

## Извлечение метаданных ID3 с помощью языка программирования Python

*Кизянов Антон Олегович*

*Приамурский государственный университет имени Шолом-Алейхема  
Студент*

### **Аннотация**

В данной статье рассмотрено место расположения метаданных в аудиофайлах и способы его извлечения и обработки.

**Ключевые слова:** MP3, ID3

### **Extracting ID3 metadata using the Python programming language**

*Kizyanov Anton Olegovich*

*Sholom-Aleichem Priamursky State University  
student*

### **Abstract**

This article discusses the location of metadata in audio files and how to extract and process them.

**Keywords:** MP3, ID3

Для обычных пользователей аудиофайл является только файлом со звуком, но для специалистов он представляет собой файл богатый дополнительными данными, такими как метаданными. В них могут храниться данные от имени исполнителя, до секретной шифровки.

Цель исследования – это демонстрация извлечения метаданных из аудиофайлов.

Ранее этим вопросом интересовались Р.Д. Апсалямова, А.В. Душкин, А.С. Кравченко, С.Н. Паньчев, С.Л. Сахаров развивали тему «Обеспечение информационной безопасности систем обработки данных путем поиска стеганографических вложений в метаданных аудиофайлов» [1] в которой рассмотрен алгоритм обнаружения стеганографических вложений в целях пресечения скрытых каналов передачи (утечки) информации в цифровых системах обработки данных. А.С.Кольцов, В.А.Печенкин, С.Л.Сахаров с темой «Анализ тегов ID3 V2.3 MP3 файловна предмет наличия нелегальных вложений» [2], а подробнее про методику поиска скрытой передачи информации в медиафайлах на основе статистического анализа содержимого кадров. С.Л.Сахаров, В.И.Сумин, Я.Г.Долматова опубликовали статью «Исследование и обнаружение стеговложений при передаче медиафайлов» [3] рассказали результаты анализа одиночного файла, имеющего

одновременно как нескрываемые («легальные»), так и стеганографические вложения.

Метаданные ID3 очень часто связаны с файлами MP3. Существуют две версии структуры: ID3v1 и ID3v2. Версия ID3v1 является окончательными 128 байтами файла. Более новая версия, которая будет использоваться в статье находится в начале файла и имеет переменную длину.

Тег ID3 имеет более простую структуру по сравнению с тегами EXIF. Первые 16 байтов равномерно распределены между идентификатором тега и длиной метаданных.

Подпись файлов MP3 - это ASCII «ID3». После подписи, следуют различные теги, такие как TP1, TP2, и TCM. Это теги метаданных для исполнителя, группы и композитора, соответственно. Следующие 8 байтов после TP1 - это длина, обозначенная шестнадцатеричным 0x0B или 11. После 2-байтового буфера, это данные для автора, ранее известного как «Исполнитель». Хотя «Художник» имеет длину 10 байт с дополнительным одиночным нулевым байтом (0x00), добавленным к данным в общей сложности 11 байтов. В статье будет использоваться модуль с именем Mutagen для загрузки файла и чтения любых существующих тегов ID3.

Некоторые файлы MP3 могут не иметь встроенных метаданных ID3. В этом случае теги могут отсутствовать.

Mutagen (версия 1.31) способна читать и записывать разные аудио форматы метаданных. Mutagen поддерживает широкий спектр встроенных аудио форматов, таких как ASF, FLAC, M4A и MP3 (ID3). Чтобы установить библиотеку нужно ввести следующую команду в консоль:

```
pip install mutagen
```

Использовать Mutagen просто. Нужно создать объект ID3, открыть MP3-файл, потом искать определенные теги в словаре следующим образом:

```
from mutagen import id3
id = id3.ID3('music.mp3')
if 'TP1' in id.keys():
    print 'Artist:', id['TP1']
```

Функция id3Parser, проверяет подпись файла, а затем вызывает getTags() функцию. Функция getTags() опирается на mutagen модуль для разбора тегов MP3 и ID3.

```
import os
from time import gmtime, strftime
```

```
from mutagen import mp3, id3
import processors
def id3Parser():
def getTags():
```

Как работает функция id3Parser ()

Формат MP3 имеет только одну подпись файла 0x494433. Когда вызывается функция checkHeader(), сравнивается сигнатура файла. Если сигнатуры совпадают, управление передается функции getTags().

```
def id3Parser(filename):
```

```
    """
```

Функция id3Parser подтверждает тип файла.

:param filename: имя файла, потенциально содержащего метаданные.

:return: словарь из getTags, содержащий встроенные метаданные.

```
    """
```

Следующий код проверяет подпись файла.

```
signatures = ['494433']
if processors.utility.checkHeader(filename, signatures, 3) == True:
    return getTags(filename)
else:
    print 'Файловая подпись не соответствует известным MP3-
подписям.'
    raise TypeError('Файловая подпись не соответствует MP3.')
```

В функции getTags() нужно создать CSV-заголовки. Эти заголовки представляют собой возможные ключи, которыми может обладать словарь.

```
def getTags(filename):
```

```
    """
```

Функция getTags извлекает метаданные ID3 из объекта данных.

:param имя\_файла: путь и имя объекта данных.

:return: теги и заголовки, теги - словарь, содержащий метаданные ID3, а заголовки - это порядок ключей для выхода CSV.

```
    """
```

```
    header = ['Path', 'Name', 'Size', 'Filesystem CTime', 'Filesystem MTime',
'Title', 'Subtitle', 'Artist', 'Album', 'Album/Artist', 'Length (Sec)', 'Year',
```

```
'Category', 'Track Number', 'Comments', 'Publisher', 'Bitrate', 'Sample Rate',  
'Encoding', 'Channels', 'Audio Layer']
```

Создается tags словарь и заполняется некоторыми метаданными файловой системы, а именно:

```
tags = {  
    tags['Path'] = filename  
    tags['Name'] = os.path.basename(filename)  
    tags['Size'] = processors.utility.convertSize(os.path.getsize(filename))  
    tags['Filesystem CTime'] = strftime('%m/%d/%Y %H:%M:%S',  
    gmtime(os.path.getctime(filename)))  
    tags['Filesystem MTime'] = strftime('%m/%d/%Y %H:%M:%S',  
    gmtime(os.path.getmtime(filename)))
```

Mutagen имеет два класса, которые можно использовать для извлечения метаданных из файлов MP3. Первый класс, MP3 имеет некоторые стандартные метаданные, хранящиеся в файлах MP3, таких как битрейт, каналы и длина в секундах. Mutagen имеет встроенные функции для доступа к этой информации. Во-первых, нужно создать MP3 объект, используя функцию mp3.MP3(). Затем нужно использовать info.bitrate() функцию, например, для возврата битрейта файла MP3. Сохраняем эти значения в tags словаре следующим образом:

```
audio = mp3.MP3(filename)  
if 'TENC' in audio.keys():  
    tags['Encoding'] = audio['TENC'][0]  
tags['Bitrate'] = audio.info.bitrate  
tags['Channels'] = audio.info.channels  
tags['Audio Layer'] = audio.info.layer  
tags['Length (Sec)'] = audio.info.length  
tags['Sample Rate'] = audio.info.sample_rate
```

Второй класс, ID3 извлекает теги ID3 из файла MP3. Нужно сначала создать объект ID3 с помощью функции id3.ID3(). Это вернет словарь тегов ID3 в виде ключей.

```
{'TPE1': TPE1(encoding=0, text=[u'The Artist']),...}
```

Чтобы получить доступ к значению 'The Artist', нужно обработать значение в виде списка и указать элемент в нулевом индексе.

Аналогичным образом ищется каждый интересующий тэг и сохраняется первый элемент в значении tags словаря. В конце этого процесса возвращается обратно tags и header объекты обратно id3Parser(), что в свою очередь возвращает его скрипту:

```
id = id3.ID3(filename)
if 'TPE1' in id.keys():
    tags['Artist'] = id['TPE1'][0]
if 'TRCK' in id.keys():
    tags['Track Number'] = id['TRCK'][0]
if 'TIT3' in id.keys():
    tags['Subtitle'] = id['TIT3'][0]
if 'COMM::eng' in id.keys():
    tags['Comments'] = id['COMM::eng'][0]
if 'TDRC' in id.keys():
    tags['Year'] = id['TDRC'][0]
if 'TALB' in id.keys():
    tags['Album'] = id['TALB'][0]
if 'TIT2' in id.keys():
    tags['Title'] = id['TIT2'][0]
if 'TCON' in id.keys():
    tags['Category'] = id['TCON'][0]
if 'TPE2' in id.keys():
    tags['Album/Artist'] = id['TPE2'][0]
if 'TPUB' in id.keys():
    tags['Publisher'] = id['TPUB'][0]
return tags, header
```

### Вывод

Таким образом, можно передавать дополнительную информацию в метаданных аудиофайлов и извлекать её. А для различных аудиоплееров это будет полезным для извлечения битрейта, исполнителя и так далее.

### Библиографический список

1. Апсаямова Р.Д., Душкин А.В., Кравченко А.С., Паньчев С.Н., Сахаров С.Л. Обеспечение информационной безопасности систем обработки данных путем поиска стеганографических вложений в метаданных аудиофайлов // Современные наукоемкие технологии. 2016. №8-1 С. 13-17. URL: <https://elibrary.ru/item.asp?id=26397717> (Дата обращения: 26.01.2018)
2. Кольцов А. С., Печенкин В. А., Сахаров С. Л. Анализ тегов ID3 V2.3 MP3 файлов на предмет наличия нелегальных вложений // Уголовно-

исполнительная политика и вопросы исполнения уголовных наказаний 2016. С. 1041-1044. URL: <https://elibrary.ru/item.asp?id=28959469> (Дата обращения: 26.01.2018)

3. Сахаров С. Л., Сумин В. И., Долматова Я. Г. Исследование и обнаружение стеговложений при передаче медиафайлов // Пожарная безопасность: проблемы и перспективы. 2015. №2. С. 269-272. URL: <https://elibrary.ru/item.asp?id=26741336> (Дата обращения: 26.01.2018)