и для профессиональной работы.

2 Возможности и настройки инструментов

2.1 WebRTC Grabber

WebRTC Grabber является свободным программным обеспечением и поэтому доступно для загрузки на сайте <https://github.com/irdkwmsb/webrtc-grabber/releases>.

2.1.1 Подсистема Grabber

На компьютеры участников необходимо извлечь файлы из zip-архива `webrtc_grabber_grabber_<платформа>_<архитектура>.zip`, который можно найти на странице загрузки релизов. Далее будут рассмотрены настройки Windows версии приложения, однако существуют Linux и MacOS версии.

Для запуска приложения используется пакетный файл `runner.bat`.

Код файла `runner.bat` приведен в листинге 1.

```
1 %~dp0grabber.exe . --peerName={id} --signalingUrl="{url}"
```

Листинг 1: Пакетный файл `runner.bat`

Здесь `id` является некоторым индексом текущего компьютера, а `url` — локальный адрес Signaling сервера.

Так как программа была создана для соревнований по программированию, чтобы не отвлекать участников подсистема Grabber запускается в скрытом режиме и не обладает каким-либо интерфейсом. Поэтому для тестирования подсистемы есть специальный пакетный файл `tester.bat`.

Для отключения приложения используется пакетный файл `stopper.bat`

2.1.2 Подсистема Signaling

Signaling представляет собой полноценный HTTP сервер с двумя точками сокета `socket.io`. Одна для подключения одноранговых узлов, а другая для подключения зрителей. Точка доступа для зрителей используется для просмотра всех доступных одноранговых узлов и инициирования соединения с ними. Также точка доступа для зрителей может быть защищена паролем, который указан в файле конфигурации.

На компьютер предназначенный для Signaling сервера необходимо извлечь файлы из zip-архива `webrtc_grabber_signaling_<платформа>_<архитектура>.zip`, который можно найти на странице загрузки релизов. Да-

лее будут рассмотрены настройки Windows версии приложения, однако существуют Linux и MacOS версии.

Для запуска сервера используется файл `signaling.cmd`.

2.2 ICPC Live V3

ICPC Live V3 является свободным программным обеспечением и поэтому доступно для загрузки на сайте <https://github.com/icpc/live-v3/releases/download/v3.0.8/live-v3-3.0.8.jar>.

Для настройки самого соревнования используются два конфигурационных файла: `events.properties` и `advanced.json`.

2.2.1 Конфигурационный файл `events.properties`

Конфигурационный файл `events.properties` содержит информацию о самом соревновании. ICPC Live V3 поддерживает несколько самых популярных систем. Для проведения соревнований в СГУ используются системы PCMS и Codeforces.

2.2.2 Конфигурационный файл `advanced.json`

Конфигурационный файл `advanced.json` предназначен для регулировки полученных из системы соревнования данных.

Параметр `teamOverrides` позволяет настроить полученной информации о каждом отдельном участнике. Также возможно создать шаблоны для медиа, которые будут применены ко всем участникам.

Параметр `scoreboardOverrides` позволяет настроить правила ранжирования участников в таблице.

Параметр `problemOverrides` позволяет настроить данные задач соревнования.

Также доступны параметры настройки относящиеся ко времени соревнования.

3 Интеграция инструментов и тестирование системы

Для тестирования интеграции инструментов будем использовать три компьютера, имитирующие компьютеры участников соревнований, один компьютер для Signaling сервера WebRTC Grabber и один компьютер для видеотрансляции. Все компьютеры находятся в одной локальной сети.

Запустим WebRTC Grabber на компьютере с peerName равным 001. Для удобства тестирования воспользуемся отладочным режимом граббера и запустим его через файл `tester.bat`. Таким образом мы сможем проверить корректность захвата экрана и веб-камеры.

Теперь посмотрим на изменения в интерфейсе веб-приложения WebRTC Grabber Signaling сервера, после включения одного из грабберов.

Signaling сервер смог произвести пинг граббера с peerName равным 001.

Теперь при выборе компьютера с peerName равным 001 зритель может просматривать изображение экрана и веб-камеру участника. Видео может быть открыто в отдельной вкладке, при нажатии на ссылку `webcam` или `desktop`, или встроено в текущую веб-страницу при нажатии на кнопку с аналогичными названиями.

Таким образом было проверено, что Singaling сервер успешно подключается ко включенным грабберам на удаленных компьютерах и получается видео рабочего стола и веб-камеры.

Следующим этапом является настройка и тестирование приложения ICPC Live V3 и его интеграция с WebRTC Grabber Signaling сервером.

Для тестирования использовался уже завершившееся соревнование проведенное на платформе Codeforces.

После запуска приложения ICPC Live V3, становится доступен веб-интерфейс для настройки оверлея трансляции.

В приложении доступны следующие вкладки:

- `controls`;
- `titles`;
- `teamview`;
- `teampvp`;
- `splitscreen`;
- `scoreboard`;
- `ticker`;

- dashboard;
- backend log;
- analytics;
- spotlight;
- overlay preview.

Рассмотрим некоторые из них более подробно.

Вкладка `controls` используется для отображения основной информации соревнования: таблицы соревнования, очереди посылок, статистики по задачам, бегущей строки, таймер соревнования.

Вкладка `titles` предназначена для добавления в оверлей трансляции некоторой дополнительной информации.

Вкладка `teamview` предназначена для отображения некоторой информации участников.

Вкладка `teampvp` устроена аналогично вкладке `teamview`, но позволяет отображать информацию сразу двух команд, где выводимая информация настраивается отдельно для каждой команды.

Вкладка `splitscreen` устроена аналогично вкладке `teamview`, но позволяет отображать информацию сразу четырех команд, где выводимая информация настраивается отдельно для каждой команды.

Вкладка `scoreboard` предназначена для настройки отображения таблицы соревнования.

Вкладка `ticker` предназначена для настройки отображения бегущей строки.

Вкладка `dashboard` объединяет в себе вкладки `controls`, `titles`, `scoreboard` и `ticker` для более удобной работы с ними.

Вкладка `backend log` содержит в себе лог серверной части приложения.

Вкладка `spotlight` содержит таблицу участников и значение их «популярности», которое зависит от текущих результатов, но может быть модифицировано вручную. Используя эти данные модуль `teamview` в автоматическом режиме будет выбирать наиболее «популярных» участников для показа на трансляции.

Были рассмотрены и протестированы все возможности приложения ICPC Live V3.

Далее предстоит настроить и протестировать OBS Studio и его интегра-

цию с ICPC Live V3.

После запуска приложения OBS Studio импортируем пресет от команды ICPC Live, который можно скачать по ссылке gg.gg/icpclive_obs. А затем активируем добавленный пресет как показано на рисунке.

После активации пресета будут созданы различные сцены, такие как Starting Soon, Speaker Local и другие. Также будет добавлена сцена, которая связана с веб-страницей ICPC Live V3.

Таким образом все инструменты были успешно интегрированы и протестирована их совместная работа.

В ходе тестирования было выявлено, что использование WebRTC Grabber на компьютерах участников работает достаточно оптимально. При отсутствии запроса на захват рабочего стола и веб-камеры количество используемой оперативной памяти не превышало 5Mb и не более 5% ресурсов процессора (на тестируемом компьютере использовался процессор Intel Core i5-8400). При продолжительной передаче видео рабочего стола и веб-камеры количество используемой оперативной памяти повышалось до 500Mb, а ресурсов процессора до 10–15%. Таким образом, можно заключить, что использование данной программы не мешает участникам соревнований.

Так как система видеотрансляций была полностью протестирована локально, следующим шагом будет создание тестовой видеотрансляции на одной из интернет площадок. В данной работе рассмотрим трансляцию на площадке Twitch.

Для начала видеотрансляции на площадке Twitch достаточно зарегистрироваться на сайте twitch.tv. Затем в панель управления, а затем в категорию Трансляции. На этой веб-странице будет доступен для копирования Ключ основной трансляции.

Затем необходимо настроить OBS Studio. Для этого достаточно выбрать пункт меню File, а затем Settings. В открывшемся окне выбрать категорию Stream. В поле Service выбрать Twitch, а в поле Stream Key вставить полученный на Twitch ключ.

После этого все настройки завершены и можно нажимать кнопку Start Streaming в OBS Studio для начала трансляции. После этого трансляция будет запущена, а сама видеотрансляция будет доступна по адресу `twitch.tv/username`, где `username` — ваше имя пользователя на платформе Twitch.

В итоге мы получили полностью функционирующую систему для трансляции соревнования по программированию.

ЗАКЛЮЧЕНИЕ

В рамках данной работы были рассмотрены существующие инструменты для реализации системы видеотрансляции соревнований по программированию. В частности были рассмотрены такие приложения, как WebRTC Grabber, позволяющее захватывать видео рабочего стола участников и передавать его на отдельный сервер, ICPC Live V3, позволяющее создавать и настраивать оверлей на видеотрансляции, а также OBS Studio для видеотрансляции на популярные стриминговые площадки.

Вышеописанные приложения были успешно интегрированы и протестирована их совместная работоспособность. Были выявлены как отдельные преимущества приложений, так и их совокупные плюсы.

В итоге была реализована работоспособная система для видеотрансляций предстоящих соревнований по программированию, проводимых Саратовским государственным университетом. Эта система позволит увеличить охват заинтересованной аудитории и расширит ее.

Таким образом все задачи дипломной работы были решены. Цель достигнута. Развернутая система является важным инструментом для повышения эффективности, наглядности, уровня и качества проводимых соревнований.