УДК 003.295

Е.В. Обухова, маг.; Н.П. Шутько, доц. (БГТУ, г. Минск, РБ)

АНАЛИЗ КОДА ПОЛУЧЕНИЯ ИНФОРМАЦИИ ИЗ КОНТЕЙНЕРА MP4

В современном мире цифровых технологий мультимедийные контейнеры играют ключевую роль в хранении, передаче и воспроизведении мультимедийного контента. Получение информации из таких контейнеров позволяет анализировать структуру мультимедиа, обеспечивать корректное воспроизведение на различных устройствах и платформах, а также оптимизировать процессы обработки видео- и аудиоданных.

MP4 – это мультимедийный контейнер, основанный на формате QuickTime MOV [1]. Он поддерживает видео, аудио, субтитры и изображения. Преимущества MP4 включают высокую степень сжатия данных, компактность, широкую совместимость и поддержку различных типов мультимедиа.

Файлы MP4 обладают модульной структурой, в которой данные организованы в виде атомов. Каждый атом включает в себя заголовок размером 8 байт, содержащий информацию о размере и типе атома, а также непосредственно данные. Среди ключевых атомов можно выделить следующие:

- ftyp: определяет тип файла и его совместимость с различными стандартами;
- moov: содержит метаданные, связанные с фильмом, такие как частота кадров, продолжительность видео и конфигурация декодеров;
- mdat: хранит медиаданные, включая видео- и аудиокадры;
- stts, stsc, stsz: таблицы, описывающие временные выборки, фрагменты и их размеры;
- meta: включает дополнительные метаданные, которые могут быть полезны для расширенного анализа контента.

Знание структуры и организации данных в файлах MP4 имеет решающее значение для эффективной обработки и конвертации мультимедийного контента. На основе этой информации можно разработать программные решения, способные анализировать и обрабатывать мультимедийные потоки. Анализируемый программный код [2], части которого будут представлены далее, предназначен для рассмотрения структуры контейнера MP4, включая содержимое атомов и используемые кодеки. Это позволяет определить характеристики мультимедийного потока, облегчая декодирование и последующее кодирование.

Полученная информация используется для выбора подходящего

алгоритма обработки, обеспечивая корректное преобразование аудио- и видеопотоков с оптимальными параметрами качества и размера. Данный код реализован на языке программирования Python, что обеспечивает совместимость с существующим проектом и упрощает разработку за счет доступа к библиотекам для работы с мультимедийными данными. Для обработки бинарных данных используется библиотека struct, позволяющая упаковывать и распаковывать данные для извлечения метаданных, таких как частота кадров и параметры кодека. Рассмотрим основные функции анализируемой программы.

Функция find_boxes выполняет поиск атомов в файле MP4 и определяет их смещения. В результате работы функции формируется словарь, где ключами являются типы обнаруженных атомов, например, 'ftyp' или 'moov', а значениями – кортежи с абсолютными смещениями начала и конца каждого атома.

Функция examine_mp4, часть кода которой представлена в листинге 1, выполняет диагностику структуры MP4-файла, проверяя наличие критически важных атомов и извлекая информацию о них. Этот процесс позволяет убедиться в целостности контейнера, а также подготовить данные для дальнейшей обработки.

```
moov_boxes = find_boxes(f, boxes[b"moov"][0] + 8,
boxes[b"moov"][1])
print("Moov boxes:",moov_boxes)trak_boxes = find_boxes(f,
moov_boxes[b"trak"][0]+8, moov_boxes[b"trak"][1])
print(trak_boxes)
udta_boxes = find_boxes(f, moov_boxes[b"udta"][0]+8,
moov_boxes[b"udta"][1])
print(udta_boxes, end="\n\n")
```

Листинг 1 – Диагностика структуры MP4-файла

Для получения информации о кодеке видео в исходный код была добавлена функция find_stsd, анализирующая содержимое атома stsd. Этот атом содержит сведения о параметрах сжатия, используемом кодеке и других характеристиках видеопотока. Доступ к этим данным важен при конвертации мультимедийных файлов, поскольку он позволяет определить, какие параметры кодека необходимо адаптировать. Анализ атома stsd представлен в листинге 2.

```
f.seek(stsd_start)
stsd_data = f.read(stsd_end - stsd_start)
version = stsd_data[0]
entry_count = int.from_bytes(stsd_data[4:8], "big")
print(f"'stsd' Box Info: Version={version}, Entry Count={en-
try_count}")
offset = 8 for i in range(entry_count):
```

```
sample_description_size   =   int.from_bytes(stsd_data[off-
set:offset + 4],"big")
sample_type = stsd_data[offset + 4:offset + 8]
print(f"Sample  {i  +  1}:  Size={sample_description_size},
Type={sample_type.decode('utf-8')}")
offset += sample_description_size
```

Листинг 2 – Анализ атома stsd

Кроме функций анализа, в исходный код также были добавлены инструменты для управления файлами. В частности, функция copy_file, часть которой представлена в листинге 3, выполняет копирование видеофайла из одной директории в другую и возвращает его содержимое в виде переменной. Этот механизм полезен при автоматизированной обработке видео, конвертации и интеграции медиаконтента в игровые файлы.

```
if test_files:
if not os.path.exists(output_dir):
os.makedirs(output_dir)
for test_file in test_files:
output_file_path = os.path.join(output_dir, test_file)
test_dir_copy = os.path.join(test_dir, test_file)
print(output_file_path)
print(test_dir_copy)
file_data = copy_file_to_variable_and_output(test_dir_copy,
output_file_path)
if file_data:
examine_mp4(test_dir_copy)
print(f"File copied: {output_file_path}", end="\n\n")else:
print("Can't copy file", end="\n\n")
```

Листинг 3 – Копирование видеофайла

На рис. 1 отображен результат выполнения программного кода, представленного выше, получающего информацию из видеоролика, имеющего расширение mp4.

```
Output_folder_test\snow.mp4
Input_folder_test\snow.mp4
Found boxes: {b'ftyp': (0, 24), b'moov': (24, 9520), b'uuid': (9520, 20928), b'mdat': (20928, 20929),
b'dat\x00': (20929, 21294), b'\xc5/-\x08': (21294, 3226507351)}

Moov boxes: {b'mvhd': (32, 140), b'trak': (4238, 9456), b'udta': (9456, 9520)}
{b'tkhd': (4246, 4338), b'edts': (4338, 4374), b'mdia': (4374, 9456)}

'stsd' Box Info: Version=0, Entry Count=1
Sample 1: Size=75, Type=mp4a

Duration (sec): 20.778666666666666
{b'\xa9TIM': (9464, 9487), b'\xa9TSC': (9487, 9504), b'\xa9TSZ': (9504, 9520)}

File copied: Output_folder_test\snow.mp4
```

Рисунок 1 – Полученная информация из тестируемого видеоролика

Любая программа, независимо от уровня ее проработки, может содержать узкие места, которые снижают производительность, ограничивают гибкость или делают код уязвимым к ошибкам. Рассматриваемый код также имеет ряд узких мест, устранение которых позволит повысить его надежность, масштабируемость и производительность:

– неэффективная работа с памятью, что неэффективно для больших видеофайлов;

– недостаточная проверка структуры MP4-файла;

– жестко заданные значения, что снижает гибкость программы;

– ограниченное обрабатывание иерархии боксов. Код предполагает фиксированную иерархию боксов без учета вложенных структур;

– ограниченная поддержка треков.

Анализ и обработка данных контейнеров MP4 играют ключевую роль в современном мультимедийном контенте, обеспечивая эффективную работу с видео- и аудиофайлами. Представленный программный код выполняет функции извлечения данных и анализа структуры MP4, что облегчает процесс декодирования и перекодирования. Однако в текущей реализации есть несколько узких мест, которые могут ограничивать производительность и стабильность программы. Внедрение улучшений значительно повысит функциональность кода.

ЛИТЕРАТУРА

1. Обухова Е.В., Шутько Н.П. Сравнительный анализ алгоритмов видеокодирования THEORA, MPEG-4 и H.263 / Е.В. Обухова, Н.П. Шутько // Наука и творчество: вклад молодежи: материалы международной молодежной научно-практической конференции студентов, аспирантов и молодых ученых, Махачкала, 19-20 ноября 2024 г. – Махачкала, 2024 (в печати).

2. Examine MP4 files with Python only [Электронный ресурс] // Kaggle. – URL: https://www.kaggle.com/code/humananalog/examine-mp4-files-with-pyhon-only (дата обращения: 02.02.2025).

УДК 004.42

Я.А. Игнаткова, ст. преп.; А.Н. Шербакова ст. преп.
(БГТУ, г. Минск, РБ)

ТРЕНДЫ ВЕБ-ДИЗАЙНА В 2025 ГОДУ

Веб-дизайн является динамичной сферой, где ежегодно возникают новые тенденции, существенно изменяющие способы взаимодействия пользователя с веб-ресурсами. В 2025 году наблюдается эволюция ключевых аспектов веб-дизайна, среди которых можно выделить