Play With WebAssembly

Nobuhiro Kasai

目次

1. 自己紹介
2. アセンブリ言語とは
3. WebAssemblyとは
4. 実演
5. まとめ
6. 質疑応答など

目次

1. **自己紹介**
2. アセンブリ言語とは
3. WebAssemblyとは
4. 実演
5. まとめ
6. 質疑応答など

自己紹介

- Nobuhiro Kasai
- Twitter : @sh4869sh

自己紹介

なんかC言語の教育資料とかも作ったりしました

- http://sh4869.net/CLangCourse/

いい感じに使ってください

目次

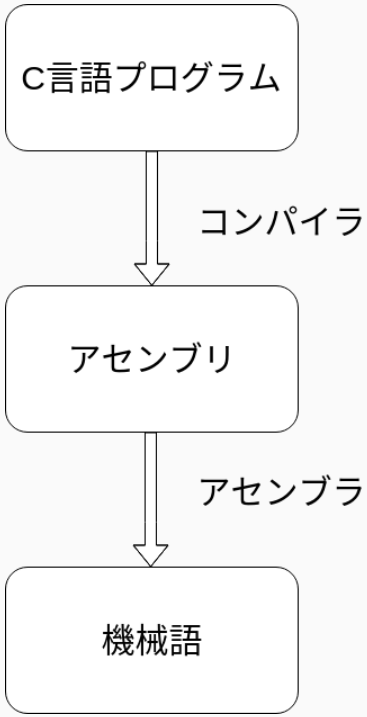
1. 自己紹介
2. **アセンブリ言語とは**
3. WebAssemblyとは
4. 実演
5. まとめ
6. 質疑応答など

アセンブリ言語とは

- CPUへの命令を記号表記で表現する言語
- アセンブラによって機械語に翻訳される

アセンブリ言語とは

- 普段C言語を書くと、コンパイラはまずC言語をアセンブリ言語に変換してくれる
- 右のような流れで実行ファイルになる



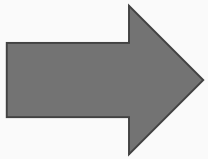
アセンブリ言語

- CPUの種類によってアセンブリ言語の種類は違う
  - MIPS
  - x86
  - etc……

アセンブリ言語の例

実際の例

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



```
  .file "test.c"
  .text
  .section  .rodata
.LC0:
  .string "Hello World"
  .text
  .globl  main
  .type main, @function
main:
.LFB0:
  .cfi_startproc
  pushq %rbp
  .cfi_def_cfa_offset 16
  .cfi_offset 6, -16
  movq  %rsp, %rbp
  .cfi_def_cfa_register 6
  leaq  .LC0(%rip), %rdi
  call  puts@PLT
  movl  $0, %eax
  popq  %rbp
  .cfi_def_cfa 7, 8
  ret
  .cfi_endproc
.LFE0:
  .size main, .-main
  .ident  "GCC: (GNU) 7.3.0"
  .section  .note.GNU-stack,"",@progbits
```

アセンブリ言語

- 正直よくわからないと思う
- CPUの命令を書いてプログラムをするから大変とだけわかっておけばいいと思います
- アセンブリ言語の概要を喋りだしたら講義になってしまう

アセンブリ勉強の参考

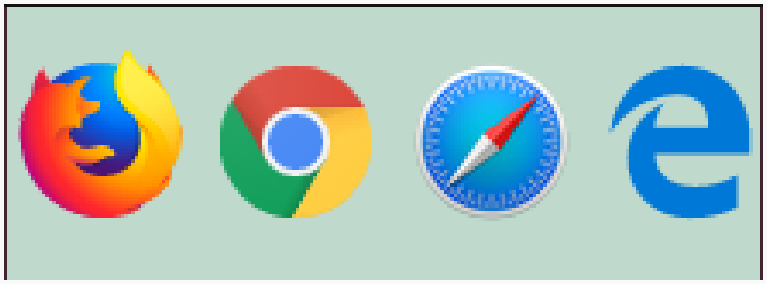
- ARMアセンブリでLチカ2
  http://idken.net/posts/2017-12-11-arm_asm
- コンピューターの構成と設計
  - 通称パタヘネ

目次

1. 自己紹介
2. アセンブリ言語とは
3. **WebAssemblyとは**
4. 実演
5. まとめ
6. 質疑応答など

Webブラウザ

- Webページを見るためのソフト
  - Chromeとか
  - Firefoxとか
  - Safariとか
  - Edgeとか
  - IE？知らない子ですね……



JavaScript

Webブラウザで動くプログラミング言語

- ブラウザで動くアプリケーションはこれのおかげで動いている
  - Twitterとか
- ブラウザ上にJavaScriptを実行するためのエンジンが搭載されている

ここでいい感じに実演をする

Web API

ブラウザの機能をコントロールするためのAPI

- APIというのは
  アプリケーションプロトコルインターフェースの略
- そのプログラムが持つ機能を使える関数のようなもの
- STM32ならファームウェアの関数がAPI

Web API

- Audio関連の機能
- DOMの機能
  - DOMはDocument Object Modelの略称で、ブラウザ上にある要素のことをさす
  - よくわからないと思うので実演します

JavaScriptからこれを叩くことができる

JavaScriptの問題

- 書きづらい
  - 癖のある言語（オブラートに包んだ物言い）なので書くのがつらい
  - 小さい規模ならよかったけど、プロジェクトが大きくなってると管理が大変
  - 型がない（静的にチェックされない）のがつらい

JavaScriptの問題

- これを解決するためにいろいろ生まれた
  - JavaScriptに変換されるための言語 (AltJS)
    - 例：TypeScript,CoffeeScript,Dart
  - JavaScript自体の仕様のアップデート
  - JavaScriptプロジェクトをビルドするための環境
    - Webpack,babel等

JavaScriptの問題

最近はだいぶ書きやすくなったが、問題はまだある

- そもそも動的にパースされるので、プログラムが複雑になるとパース自体に時間がかかってしまうのでプログラムが遅くなってしまう
- 大きな規模になると読み込むのに時間がかかる
- PCなら大したことなくても、モバイルだとしんどい
- 3Dエンジンとかやるとパフォーマンスが必要になる
- つらい……

WebAssembly

このようなパフォーマンスの問題を解決するために生まれたのが**WebAssembly**

- ブラウザ上で動くアセンブリに**似た言語**
  - 正確にはアセンブリではないが（GCのサポートとかある）、細かい話は省略
- コンパイルして生成されることを前提としている

WebAssembly

- C/C++やRustといった言語からコンパイルすることができる
  - C言語が高級言語かどうかはおいておく（炎上したくないので）
  - その他の言語もサポートされるかも

脱線:Rustについて

Rustというプログラミング言語の紹介

- Cとかを置き換える目的で作られた、静的型付け言語
- ガベージコレクションを持たない
  - ガベージコレクションというのは開放したメモリをよしなにやってくれるやつ
  - ガベージコレクションがあると楽だが、参照セマンティクスになるのとバイナリサイズがでかくなるのがつらい
  - 近代的な言語の多くがガベージコレクションを持つ

脱線:Rustについて

- メモリ管理をコンパイラがやってくれるので
  Cのような速度で安全なプログラムが書ける
- ちょっとむずかしい概念もあるが面白い言語なので
  おすすめ
- FirefoxのJavaScriptエンジンServoはRustで書かれてる

WebAssemblyのメリット

ほとんどバイナリなのでファイルサイズが小さい（はず）

- 今のJavaScriptは他の言語から変換に変換をかますのでやたらファイルサイズが大きくなることがある
- Safariだとキャッシュがいつ削除されるかわからなくてつらい
- 海外だとまだ3G回線とか多いので致命的
- ただまだコンパイルサイズを小さくする段階ではないので今はそれなりにデカイ
  - 600kbぐらい

WebAssemblyのメリット

- 命令列がそのまま記述されているのでコンパイルがいらない
  - 要はパースの時間がなくなるので早い
- どのブラウザでも動くように作られている
  - 各種ブラウザベンダが頑張ってる
  - OSが違っても高速に動くように設計されている

WebAssemblyのメリット

- バイナリだけど読める
  - Readableなフォーマットが策定されている
    （まだ策定中だが）
  - 普通のバイナリと違ってある程度人がデバックできるようになっている

WebAssemblyのメリット

- 各種ブラウザがちゃんと実装してる
  - 新しい機能は一つのブラウザしか実装してない場合が多い
    - 政治的な理由が多い、社会はつらい
  - WebAssemblyは実装状況がよい
    - Chrome、Edge、Safari、Firefox、Operaで動く
      - だがIEテメーは別だ
  - 割とうまく行く技術なんじゃないかと思う
  - 五年後には普通に使えてそう

WebAssemblyはどのように使うのか

- 実はまだWebAPIは叩けない
  - 将来的には使えるようになる設計がある
    - いつ実装されるかはわからない
    - セキュリティの問題・元言語側の対応があるのでそれなりに時間はかかりそう。

WebAssemblyは どのように使うのか

- WebAssemblyのプログラム自体をそのまま実行できるわけじゃない
  - WebAssemblyからできるのは関数などのexport
  - それをJavaScript側から読みこんで実行する形
  - JavaScript側にWebAssemblyにまつわるモジュールが用意されている

WebAssemblyは どのように使うのか

- exportできる関数に縛りがある
  - 関数の戻り地・引数に条件がある
  - たとえば配列を引数に取ることはできない
  - 共通で使えるメモリテーブルにデータを保存する必要がある
- 時間のかかる処理だけWebAssemblyで補うイメージ
  - C/C++で書かれた資産（コード）がうまく使えそうなら嬉しい
  - すべてWebAssemblyで書けるのは流石にまだ先

WebAssemblyをはじめる

現状3つの方法がある

1. C/C++からWebAssemblyに変換する
2. RustからWebAssemblyに変換する
3. WebAssemblyをそのまま書く

WebAssemblyを始める：C/C++編

- C/C++ではEmscriptenというC言語をJavaScriptに変えるコンパイラのオプションとして、WebAssemblyを吐き出すことができる
- これを使うと比較的簡単に吐き出せる

WebAssemblyを始める：C/C++編

1. Emscriptenをインストール
   https://developer.mozilla.org/ja/docs/WebAssembly/C_to_wasm に詳しく乗ってます
2. 簡単なC言語のプログラムを書く
3. Emscriptenでコンパイル
   a. emcc main.c -s WASM=1 -o hello.html
4. hello.htmlを開く

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

WebAssemblyを始める：Rust編

1. Rustの環境をインストールする
2. Rustのnightlyをインストールする
3. プログラムを書く
4. wasmにコンパイルする
5. JavaScriptプログラムを書く

WebAssemblyを始める：Rust編

- 手順自体はシンプルなのだけど、環境の整備が多少大変
- MacとLinux用にならまとめてあるブログがあります

https://sbfl.net/blog/2017/11/27/rust-webassembly-standalone/

- Windows版は暇だったら書きます
- C/C++と違ってシンプルで使いやすいので、プログラミングに慣れている人はこちらの方がおすすめ

WebAssemblyを始める：直書き

- WebAssemblyにはテキスト形式で書けるフォーマットが用意されている
  - watという
- これは可読性があるので、普通に書ける
  - S式で書く
  - Lispみたいに見える
  - ちょっとプログラムの難易度は高いかも？
- wat2wasmというツールを使うとwasmに変換できる

目次

1. 自己紹介
2. アセンブリ言語とは
3. WebAssemblyとは
4. **実演**
5. まとめ
6. 質疑応答など

実演

- マウス迷路のプログラムをWebAssemblyで
  解いてみよう
- https://github.com/sh4869/MouseSimulatorOnWeb

目次

1. 自己紹介
2. アセンブリ言語とは
3. WebAssemblyとは
4. 実演
5. **まとめ**
6. 質疑応答など

まとめ

- WebAssemblyという技術がある
  - Webブラウザ上で動くアセンブリのような言語
  - 早い、軽い、楽しい
  - CやC++で書ける

まとめ

- Web技術は面白い
  - 進化のペースが早い
  - みんなが「俺の考えた最強の思想」をぶつけ合ってる
    - 技術はこうあるべきなんじゃないかなと思う
      - 政治とかあるので大変なんですけど……
  - 興味があったらWeb触ってみてほしい

目次

1. 自己紹介
2. アセンブリ言語とは
3. WebAssemblyとは
4. 実演
5. まとめ
6. 質疑応答など

質疑応答

参考文献

- WebAssembly | MDN
  https://developer.mozilla.org/ja/docs/WebAssembly
- WebAssembly http://webassembly.org/
- Rust単体でWebAssemblyをコンパイルする（Emscripten無し）
  https://sbfl.net/blog/2017/11/27/rust-webassembly-standalone/