アセンブラ漢文

坂井弘亮
(KOZOSプロジェクト)
TwitterID:kozossakai

早速ですが

OSCのLTから生まれた書籍

ハロー

"Hello, World"

あなたが最初に書いた
プログラムにはまだ
「知らないこと」がたくさんある。

printf() 関数の先には何があるんだろう？
main() 関数が呼び出される前には何があるんだろう？
main() 関数からリターンする先はどこだ？

秀和システム

どんな書籍か

C言語の入門書で最初に書く単なる文字列出力プログラム (いわゆる「ハローワールド」)を徹底的に解析するという本

書籍中で
やっていること

ハロー・ワールドをデバッガで動的解析してシステムコール呼び出し部分まで追い，さらにLinuxカーネルのシステムコール処理を見て，次に標準Cライブラリである glibcのソースコード見て確認してついでにglibcを自前でビルドしてデバッグ情報 埋め込んでシンボルベースで解析し，さらに実行ファイルのELFを解析し，x86以外のアーキ(ARMとMIPS)やLinux以外のOS環境(FreeBSD)ではどうなるのか調べ，最

要約すると

ハロー・ワールドを
本気解析
する，という本
(自分的にはツボで
ちょう面白い)

そんな
ハロー "Hello, World"
ですが

最も苦労した点

書籍タイトル

当初に編集さんに言われたこと

「ハードな感じでいきましょう」
(たぶん「硬派な感じ」と言っている)

ハード?

ハードボイルド
ということか？

だったら
探偵だろう

ということで
最初に出した
タイトル案

「ハロー・ワールド探偵」(実話)

これはカッコいいぞ!

調子に乗る

次に出した
タイトル案

ハロー・ワールド
刑事(デカ)
(実話)

当時のメールを
読み返してみた

(全部実際に出した案)

ハロー・ワールド捜査官 ハロー・ワールド捜査団 ハロー・ワールド捜査班 ハロー・ワールド捜査委員会 ハロー・ワールド探偵団 ハロー・ワールド探険団 ハロー・ワールド探険隊 ハロー・ワールド非常線 ハロー・ワールド登山隊 ハロー・ワールド潜入捜査 ハロー・ワールド刑事(デカ) ハロー・ワールド危険地帯 ハロー・ワールドへの挑戦 ハロー・ワールド・ハンター 怪盗ハロー・ワールド 非情のハロー・ワールド 大いなるハロー・ワールド 愛しきハロー・ワー

まだまだあります

標的はハロー・ワールド ハロー・ワールドの街角 ハロー・ワールドで一杯 今夜はハロー・ワールドで一杯 蘇えるハロー・ワールド 探偵はハロー・ワールドを読む 発酵ハロー・ワールド ハロー・ワールドを醸す 野性のハロー・ワールド ハロー・ワールドは旨い ハロー・ワールド青春時代 無限のハロー・ワールド ハロー・ワールドの銀河 ハロー・ワールド千夜一夜 ハロー・ワールド物語 ハロー・ワールドでいこう ハロー・ワールドの要塞 必殺ハロー・ワールド あ

まだまだまだまだあります
(図に乗るタイプ)

じゃじゃ馬ハロー・ワールド ハロー・ワールド倶楽部 ハロー・ワールド盗難事件 ハロー・ワールド迷宮事件 ハロー・ワールド深海探査 ハロー・ワールド深海探索 ハロー・ワールド深海探査船 ハロー・ワールド・ダイバー ダイビング・ハロー・ワールド ダイビング・イン・ハロー・ワールド ハロー・ワールドッチャブル ハロー・ワールド大作戦 ハロー・ワールド・ホームズ ハロー・ワールド・シャーロックホームズ ホーム

自分的にヒットなタイトル案

「ハロー・ワールド盗難事件」

なんじゃそりゃ！読みてえ！
(探偵小説好き．とくに海外ものと金田一(少年じゃないほう)専門)

「ハロー・ワールドッチャブル」

(おそらくスーパーハードボイルド映画「アンタッチャブル」の影響．でもどう見てもハードボイルドでなく色モノにしか思えないが，当時は やたら気にいって必死に推していた．却下してくれた編集さんありがとうございます)

「ハロー・ワールドの銀河」

(スケールでかすぎだろう)

「じゃじゃ馬ハロー・ワールド」

(もはや何なのか全然わからないが,とりあえず数ページ読んでみたい)

そんな芸風で
やっていますが

本日は
アセンブラの
話です

まず軽い話から
入ります

OSCのLTから生まれた崇高な芸術活動

アセンブラ短歌

アセンブラ短歌とは何か

アセンブラ短歌とは

五･七･五･七･七の三十一バイト(みそひとバイト)から成る機械語コードでプログラムを書いてみるという近未来の文化的趣味であり, 近年, 国内のハッカー間で密かなブームが起きています.

こんな感じです

タイトル:「夏休み」(詠み人:坂井弘亮)

6a 00 58 50 40
68 79 61 6d 61 50 40
6a 08 5a 5b 40
68 57 61 6b 61 54 40
59 cd 80 58 58 58 c3

アセンブラ短歌の問題点

固定長命令や
偶数長命令の
CPUでは原理的
にできない

アセンブラ短歌をやることを想定して設計された(と思われる)CPU一覧 (Tankable Architectures)

x86(Intel) M32C(三菱)
MN10300(松下)
RL78(ルネサス) RX(ルネサス)
Xtensa(テンシリカ)

x86は短歌対応されているので可能だが,ARMは短歌非対応なので不可能

つまりスマフォでは短歌は詠めない

(スマフォ嫌いの人がPCを使う大きな理由のひとつと思われる)

しかし待てよ

固定長命令

↓

1句の長さが固定ならば書きやすい

漢詩なら書けるのでは

春暁(孟浩然)

春眠不覺曉
處處聞啼鳥
夜來風雨聲
花落知多少

(五言絶句)

五言絶句 ... 1句が5字で4句
五言律詩 ... 1句が5字で8句
七言絶句 ... 1句が7字で4句
七言律詩 ... 1句が7字で8句

アセンブラ漢詩

固定の句長の機械語コードでプログラムを書いてみるという 近未来の文化的趣味であり, 近年, 国内のハッカー間で密かなブームが起きると思われます.

やってみた

手始めに練習としてx86で

x86
五言律詩

アセンブラ

(FreeBSD/Linux両方で動作するというおまけつき)

```
push %ebx
push $4
push $10
pop %edx
pop %eax
xor %ebx, %ebx
inc %ebx
push $0xfcda
push $0xb4ebd4c9
push $0xb2ccd5bd
mov %esp, %ecx
push %edx
push %ecx
push %ebx
push $0
int $0x80
pop %eax
add $24, %esp
pop %ebx
ret
```

83	6A	89	68	68	68	5A	53
C4	00	E1	BD	C9	DA	58	6A
18	CD	52	D5	D4	FC	31	04
5B	80	51	CC	EB	00	DB	6A
C3	58	53	B2	B4	00	43	0A

83	6A	89	68	68	68	5A	53
C4	00	E1	BD	C9	DA	58	6A
18	CD	52	D5	D4	FC	31	04
5B	80	51	CC	EB	00	DB	6A
C3	58	53	B2	B4	00	43	0A

(注意:右上から縦読みです)

実行結果

春眠不覺曉

読みやすくしたい

x86はリトルエンディアン

(実際の値)
0x12345678

(メモリ上のデータ配置)

78	56	34	12

リトルエンディアンのため
値がひっくり返っている位置

83	6A	89	68	68	68	5A	53
C4	00	E1	BD	C9	DA	58	6A
18	CD	52	D5	D4	FC	31	04
5B	80	51	CC	EB	00	DB	6A
C3	58	53	B2	B4	00	43	0A

値がひっくり返っているので

レ点を打つ

83	6A	89	68	68	68	5A	53
C4	00	E1	BD	C9	DA	58	6A
18	CD	52	D5	D4	FC	31	04
5B	80	51	CC	EB	00	DB	6A
C3	58	53	B2	B4	00	43	0A

83	6A	89	68	68	68	5A	53
C4	00	E1	BD	C9	DA	58	6A
18	CD	52	D5	D4	FC	31	04
5B	80	51	CC	EB	00	DB	6A
C3	58	53	B2	B4	00	43	0A

読みやすくなった

次

MIPSでやってみる

MIPSは4バイト固定長命令なので，普通に書くだけで漢詩になる(ただし五言とか七言とかはムリ)

MIPS
八言律詩

アセンブラ

```
lui     $v0,0xbdd5
ori     $v0,$v0,0xccb2
sw      $v0,0($sp)
lui     $v0,0xc9d4
ori     $v0,$v0,0xebb4
sw      $v0,4($sp)
lui     $v0,0xdafc
sw      $v0,8($sp)
li      $a0,1
move    $a1,$sp
li      $a2,10
jal     __write
li      $a0,0
jal     __exit
li      $v0,0
jr      $ra
```

03 0C 0C 24 3C 34 AF 3C
E0 00 00 04 02 42 A2 02
00 00 00 00 DA EB 00 BD
08 07 0F 01 FC B4 00 D5
24 24 24 03 AF AF 3C 34
02 04 06 A0 A2 A2 02 42
00 00 00 28 00 00 C9 CC
00 00 0A 21 08 04 D4 B2

実行結果

春眠不覺曉

読みやすくしたい

MIPSは遅延スロットを持つ

遅延スロット
ジャンプ命令の直後の命令も
実行される
↓
読むときはジャンプ命令と直後
の命令をひっくり返す感じ

ひっくり返す？

一二点を打たなければ

03 0C 0C 24 3C 34 AF 3C
E0 00 00 04 02 42 A2 02
00 00 00 00 DA EB 00 BD
08 07 0F 01 FC B4 00 D5
24 24 24 03 AF AF 3C 34
02 04 06 A0 A2 A2 02 42
00 00 00 28 00 00 C9 CC
00 00 0A 21 08 04 D4 B2

03 0C 0C 24 3C 34 AF 3C
E0 00 00 04 02 42 A2 02
00 00 00 00 DA EB 00 BD
08 07 0F 01 FC B4 00 D5
24 24 24 03 AF AF 3C 34
02 04 06 A0 A2 A2 02 42
00 00 00 28 00 00 C9 CC
00 00 0A 21 08 04 D4 B2

まとめ

MIPSやARMなどの固定長命令のアーキテクチャは 短歌は不可能な場合が多い(Untankable Architectures)が，漢詩ならば可能(Kanshable Architectures)である

どうもありがとう
ございました