
Subject: Re: About WebAssembly
Posted by [Tom1](#) on Tue, 09 Feb 2021 11:42:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Tue, 09 February 2021 00:00mirek wrote on Mon, 08 February 2021 14:49Novo wrote on Mon, 08 February 2021 19:35AFAIK, WebAssembly requires all data to be aligned by at least the word size. This can be a problem ...

Even char? Like you cannot do `char *s = ...; *s++?`

Probably, you can do that. Last time I had to deal with compiling C++ to Web was, probably, ~5 years ago. And that was Emscripten.

The only thing I remember for sure I had to properly align unaligned data structures, otherwise we were getting runtime exceptions. Part of the code had to be disabled because of problems with data alignment.

This was ~FIVE years ago. Life has changed since that time. And we have WebAssembly in addition to Emscripten now.

Another thing: compilation with Emscripten wasn't a problem at all.

P.S. I personally would prefer to use Turtle because of security reasons. WebAssembly can be easily decompiled, and in case of Turtle the only thing people can steal is a picture in a Web-browser. :roll:

Hi,

Thanks for your response. Yes, I agree very much: Turtle is and remains the solution for code to be kept safe. However, to move the computational load to the client side requires different approach. (Maybe even a hybrid approach, where GUI runs on the client side and the critical algorithms on the server side out of reach.)

Anyway, I tried out the structure alignment and it seems to work just fine like in MSC/CLANG/GCC when using `#pragma pack(push,1)`:
`#include <stdio.h>`

```
#pragma pack(push,1)
```

```
typedef struct{  
    char  a;  
    short b;  
    int   c;  
}struct_t;
```

```
#pragma pack(pop)
```

```
int main(){  
    char buffer[]={0,1,2,3,4,5,6,7,8,9,10,11};  
    struct_t &s=(struct_t*)buffer;
```

```
    printf("a = %XH\n",s.a);  
    printf("b = %XH\n",s.b);  
    printf("c = %XH\n",s.c);  
    return 0;  
}
```

The result is:

```
tom@TomVM:~/test$ emcc test.cpp  
tom@TomVM:~/test$ node a.out.js  
a = 0H  
b = 201H  
c = 6050403H
```

(When I tried it without `#pragma pack()`, the structure alignment was on 16 bit boundaries by default, just as you pointed out.)

This was using Emscripten producing WebAssembly+JS output files.

Well, I guess it is still a long way to a U++ based GUI app running on a browser as WebAssembly with WebGL graphics backend.

Best regards,

Tom
