Emscripten: C++ для web

Andrew Karpushin

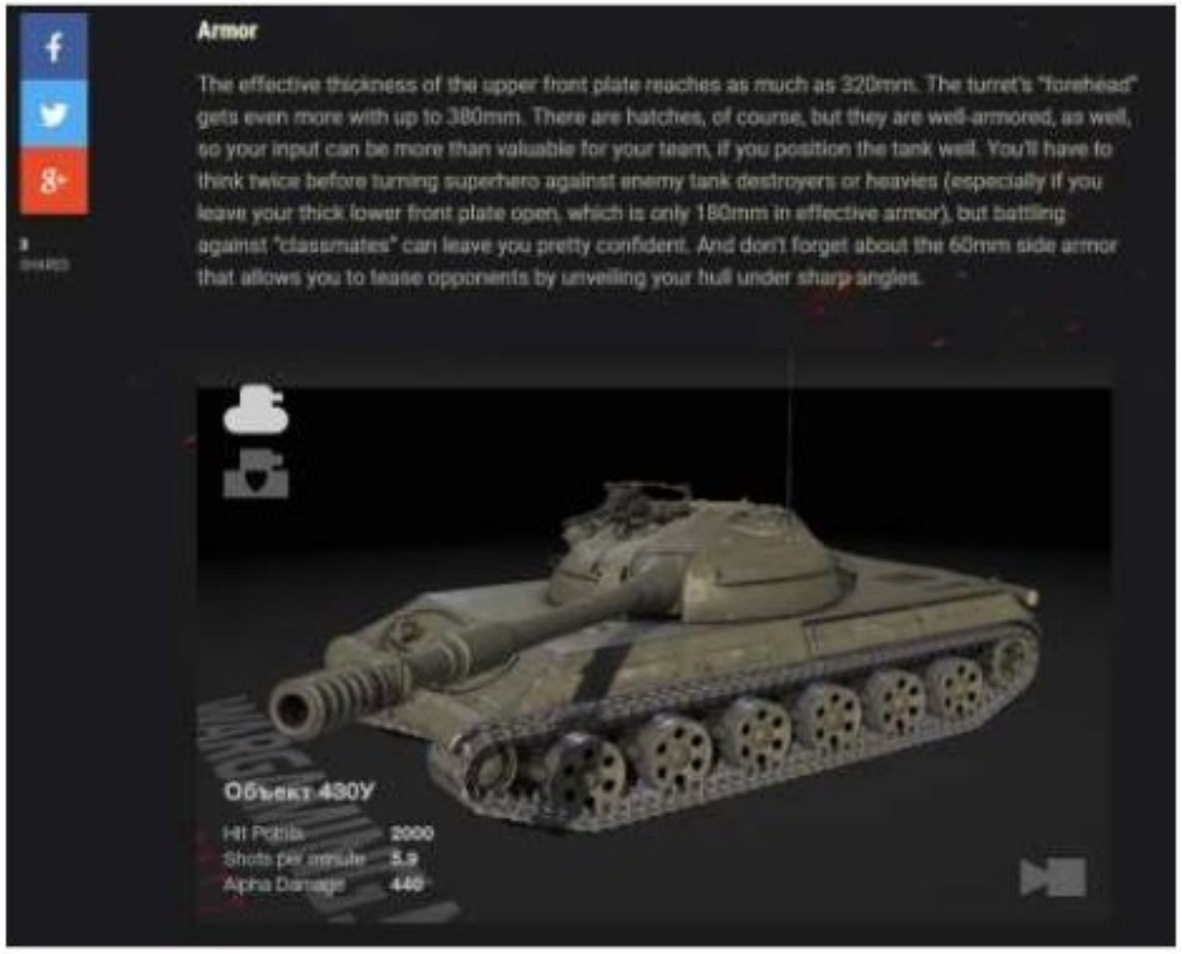
О чем презентация

- О практическом опыте портирования приложений на C++ в веб
- О проблемах, с которыми я столкнулся в процессе
- О том какие есть ограничения, что работает, а что нет

Обо мне

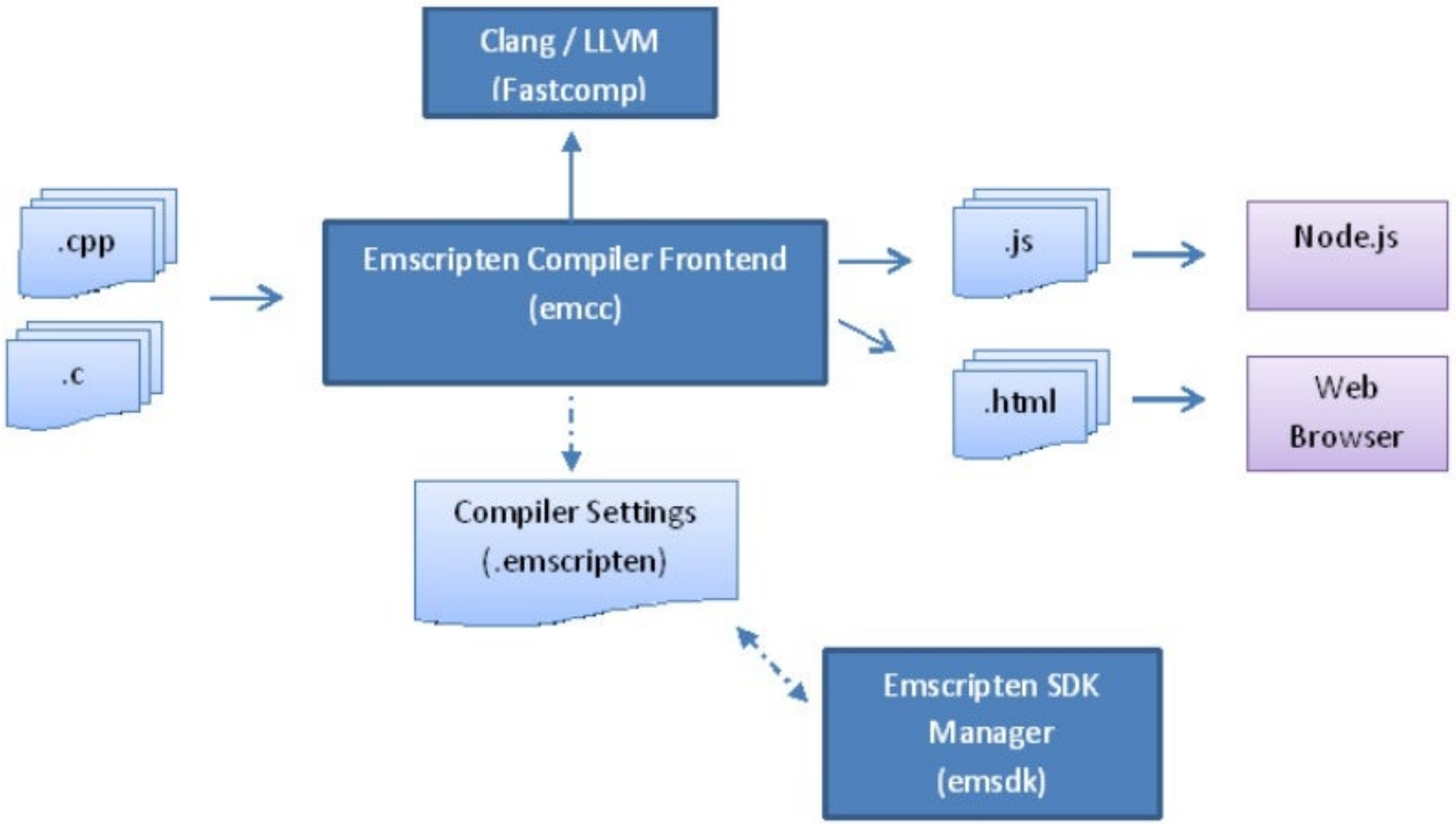
- 5 лет разработки игр на C++ (PC, Xbox)
- 5 лет в serious games (Gamebryo, Unity3D)
- 4 года indie, мобильные и веб приложения/сервисы

Примеры


Armor
The effective thickness of the upper front plate reaches as much as 320mm. The turret's "forehead" gets even more with up to 380mm. There are hatches, of course, but they are well-armored, as well, so your input can be more than valuable for your team, if you position the tank well. You'll have to think twice before turning superhero against enemy tank destroyers or heavies (especially if you leave your thick lower front plate open, which is only 180mm in effective armor), but battling against "classmates" can leave you pretty confident. And don't forget about the 60mm side armor that allows you to tease opponents by unveiling your hull under sharp angles.
Объект 430У
Hit Points 2000
Shots per minute 5.9
Alpha Damage 440


wot inspector.com
map inspector pc
реплеи
реплей остановлен

emscripten


Clang / LLVM
(Fastcomp)
.cpp
.c
Emscripten Compiler Frontend
(emcc)
.js
Node.js
.html
Web
Browser
Compiler Settings
(.emscripten)
Emscripten SDK
Manager
(emsdk)

- Поддержка C++ runtime
- C++ standard library
- C++ exceptions
- SDL
- OpenGL
- OpenAL

Не полностью поддерживаются:

- threads
- sockets

Среда выполнения

- нет доступа к файловой системе
- никакой информации о других процессах в системе
- ограничения для HTTP запросов (CORS)
- основной цикл, в котором выполняется приложение
- float -> double

Сложности с запуском двух приложений на одной web-странице.

Вызов C++ из JavaScript

C++:

```
extern “C” int multiply(int a, int b) { return a*b; }
```

JavaScript:

```
console.log(Module.ccall(‘multiply’, ‘number’, [‘number’, ‘number’], [5, 6]));
```

Есть еще **embind** и **WebIDL**.

JavaScript из C++

Простой способ:

```
emscripten_run_script("alert('hi')");
EM_ASM({ window.location.reload(); });
EM_ASM_({ Module.print('the number is' + $0); }, 100);
```

Другой способ:

foo.h:

```
extern "C"
{
    extern void bar(const char* text);
}
```

foo.js:

```
mergeInto(LibraryManager.library, {
        bar:function (text) {...}
});
```

JavaScript из C++ (строки)

- EM_ASM_({
  window.location.replace(**Module.Pointer_stringify**($0));
  }, requestUrl.c_str());

- int stringPointer = EM_ASM_INT_V({
  var url = window.location.href;
  return **allocate(intArrayFromString(url), 'i8', ALLOC_STACK)**;
  });
  const char * url = reinterpret_cast<const char *>(stringPointer);

Пример: file uploads (C++)

```
extern "C"
{
    void uploadReplay(const uint8_t * buffer, unsigned length)
    {
            ...
    }
}
```

Пример: file uploads (JavaScript)

```
<input type="file" id="fileupload" onchange="processFile()" accept=".wotreplay"/>


function processFile(){
    var file = document.getElementById('fileupload').files[0];
    var reader = new FileReader();
    reader.onload = function(e) {
        var bytes = reader.result;
        var dataPtr = Module._malloc(bytes.byteLength);
        var dataHeap = new Uint8Array(Module.HEAPU8.buffer, dataPtr, bytes.byteLength);
        dataHeap.set(new Uint8Array(bytes));

        Module.ccall('uploadReplay', null, ['number', 'number'], [dataHeap.byteOffset,
dataHeap.length]);
        Module._free(dataHeap.byteOffset);
    }
    reader.readAsArrayBuffer(file);
}
```

Файловая система

- MEMFS
  Виртуальная файловая система, но изменения не сохраняются

- NODEFS
  Доступ к реальной файловой системе, но только под Node.js

- IDBFS
  Позволяет хранить данные постоянно, но доступ асинхронный

IDBFS

Необходимо изменение кода для асинхронного чтения\записи в IDBFS

До:

```
loadSettings();
doSomething();
```

После:

```
emscripten_idb_async_load(..., ..., ..., &loadCallback, &errorCallback);

void loadCallback(void * arg, void * buffer, int bufferSize)
{
    loadSettings();
    doSomething();
}

void errorCallback(void * arg)
{
    doSomething();
}
```

Сборка

- **emconfigure**, **emmake**, **emcmake**
- код, оптимизированный по размеру может выполняться быстрее, чем оптимизированный по скорости
- можно компилировать в WebAssembly
  - поддерживается не всеми браузерами
- множество флагов компиляции
  - размер heap
  - экспериментальные функции (posix threads, asyncify, emterpreter, etc.)
  - линковка с другими JavaScript библиотеками
  - линковка с виртуальной файловой системой
  - опции отладки

Отладка

```
▶Uncaught abort(9) at Error
    at jsStackTrace (http://localhost/map-inspector.js:1:26132)
    at stackTrace (http://localhost/map-inspector.js:1:26303)
    at abort (http://localhost/map-inspector.js:24:13315)
    at k7b (http://localhost/map-inspector.js:17:21918)
    at Iab (http://localhost/map-inspector.js:6:247836)
    at V4a (http://localhost/map-inspector.js:6:122716)
    at k4a (http://localhost/map-inspector.js:6:112558)
    at O0a (http://localhost/map-inspector.js:6:57910)
    at m1a (http://localhost/map-inspector.js:6:66671)
    at f5b (http://localhost/map-inspector.js:17:15834)
```

Отладка

- сохранение имен функций
- подключение C++ source map
- runtime checks
  - ошибки доступа к памяти
  - переполнение стека
- stackTrace()
- логи

Другие темы

- Web Audio API
- WebGL
- WebSockets
- WebVR
- профайлинг, Emscripten tracing API
- использование в Node.js

Резюме

- C++ код может работать под web с минимальными изменениями
- Основные проблемы:
  - файловая система
  - сеть

Для удобства приложение должно стартовать как можно раньше и подгружать ресурсы по мере необходимости.

Полезные ссылки

- kripken.github.io
  - http://kripken.github.io/emscripten-site/docs/getting_started/Tutorial.html
  - http://kripken.github.io/emscripten-site/docs/api_reference/index.html
- обсуждение
  - https://groups.google.com/forum/#!forum/emscripten-discuss
- примеры
  - https://github.com/learnopengles/airhockey
  - https://github.com/inolen/quakejs
- демо
  - https://github.com/kripken/emscripten/wiki/Porting-Examples-and-Demos

Спасибо за внимание

Вопросы

Andrew Karpushin ✉ andrew@wotinspector.com