Porting to the Web

An Introduction to Emscripten

Ting-Yuan Huang
thuang@mozilla.com

Web as the Platform

- HTML5
- JavaScript, DOM, CSS
- <video>, <audio>, <canvas>
- WebSocket, WebGL, WebRTC, WebWorker, IndexedDB, Filesystem API, Geolocation API, etc.

The Good Old Things

- Existing codes, tools and developers
- Supporting other languages
- Performance

Compiling to JavaScript

- Flascc, GWT, Emscripten
- JavaScript is standard based (not only language, but also APIs)
- It's the only language that runs on every browser
- JavaScript is getting faster these years

Demo
Unreal Engine 3

Emscripten

C/C++

=> (clang) => LLVM bitcodes

=> (optimizations) => LLVM bitcodes

=> (Emscripten) => JavaScript

- Developed by Dr. Alon Zakai
- Compiles LLVM bitcodes to JavaScript
- Implements C/C++ runtime libraries
- Implements SDL, OpenGL, EGL, etc.
- Provides various utilities

Memory Model

- A large, unified ArrayBuffer as the C/C++ HEAP
```
var buffer = new ArrayBuffer(TOTAL_MEMORY);
```
- Various views to HEAP
```
HEAP32 = new Int32Array(buffer);
HEAPU8 = new Uint8Array(buffer);
```
- Memory addresses / Pointers as offsets

Types

- Integer arithmetics
  ```
  (x + y) | 0
  ```
- Load a 32-bit integer from memory
  ```
  HEAP32[p >> 2] | 0
  ```

Hello, World

```
#include <stdio.h>
int main()
{
    printf("Hello");
    return 0;
}

> emcc hello.c
> js a.out.js
Hello
```

```
Allocate([72,101,108,111,44]\
   /*"Hello"*/,…, 5242880);

function main()
{
    _puts(5242880);
    return 0;
}
```

Hello, strlen()

```
function strlen() {
    ptr = ptr | 0;
    var curr = ptr;
    while (HEAPU8[curr] | 0 != 0) {
        curr = (curr + 1) | 0;
    }
    return (curr - ptr) | 0;
}
```

Performance

- No objects, properties
- No dynamic typing
- No garbage collection
- Only numbers, functions and local variables.
- LLVM optimization passes
- Co-designs

asm.js

- An extraordinarily optimizable, low-level subset of JavaScript
- Fully compatible to standard JavaScript; It's a subset!
- “use asm”;
  - Optional to JavaScript engines
  - Ahead-of-Time compilation
  - Eliminating runtime type checks, boxing/unboxing

OdinMonkey

- An JavaScript engine recognizing “use asm”;
- Runs within 2x times compared to native!
- Still improving
- Enabled by default in Firefox 22

Limitations

- Multithread
- Local event loops
- Memory
- Non-portable codes

Projects Using Emscripten

- Unreal Engine 3
- BananaBread
- UNIGINE
- Emscripten-QT
- lzma.js
- sql.js

Thanks!