



seismix player

Оглавление

smeditor.....	1
Графический интерфейс.....	1
Скрипт «smeditor.lua»	3
seismix	4
Способы запуска	5
Хостовые функции	5
mci.play	5
mci.send	6
mci.pause	6
Предоставляемые возможности	6

Программа состоит из двух исполнимых модулей:

- seismix.exe – консольное приложение, позволяющее гибко заскриптовать последовательность воспроизведения звуковых файлов, а также их микширование.
- smeditor.exe – приложение с графическим интерфейсом, предоставляющее простой наглядный редактор для списков воспроизведения.

smeditor

Модуль состоит из 3 файлов:

- smeditor.exe – исполнимый файл, показывающий графический интерфейс; хостовая программа для нижеописанных скриптов.
- smeditor.lua – скрипт, позволяющий перепрограммировать некоторые возможности модуля. Скрипт написан в кодировке UTF-8 (без BOM).
- keyfiles.lua – упрощённый список ассоциаций идентификаторов и звуковых файлов. Скрипт написан в кодировке UTF-8 (без BOM).

Графический интерфейс

Окно программы состоит из двух секций: проигрыватель и список ассоциаций. Начать работу нужно с заполнения списка ассоциаций, после чего можно составлять и проигрывать плейлисты.

Проигрыватель. В этой секции содержатся элементы управления, позволяющие воспроизводить и тестировать плейлисты.

Имеется длинное поле редактирования, где можно набирать плейлисты. Синтаксис плейлистов простой: это перечень идентификаторов.

Справа от поля есть две кнопки:

- «Играть/Пауза» – позволяет запустить на воспроизведение плейлист слева от этой кнопки. Если плейлист уже играет, то кнопка позволяет поставить на паузу – временная приостановка воспроизведения; возобновить с места остановки можно повторным нажатием на эту кнопку.
- «Стоп» – позволяет остановить воспроизведение без возможности дальнейшего возобновления с места остановки.

Во время воспроизведения запрещается удаление записей из списка ассоциаций, а также изменение идентификаторов. Это сделано во избежание ошибок воспроизведения. Добавлять новые записи во время воспроизведения и редактировать идентификаторы, которых нет в воспроизводимом плейлисте – можно.

Можно перетащить мышью из проводника TXT-файлы, содержащие плейлист, на окно программы, при этом они будут добавлены в выпадающий список поля редактирования плейлиста. При открытии, весь плейлист из каждого TXT-файла преобразуется в строку, переводы на новые строки, как и другие непечатные символы при этом вырезаются.

Список ассоциаций. Этот список содержит сопоставления звуковых файлов с идентификаторами, из которых строится плейлист. В левой колонке отображаются идентификаторы, в правой – звуковые файлы. Идентификаторы и файлы можно отсортировать в алфавитном порядке, в прямом или обратном, если кликнуть мышью по заголовку столбца. Сортировка при отображении списка – без учёта регистра букв.

Создать новые записи в списке можно двумя способами:

- 1) Выделить в проводнике нужные звуковые файлы, и перетащить их мышью на окно программы. Чтобы выделить группу файлов в проводнике, нужно кликать мышью по ним при нажатой кнопке Ctrl, либо выделить резиновым контуром. Выделить все файлы в папке можно нажатием на Ctrl+A.
- 2) Нажать на кнопку «Открыть аудиофайлы...», выделить в открывшемся диалоге выбора файлов один или несколько файлов, нажать на «Открыть». В данном случае в диалоге открытия можно открыть сразу несколько файлов.

Новые записи в списке ассоциаций создаются только при двух условиях:

1. Файлы имеют допустимые расширения звуковых файлов.
2. Файлов со сгенерированными на основе их имён идентификаторами – ещё нет в списке.

Если необходимо создать две записи с одним и тем же файлом, но разными идентификаторами, следует сначала изменить идентификатор, присвоенный по умолчанию у первой записи, потом создать вторую.

Идентификаторы представляют собой набор чувствительных к регистру ASCII-символов. При создании новых записей в листе, идентификаторы создаются на основе имён файлов. Чтобы изменить идентификатор, нужно выделить в списке соответствующий пункт, и кликнуть по нему мышью, при этом слева должно появиться поле для редактирования. В этом поле нужно набрать новый идентификатор и нажать ввод, либо кликнуть в другое место. Нажатием на кнопку «Esc» редактирование будет отменено. Если был набран идентификатор, который уже есть в списке, то будет восстановлен прежний. Набранные при редактировании идентификатора пробелы заменяются знаками подчёркивания.

Примечание. Поскольку идентификаторы представляют собой последовательность однобайтных символов, то в качестве этих символов не могут быть Unicode-символы (например, из других языков, математические и т. п.). Однако, если в системе установлена русская локализация по умолчанию, то идентификаторы могут содержать также русские буквы, для которых будет

использоваться ANSI-кодировка (то есть использование КОИ-8 для русских букв).

Справа от списка имеются несколько кнопок:

- «Путь» – если кнопка вдавлена, позволяет отображать в списке полный путь к файлу; если кнопка не вдавлена, отображается только одно имя файла без пути.
- «Расширения» – если кнопка вдавлена, то отображаются в списке расширения файлов; если кнопка не вдавлена, то расширения скрываются.
- «Пропускать ошибки» – если кнопка вдавлена, то ошибки в плейлисте – пропускаются при воспроизведении без каких-либо сообщений. Если кнопка не вдавлена, то при обнаружении первого идентификатора, который невозможно воспроизвести, воспроизведение будет остановлено, и отобразится сообщение об ошибке. Ошибки могут быть двух видов: 1) указан идентификатор (символ), которому ничего не сопоставлено, в этом случае нужно добавить запись в список ассоциаций, либо удалить идентификатор из плейлиста; 2) ошибка при воспроизведении файла, она может возникнуть, например, при указании файла, который невозможно воспроизвести, либо при других системных ошибках.
- «Удалить файл из списка» – нажатием на эту кнопку можно удалить выделенные в списке файлы. Выделить и удалить можно, как только один файл, так и группу файлов. Выделить группу файлов можно, кликая по пунктам списка при нажатой клавише Ctrl; можно выделить список, кликая при нажатой клавише Shift. Кроме того, пункты выделяются автоматически при задержке курсора мыши над пунктом.

Под списком ассоциаций имеются две кнопки:

- «Открыть аудиофайлы...» – при нажатии на эту кнопку откроется диалог открытия файлов, с помощью которого можно выбрать один или несколько файлов для создания новых записей в списке ассоциаций.
- «Сохранить ассоциации» – финальная кнопка работы с программой. По завершению редактирования, нужно нажать на эту кнопку, чтобы сохранить список воспроизведения в скрипт. Список сохраняется всегда в один и тот же файл «keyfiles.lua», который в дальнейшем можно редактировать. Это исполнимый скрипт, который выполняется при запуске программы.

Скрипт «smeditor.lua»

Примечание № 1. Этот раздел содержит сведения о возможностях изменения функциональности программы. Перед внесением изменений в алгоритмы скрипта, рекомендую сначала ознакомиться с языком [Lua](#).

Примечание № 2. Для редактирования этого скрипта рекомендую использовать скриптовый редактор [Nodepad++](#), поскольку есть подсветка синтаксиса Lua и поддержка кодировок. Скрипт написан в кодировке UTF-8 (без BOM).

В начале скрипт содержит перечень сообщений всплывающих подсказок, здесь можно изменить текст сообщений на какие-либо другие, или изменить язык сообщений.

Далее следует список «ExtFilter», который определяет перечень фильтров расширений звуковых файлов для диалога открытия файлов, вызываемого нажатием на кнопку «Открыть аудиофайлы...». Здесь можно добавить фильтры для других форматов файлов, на которые установлены в системе аудиокодеки. Каждая запись состоит из двух строк: первая строка содержит отображаемое в диалоге сообщение; вторая строка определяет расширения для фильтрации файлов.

Если необходимо указать сразу несколько расширений, они должны разделяться точкой с запятой, например «*.aif;*.aiff».

Затем следует список «ExtAllowed» в котором определены расширения для разрешённых типов файлов. Именно по расширениям, указанным в этом списке, файлы фильтруются при добавлении в список ассоциаций. Например, если перетянуть из проводника группу файлов разных типов, то в список ассоциаций будут добавлены только те файлы, у которых расширение содержится в списке «ExtAllowed», а все остальные файлы будут проигнорированы.

Ниже содержится вспомогательная функция «function escapeLua(str)», она служит для экранирования обратных слэшей при записи строк в Lua-скрипт, это необходимо для корректной записи путей файлов.

Далее следует вспомогательная функция «function saveScript()», именно эта функция сохраняет список ассоциаций в файл «keyfiles.lua». Функция содержит полный алгоритм записи в скрипт.

Далее следует секция со скриптовой поддержкой обработки сообщений оконного интерфейса.

Функция «function JSeismix:wmDestroy()» вызывается при закрытии окна программы.

Функция «function JSeismix:wmClose()» вызывается при нажатии на правый верхний красный крестик. Здесь можно определить действия, которые должны при этом выполняться – закрытие окна программы, либо например игнорирование или вывод запроса.

Функция function «JSeismix:wmActivateApp(activated,idThread)» вызывается активации / деактивации окна программы.

Функция «function JSeismix:wmCommand(cmd,ctrl)» вызывается при каких-либо командах от оконного интерфейса программы. Здесь можно запрограммировать нужные действия от нужных элементов управления. Первым аргументом функции передаётся «cmd» – идентификатор элемента управления, в самом начале скрипта определяются константы этих идентификаторов. Численные значения констант задаются в исполнимом коде, и в скрипте присваивать другие значения – нельзя без перекомпиляции исполнимого модуля.

Функция «function JSeismix:wmHelp(ctrl,x,y)» вызывается, когда необходима подсказка по какому-либо элементу управления. Она вызывается при нажатии на знак вопроса в правом верхнем углу окна программы, и затем на элемент управления; либо при нажатии на F1, когда фокус находится на нужном элементе. Аргумент функции «ctrl» содержит идентификатор элемента управления, «x» – абсцисса курсора мыши, «y» – ордината курсора мыши. В данной реализации для нужного элемента управления выводится всплывающая подсказка с кратким описанием назначения элемента.

seismix

Модуль состоит из 3 файлов:

- seismix.exe – исполнимый файл, являющийся хостовой программой для выполнения нижеописанных скриптов.
- seismix.lua – скрипт, позволяющий перепрограммировать некоторые возможности модуля. Скрипт написан в кодировке UTF-8 (без BOM).
- keyfiles.lua – упрощённый список ассоциаций идентификаторов и звуковых файлов. Скрипт написан в кодировке UTF-8 (без BOM).

Вся основная логика программы вынесена в скрипт «seismix.lua».

Способы запуска

Программа является консольным приложением, с интерфейсом командной строки. Плейлист передаётся в параметрах командной строки в качестве аргумента. Возможно два варианта передачи:

1) Плейлист пишется непосредственно в аргументах командной строки. Например, так:

```
seismix.exe a1 c1 p a2 c2 c3 mx
```

2) Передаётся файл, в котором содержится плейлист. Например, так:

```
seismix.exe playlist.txt
```

Совет: проще всего запустить плейлист, если в проводнике зацепить мышью файл с плейлистом, и перетащить его на значок программы.

Если программа запущена без аргументов командной строки, то по умолчанию будет открыт плейлист, содержащийся в файле «playlist.txt».

Хостовые функции

Программа предоставляет скрипту три функции, позволяющие гибко запрограммировать воспроизведение и микширование файлов в любых комбинациях.

mci.play

Функция служит для проигрывания любых мультимедийных файлов, с указанием отрезка воспроизведения.

Формат записи:

```
mci.play(file,from,to)
```

Эта функция принимает 3 аргумента:

- file – имя и путь к файлу, который должен проигрываться. Эта строка должна заключаться в кавычки. Если в строке присутствуют обратные флешки "\", то они должны дублироваться (экранирование). Пример: "C:\\Sounds\\33.mp3"
- from – стартовая позиция проигрывания файла, в миллисекундах. Если необходимо проигрывание с самого начала, следует передать ноль. Если, например, необходимо воспроизвести начиная с 7-ой секунды, то передаётся значение 7000.
- to – конечная позиция воспроизведения файла, в миллисекундах. Если нужно воспроизвести до конца, то следует передать ноль. Если, например, нужно воспроизвести до 9-ой секунды, нужно передать 9000.

Пример использования:

```
mci.play("a1",0,0)
```

здесь будет проигрываться файл "a1.mp3" от начала до конца

```
mci.play("c3.mp3",500,2500)
```

здесь будет проигрываться файл "c3.mp3" начиная с 0,5 секунды до 2,5.

```
mci.play("c3.mp3",500,500+2000)
```

То же самое, что и в предыдущем примере, только конец воспроизведения задаётся

как начало плюс длина.

Возвращаемое значение: 0, если команда выполнена успешно, иначе код MCI-ошибки.

mci.send

Функция управления мультимедийным устройством.

Формат записи:

```
mci.send(command)
```

Эта функция принимает 1 аргумент:

- **command** – строка с командой, посылаемой мультимедийному устройству. Строка начинается с описателя команды, для проигрывания файла нужно написать "play", для паузы проигрываемого файла "pause", для возобновления после паузы "resume". Если воспроизводимый файл - большой, может понадобиться "stop" для останова. После описателя следует через пробел файл, для которого выполняется команда. Если эта команда - "play", то далее можно указать начало воспроизведения "from X", где X - время относительно начала файла в миллисекундах, и можно указать конец воспроизведения "to X", где X - время конца в миллисекундах. Далее можно указать атрибут "wait", который означает, что необходимо дождаться окончания выполнения команды. Например, в случае проигрывания, если есть "wait", то выполнение скрипта будет приостановлено, пока проигрывание не закончится; если "wait" не указан, то следующая функция будет выполнена сразу же после начала воспроизведения. Таким способом можно создавать простейшие миксы: запустить на воспроизведение один файл без атрибута "wait", а затем через паузу другой файл, который тоже можно смикшировать в свою очередь.

Вместо файла можно указать компакт-диск, и управлять его воспроизведением.

Пример использования:

```
mci.send("play c2.mp3 from 2000 to 5000 wait")
```

воспроизводит файл «c2.mp3» со 2-ой секунды по 5-ю.

Возвращаемое значение: 0, если команда выполнена успешно, иначе код MCI-ошибки.

mci.pause

Функция служит для вставки паузы.

Формат записи:

```
mci.pause(time)
```

Эта функция принимает 1 аргумент:

- **time** – время паузы, в миллисекундах. Например, чтобы сделать паузу в полсекунды, нужно передать значение 500.

Пример использования:

```
mci.pause(1000)
```

создаёт паузу в 1 секунду.

Предоставляемые возможности

В функции «function onMain(playlist)» скрипта «seismix.lua» осуществляется чтение плейлиста из файла, если передан файл в параметрах командной строки. Далее следует алгоритм разбора плейлиста на идентификаторы и их воспроизведение. Здесь можно изменить синтаксические правила под нужные требования. В данной реализации осуществляется чтение идентификаторов, следующих непрерывно, где более длинные идентификаторы имеют больший приоритет.

Таблица «KeyFuncs» содержит сопоставления идентификаторам функций. В отличие от таблицы «KeyFiles», в таблице «KeyFuncs» идентификаторам сопоставляются не файлы воспроизведения, а скриптовые функции, которые будут вызываться для соответствующих идентификаторов плейлиста. В этих функциях можно запрограммировать исполнение последовательности файлов в нужном порядке, и использовать их микширование.

Микширование звуков. Примеры микширования приведены в коде и комментариях скрипта. В общем случае, микширование сводится к запуску одного или нескольких звуков в асинхронном режиме (фоновом), дальше с использованием пауз и/или звуков в синхронном режиме можно добиться наложения звуков в нужной последовательности.

Управление устройством. Предоставляемая функция «mci.send» позволяет не только воспроизводить файлы, но также осуществлять полноценное управление любыми мультимедийными устройствами.

В качестве примера можно рассмотреть способ воспроизведения трека с компакт-диска из плейлиста, причём в фоновом режиме. Для этого сначала переменной «CDDevice» скрипта присваиваем букву дисковода с компакт-диском. Далее составляем плейлист. Сначала пишем идентификатор «cdo» для открытия компакт-диска как музыкального устройства. Затем пишем идентификатор «cd1», чтобы воспроизвести первый трек с компакт-диска. Поскольку этому идентификатору сопоставлено обращение к устройству со строкой «play cd from 1 to 2», будет воспроизведён 1-й трек в фоновом режиме. Далее пишем какие-либо другие идентификаторы звуков, которые должны воспроизводиться на фоне игры трека с компакт-диска, и в конце пишем идентификаторы «cds» и «cdc». Впрочем, последний писать – необязательно, если девайс использоваться всё равно не будет.

Итак, программа предоставляет гибкие возможности управления мультимедийными устройствами, в зависимости от содержимого плейлиста.