

Веб-приложения с применением технологии WebAssembly отличаются скоростью, безопасностью и другими полезными особенностями, характерными нативным приложениям. Осуществляется это за счёт того, что WebAssembly модули создаются при помощи низкоуровневых языков и архитектурных моделей, созданных для разработки нативных приложений. В связи с чем разработчику веб-приложений с применением технологии WebAssembly приходится решать задачи более широкого плана, чем без применения данной технологии.

Технология WebAssembly еще очень молода, что еще больше усложняет процесс создания таких приложений по ряду причин. Инструменты для компиляции WebAssembly модулей из низкоуровневых языков представляют собой часто обновляющиеся консольные приложения с отсутствием хорошей и полной документации. Бывали случаи, когда вышедшее обновление такого инструмента отказывалось собирать проект, который собирался предыдущей версией инструмента.

Часто при разработке веб-приложений с применением технологии WebAssembly, вместо написания всего кода на языке программирования низкого уровня, используются сторонние библиотеки с открытым исходным кодом, написанные в legacy стиле кода, с применением C и C++ кода в одном проекте, которые собираются посредством CMake, с применением Makefile, которые плохо воспринимаются набором средств для создания WebAssembly файлов Emscripten. В результате чего, приходится создавать правила компиляции с нуля, разбирая всю структуру исходного кода библиотеки. Так же, во время разбора исходного кода, для оптимизации размера итогового файла, частично исключается функционал библиотеки, оставляя только необходимые файлы. При такой компиляции приходится вручную изменять пути включения заголовочных файлов на относительные, следя какой именно файл требуется в том или ином случае.

Смешивание C и C++ файлов в одном проекте является частым явлением, но при попытке собрать такой проект, появляется следующее предупреждение:

```
1 warning generated.
clang++: warning: treating 'c' input as 'c++' when in C++ mode, this behavior is deprecated [-Wdeprecated]
```

После этого предупреждение появляется ошибка анализа файла и приходится переписывать некоторые места C кода в стиле C++:

```
227 217
228 - local int unzlocal_getlong OF((
229 + local int unzlocal_getlong (
230   const zlib_filefunc_def* pzlib_filefunc_def,
231   voidpf filestream,
232   - ulong *pX);
233 + ulong *pX);
234 - local int unzlocal_getlong (pzlib_filefunc_def,filestream,pX)
235 - const zlib_filefunc_def* pzlib_filefunc_def;
236 - voidpf filestream;
237 - ulong *pX;
238 + ulong *pX);
239 {
240   ulong x ;
241   int i = 0;
242   ...
243   @@ -263,9 +248,9 @@ local int unzlocal_getlong (pzlib_filefunc_def,fi
244
245   /* My own strcmpi / strcasecmp */
246   - local int strcmpcasensitive_internal (fileName1,fileName2)
247   - const char* fileName1;
248   - const char* fileName2;
249   + local int strcmpcasensitive_internal (
250   + const char* fileName1;
251   + const char* fileName2;
252   +
253   + const char* fileName2;
254   {
255
256   308 293 /*
257   309 + extern int ZEXPORT unzStringFileNameCompare (fileName1,fileName2,iCaseSensitivity)
258   310 - const char* fileName1;
259   311 - const char* fileName2;
260   312 - int iCaseSensitivity;
261   313 + extern int ZEXPORT unzStringFileNameCompare (
262   314 + const char* fileName1;
263   315 + const char* fileName2;
264   316 + int iCaseSensitivity);
265   317 {
266   318   if (iCaseSensitivity==0)
267   319     iCaseSensitivity=CASESENSITIVITYDEFAULTVALUE;
268   320   @@ -328,13 +313,9 @@ extern int ZEXPORT unzStringFileNameCompare (fileName1,fileName2,iCaseSe
269   321   Locate the central directory of a zipfile (at the end, just before
270   322   the global comment)
271   323   */
272   324   - local ulong unzlocal_SearchCentralDir OF((
273   325   + local ulong unzlocal_SearchCentralDir (
274   326   const zlib_filefunc_def* pzlib_filefunc_def,
275   327   voidpf filestream));
276   328   - local ulong unzlocal_SearchCentralDir(pzlib_filefunc_def,filestream)
277   329   - const zlib_filefunc_def* pzlib_filefunc_def;
278   330   - voidpf filestream;
279   331   + voidpf filestream)
280   332   {
```

Некоторые часто используемые библиотеки(например zlib) уже встроены в средства сборки и оптимизированы для WebAssembly, поэтому встречая такие библиотеки необходимо заменять их реализации подключением встроенной версии этой библиотеки:

```
C:\code\tools\emscripten>
-s USE_ZLIB=1 ^
```

Для сборки WebAssembly модуля из C++ кода, необходимо запустить компилятор em++ и передать в качестве аргументов командной строки список путей для всех задействованных в проекте файлов исходных кодов, а также указать путь к выходному файлу с флагом «-o». При большом количестве файлов, компиляцию рекомендуется проводить в несколько этапов из-за ограничений компилятора на количество входных файлов. При этом выходной файл имеет расширение «.o» и

используется как входной файл на следующем этапе компиляции. Рассмотрим данный процесс на примере компиляции библиотеки для работы с 3D-моделями Assimp:

compile1.bat:

```
em++ c-code/assimp-5.0.1/code/Common/BaseProcess.cpp ^
c-code/assimp-5.0.1/code/Common/BaseImporter.cpp ^
c-code/assimp-5.0.1/code/Common/Bitmap.cpp ^
c-code/assimp-5.0.1/code/Common/CreateAnimMesh.cpp ^
c-code/assimp-5.0.1/code/Common/DefaultIOSystem.cpp ^
c-code/assimp-5.0.1/code/Common/DefaultLogger.cpp ^
c-code/assimp-5.0.1/code/Common/Exporter.cpp ^
c-code/assimp-5.0.1/code/Common/Importer.cpp ^
c-code/assimp-5.0.1/code/Common/ImporterRegistry.cpp ^
c-code/assimp-5.0.1/code/Common/PostStepRegistry.cpp ^
c-code/assimp-5.0.1/code/Common/RemoveComments.cpp ^
c-code/assimp-5.0.1/code/Common/scene.cpp ^
c-code/assimp-5.0.1/code/Common/SceneCombiner.cpp ^
c-code/assimp-5.0.1/code/Common/ScenePreprocessor.cpp ^
c-code/assimp-5.0.1/code/Common/SGSpatialSort.cpp ^
c-code/assimp-5.0.1/code/Common/SkeletonMeshBuilder.cpp ^
c-code/assimp-5.0.1/code/Common/SpatialSort.cpp ^
c-code/assimp-5.0.1/code/Common/StandardShapes.cpp ^
c-code/assimp-5.0.1/code/Common/Subdivision.cpp ^
c-code/assimp-5.0.1/code/Common/Version.cpp ^
c-code/assimp-5.0.1/code/Common/ZipArchiveIOSystem.cpp ^
-O3 -r -o temp/common.o
```

compile.bat:

```
call compile1.bat
em++ -s ALLOW_MEMORY_GROWTH=1 ^
-s EXPORT_ES6=1 ^
-s MODULARIZE=1 ^
-s SINGLE_FILE=1 ^
-s USE_ES6_IMPORT_META=0 ^
-s ENVIRONMENT=node,web ^
-O3 ^
-std=c++11 ^
-s EXPORTED_FUNCTIONS=['_malloc','_free','_main'] ^
-s EXTRA_EXPORTED_RUNTIME_METHODS=['ccall'] ^
-s USE_ZLIB=1 ^
-o js-code/converter/converter_core.js c-code/main.cpp ^
temp/common.o ^
c-code/importer.cpp ^
c-code/assimp-5.0.1/code/Material/MaterialSystem.cpp ^
c-code/assimp-5.0.1/code/PostProcessing/ValidateDataStructure.cpp ^
c-code/assimp-5.0.1/code/PostProcessing/ProcessHelper.cpp ^
c-code/assimp-5.0.1/code/STL/STLLoader.cpp ^
c-code/assimp-5.0.1/code/STL/STLExporter.cpp ^
c-code/assimp-5.0.1/code/Obj/ObjExporter.cpp ^
c-code/assimp-5.0.1/code/Obj/ObjFileImporter.cpp ^
c-code/assimp-5.0.1/code/Obj/ObjFileMtlImporter.cpp ^
c-code/assimp-5.0.1/code/Obj/ObjFileParser.cpp ^
c-code/assimp-5.0.1/code/Collada/ColladaExporter.cpp ^
c-code/assimp-5.0.1/code/X/XFileExporter.cpp ^
c-code/assimp-5.0.1/code/X/XFileImporter.cpp ^
c-code/assimp-5.0.1/code/X/XFileParser.cpp ^
c-code/assimp-5.0.1/code/Step/StepExporter.cpp ^
c-code/assimp-5.0.1/code/Ply/PlyExporter.cpp ^
```

...

```
c-code/assimp-5.0.1/code/X3D/X3DImporter_Shape.cpp ^
c-code/assimp-5.0.1/code/X3D/X3DImporter_Light.cpp ^
c-code/assimp-5.0.1/code/X3D/X3DImporter_Geometry2D.cpp ^
c-code/assimp-5.0.1/code/X3D/X3DImporter_Geometry3D.cpp ^
c-code/assimp-5.0.1/code/X3D/X3DImporter_Rendering.cpp ^
c-code/assimp-5.0.1/code/X3D/X3DImporter_Group.cpp ^
c-code/assimp-5.0.1/code/X3D/X3DImporter_Networking.cpp ^
c-code/assimp-5.0.1/code/X3D/X3DImporter_Texturing.cpp ^
c-code/assimp-5.0.1/code/X3D/X3DImporter_Postprocess.cpp ^
c-code/assimp-5.0.1/code/X3D/X3DImporter_Metadata.cpp ^
c-code/assimp-5.0.1/code/X3D/FIReader.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXImporter.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXBinaryTokenizer.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXParser.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXDocument.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXDocumentUtil.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXConverter.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXMeshGeometry.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXDeformer.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXAnimation.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXNodeAttribute.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXProperties.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXModel.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXMaterial.cpp ^
c-code/assimp-5.0.1/code/FBX/FBXTokenizer.cpp ^
```

После того, как необходимый проект собрался, нужно продумать характеристики итогового модуля, найти и прописать необходимые флаги компиляции, что тоже не является тривиальной задачей в виду отсутствия хорошей документации. Так же на этом шаге необходимо продумать API взаимодействия JavaScript-кода и WebAssembly-кода, и вывести «наружу» соответствующие функции из модуля WebAssembly:

```
em++ -s ALLOW_MEMORY_GROWTH=1 ^
-s EXPORT_ES6=1 ^
-s MODULARIZE=1 ^
-s SINGLE_FILE=1 ^
-s USE_ES6_IMPORT_META=0 ^
-s ENVIRONMENT=node,web ^
-o3 ^
-std=c++11 ^
-s EXPORTED_FUNCTIONS=['_malloc', '_free', '_main']" ^
-s EXTRA_EXPORTED_RUNTIME_METHODS=['ccall']" ^
```

Следующий этап — это написать взаимодействие высокоуровневого JavaScript с низкоуровневым C++ используя для этого всего 4 типа данных, называемое «клеевым кодом». На стороне JavaScript необходимо реализовать полноценное ручное управление памятью, с учетом того, что этот язык, при создании, не предполагал таких возможностей, а так же превращать и упаковывать данные из высокоуровневых объектов в линейную память, к которой имеет доступ WebAssembly-модуль написанный на низкоуровневом языке, а также выполнять обратную операцию.

При этом необходимо следить, чтоб данные на стороне JavaScript всегда читались верными, и при этом не было «утечек памяти», что довольно сложно с учетом того, что одной и той же памятью(даже одними и теми же ячейками памяти) пользуются язык с ручной и автоматической очистками памяти.

В общем, примерно так страдают люди, которых угораздило заняться WebAssembly.