

続・
ハロー・ワールド
入門

坂井弘亮

(KOZOSプロジェクト)

TwitterID:kozossakai



每年恒例

バイナリカレンダー
あります

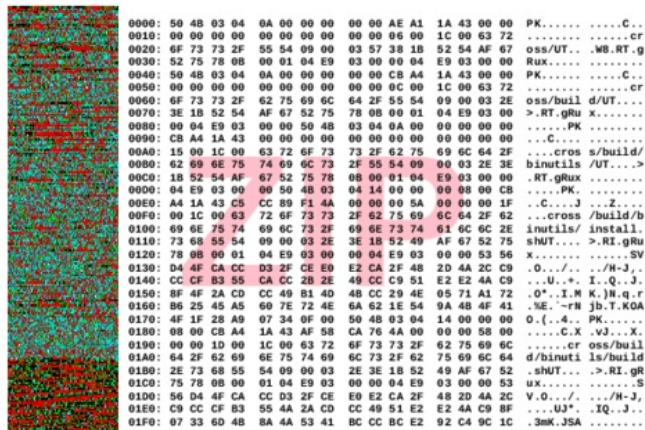
バイナリカレンダー
とは何か

(普通のカレンダー)
風景や人物がテーマ

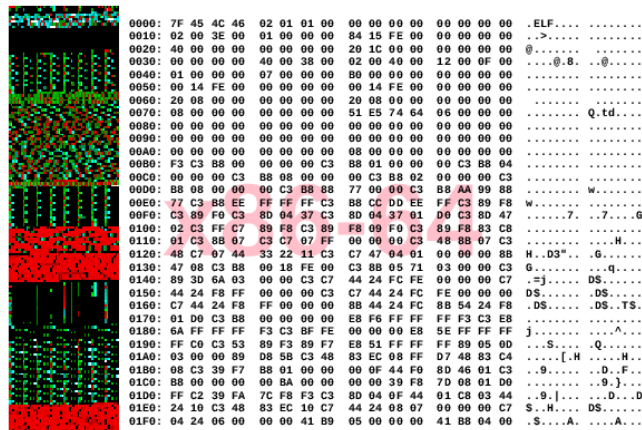
(バイナリカレンダー)
バイナリデータがテーマ

こんな感じです

(バイナリカレンダー-2014)



(アーキテクチャカレンダー-2015)



January 0x7DE

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		01	02	03	04	
		00001	00010	00011	00100	
05	06	07	08	09	0A	0B
00101	00110	00111	01000	01001	01010	01011
0C	0D	0E	0F	10	11	12
01100	01101	01110	01111	10000	10001	10010
13	14	15	16	17	18	19
10011	10100	10101	10110	10111	11000	11001
1A	1B	1C	1D	1E	1F	
11010	11011	11100	11101	11110	11111	

June 0x7DF

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	01	02	03	04	05	06
	098 / 0D5	099 / 0D4	09A / 0D3	09B / 0D2	09C / 0D1	09D / 0D0
07	08	09	0A	0B	0C	0D
09E / 0CF	09F / 0CE	0A0 / 0CD	0A1 / 0CC	0A2 / 0CB	0A3 / 0CA	0A4 / 0C9
0E	0F	10	11	12	13	14
0A5 / 0C8	0A6 / 0C7	0A7 / 0C6	0A8 / 0C5	0A9 / 0C4	0AA / 0C3	0AB / 0C2
15	16	17	18	19	1A	1B
0AC / 0C1	0AD / 0C0	0AE / 0BF	0AF / 0BE	0B0 / 0BD	0B1 / 0BC	0B2 / 0BB
1C	1D	1E				
0B3 / 0BA	0B4 / 0B9	0B5 / 0B8				

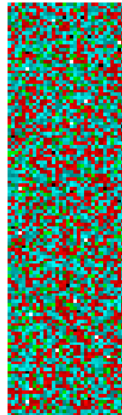
そして今年

乱数レンダラー

乱数カレンダー とは何か

乱数データをテーマとした
バイナリカレンダーの
一種です

こんな感じです



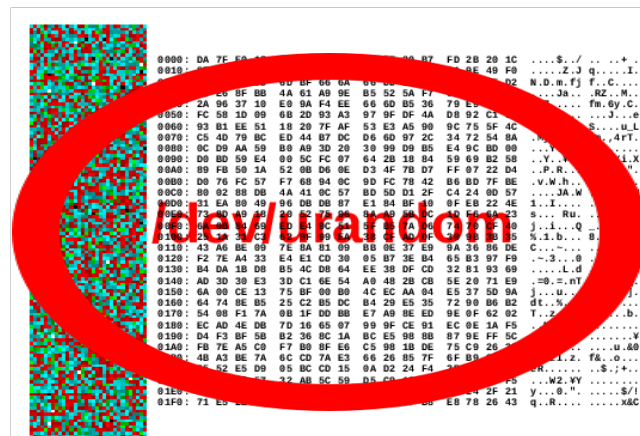
```

0000: DA 7F E9 13 24 B6 9E 2F 9C EE 29 07 FD 2B 2B 1C ....$/.....+
0001: 8B 8A E9 AC 5A AA 4A 71 B6 D4 93 11 9E 49 F9 .....2.Jq....I.
0002: 4E E7 44 D0 6D BF 66 6A 66 D0 D7 43 D3 BA CF D2 W.D.B.f} f.C....
0003: F3 E6 BF BB 4A 61 A9 9E 85 52 5A F7 0A 4D A4 99 ....Ja...RZ..M..
0004: 2A 96 37 19 E9 9A F4 EE 66 6D B5 36 78 E9 43 E9 *.7....fm.gy.C.
0005: FC 59 1D 09 6B D0 93 A3 97 9F DF 4A 08 92 C1 65 .X.K.....J...e
0006: 93 B1 EE 51 18 29 7F AF 53 E3 A5 90 9C 75 5F 4C ..Q...S...u.L
0007: C5 4D 79 BC ED 44 B7 DC 06 6D 97 2C 34 72 54 8A .My..D...m..4rT.
0008: 9C D9 AA 59 09 A5 3D 29 39 09 D5 54 9C 8D 89 ...Y...F 6.....
0009: D9 BD 59 E4 09 5C FC 07 64 2B 18 84 59 69 B2 58 ..Y.V...d+.YiX.
000A: 89 F8 59 1A 52 08 D6 0E 03 4F 7B D7 FF 07 22 D4 ..P.R...0[...".
000B: D9 76 FC 57 F7 6B 94 9C 9D FC 78 42 86 B0 7F 9E .v.W.h...x8....
000C: 89 02 8B D8 4A 41 9C 57 BD 5D D1 2F C4 24 0D 57 ....JA.W.]./.$W
000D: 31 EA 88 49 86 D8 D8 87 E1 84 BF 19 0F EB 22 4E 1..I....."N
000E: 75 09 A9 89 86 52 78 09 DA 45 86 C0 56 6A 23 S...Ru...[...jP
000F: 4B 09 08 00 04 C4 43 3F 06 FA 00 71 0F 00 j..i...Q...ztp-0
0100: 26 1A 31 C2 42 BF 99 6A 39 0E 65 9F 43 88 80 85 %1.b...8...0;5
0101: 43 A6 8E 09 7E BA 81 09 88 8E 37 E9 8A 36 86 DE G.....7f..6..
0102: F2 7E A4 33 E4 E1 C0 38 05 87 3E B4 65 B3 97 F9 ...3...8...>-d..
0103: 84 DA 1B D8 85 4C D8 64 EE 38 DF C0 32 81 93 09 ....L.d..8..2..i
0104: AD 3D 38 E3 3D C1 6E 54 A8 48 2B C8 5E 29 71 E9 ..8.=.nT..H+.A.q.
0105: 6A 09 CE 13 75 BF 08 08 4C EC AA 04 E5 37 5D 9A j.....L...[7].
0106: 64 74 8E B5 25 C2 B5 DC 84 29 E5 35 72 90 B6 B2 dt..%....).5f...
0107: 54 88 F1 7A 08 1F D0 DB E7 A9 8E ED 9E 0F 62 82 T..2.....b.
0108: EC AD 4E DB 7D 16 65 97 09 9F CE 91 EC 0E 1A F5 ..W.]..e.....
0109: D4 F3 BF 5B 82 36 8C 1A BC E5 98 8B 87 9E FF 5C ...[.6.....vY
010A: F8 7E A5 C9 F7 B9 BF E6 C5 98 1B DE 75 C9 26 39 .....u..u.80
010B: 4B A3 8E 7A 6C 0C 7A 53 66 26 95 7F 6F 8B 0E 07 K...1..2..fA..0...
010C: 65 52 E5 D9 05 BC D0 15 0A D2 24 F4 3B 2B A1 9F eR.....$.;+..
010D: 08 EA D5 57 32 AB 5C 59 D5 C9 13 8A 9F F2 00 F5 ...W2.VY.....
010E: 79 A7 07 17 39 09 22 E3 C2 83 F5 1E B2 24 2F 21 y...8..."$/I.
010F: 71 E5 2E 52 81 F9 D6 AC 02 D3 97 B9 E8 76 26 43 q...R.....x6C

```

January 0x7E0

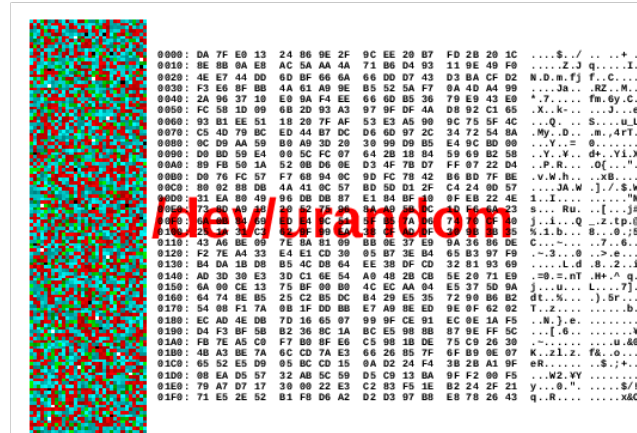
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					01 00001	02 00010
03 00011	04 00100	05 00101	06 00110	07 00111	08 01000	09 01001
0A 01010	0B 01011	0C 01100	0D 01101	0E 01110	0F 01111	10 10000
11 10001	12 10010	13 10011	14 10100	15 10101	16 10110	17 10111
18 11000	19 11001	1A 11010	1B 11011	1C 11100	1D 11101	1E 11110
1F 11111						



乱数データの
ダンプ

January 0x7E0

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					01 00001	02 00010
03 00011	04 00100	05 00101	06 00110	07 00111	08 01000	09 01001
0A 01010	0B 01011	0C 01100	0D 01101	0E 01110	0F 01111	10 10000
11 10001	12 10010	13 10011	14 10100	15 10101	16 10110	17 10111
18 11000	19 11001	1A 11010	1B 11011	1C 11100	1D 11101	1E 11110
1F 11111						



January 0x7E0

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						02
					00001	010
0111	04	05	06	07	08	01000
0A	0B	0C	0D	0E	0F	10
01010	01011	01100	01101	01110	01111	10000
11	12	13	14	15	16	17
0001	10010	10011	10100	10101	10110	101
	19	1A	1B	1C	1D	110
1E	11001	11010	11011	11100	11101	110
1F						
11111						

日付は16進数と
2進数を併記
(さすがにここは
乱数ではない)

乱数一覧

1月 /dev/urandom

7月 bashの\$RANDOM

2月 rand()

8月 Pythonの
random.randint()

3月 ガウス分布

9月 random()

4月 旧rand()

10月 Numpyの
random.randint()

5月 M系列

11月 Awkのrand()

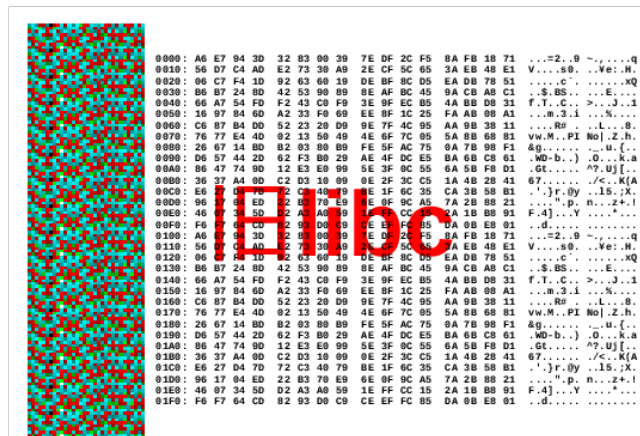
6月 Perlのrand()

12月 openssl rand

特徴的な月

4月

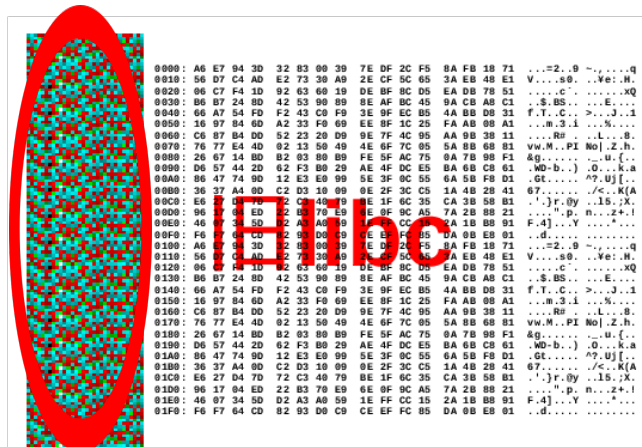
(旧libcの rand()関数)



April 0x7E0						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					01 00001	02 00010
03 00011	04 00100	05 00101	06 00110	07 00111	08 01000	09 01001
0A 01010	0B 01011	0C 01100	0D 01101	0E 01110	0F 01111	10 10000
11 10001	12 10010	13 10011	14 10100	15 10101	16 10110	17 10111
18 11000	19 11001	1A 11010	1B 11011	1C 11100	1D 11101	1E 11110

4月

(旧libcの
rand()関数)



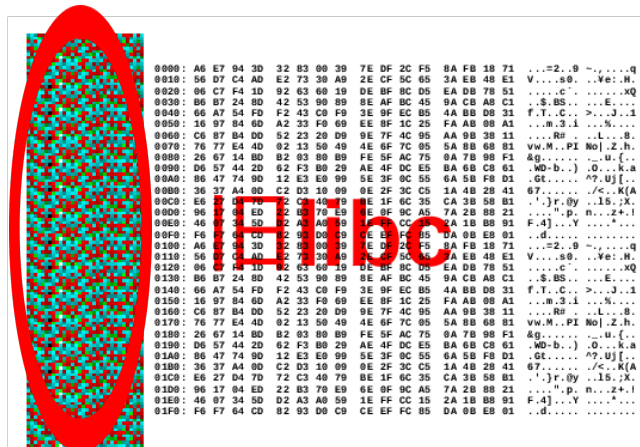
パターンが
規則的

April 0x7E0

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					01 00001	02 00010
03 00011	04 00100	05 00101	06 00110	07 00111	08 01000	09 01001
0A 01010	0B 01011	0C 01100	0D 01101	0E 01110	0F 01111	10 10000
11 10001	12 10010	13 10011	14 10100	15 10101	16 10110	17 10111
18 11000	19 11001	1A 11010	1B 11011	1C 11100	1D 11101	1E 11110

4月

(旧libcの
rand()関数)



パターンが
規則的

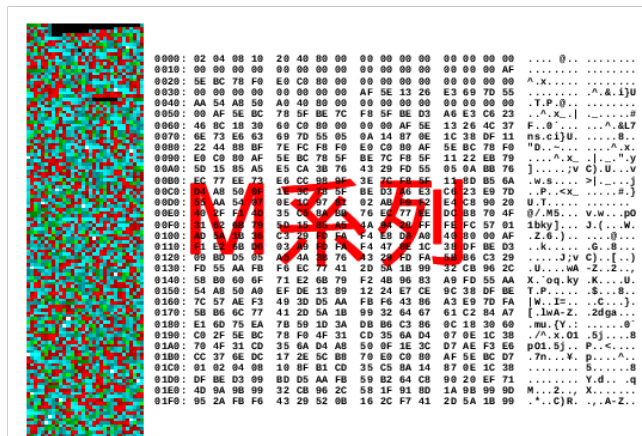
April 0x7E0

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					01	02
					00001	00010
03	04	05	06	07	08	09
00011	00100	00101	00110	00111	01000	01001
0A	0B	0C	0D	0E	0F	10
01010	01011	01100	01101	01110	01111	10000
11	12	13	14	15	16	17
10001	10011	10101	10110	10111	11000	11001
18	19	1A	1B	1C	1D	1E
11000	11001	11010	11011	11100	11101	11110

$rv = rv * 1103515245 + 12345;$

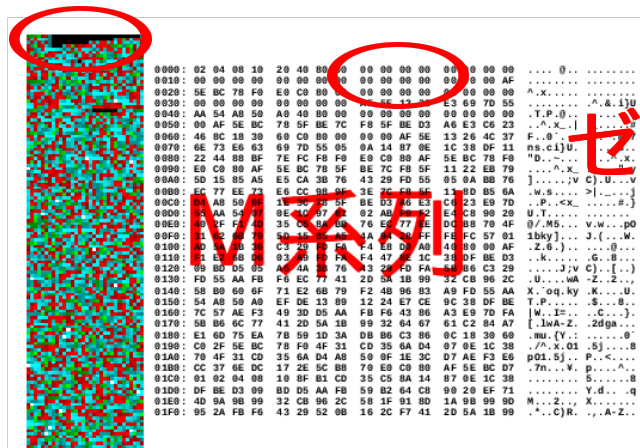
他にも

5月 (M系列)



May 0x7E0						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
01 00001	02 00010	03 00011	04 00100	05 00101	06 00110	07 00111
08 01000	09 01001	0A 01010	0B 01011	0C 01100	0D 01101	0E 01110
0F 01111	10 10000	11 10001	12 10010	13 10011	14 10100	15 10101
16 10110	17 10111	18 11000	19 11001	1A 11010	1B 11011	1C 11100
1D 11101	1E 11110	1F 11111				

5月 (M系列)

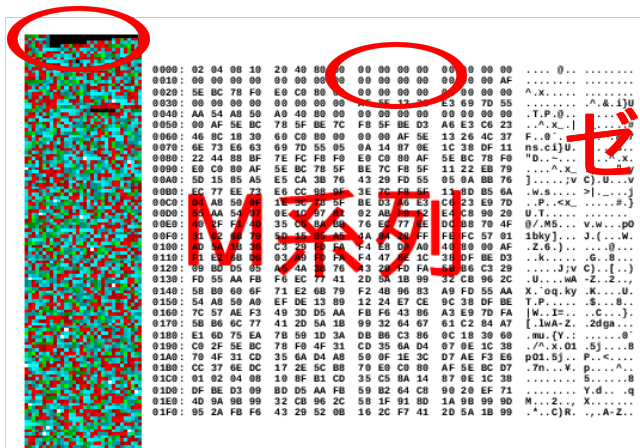


先頭に
ゼロデータの
連続

May 0x7E0

Sun	Mon	Tue	Wed	Thu	Fri	Sat
01 00001	02 00010	03 00011	04 00100	05 00101	06 00110	07 00111
08 01000	09 01001	0A 01010	0B 01011	0C 01100	0D 01101	0E 01110
0F 01111	10 10000	11 10001	12 10010	13 10011	14 10100	15 10101
16 10110	17 10111	18 11000	19 11001	1A 11010	1B 11011	1C 11100
1D 11101	1E 11110	1F 11111				

5月
(M系列)



先頭に ゼロデータの 連続

乱数の種を適切に選ばないと
乱数性が低くなる
(この例では seed = 1)

May 0x7E0						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
01 0001	02 0010	03 0011	04 0100	05 0101	06 0110	07 0111
08 01000	09 01001	10 01010	11 01011	12 01100	13 01101	14 01110
15 01111	16 10000	17 10001	18 10010	19 10011	20 10100	21 10101
22 10110	23 10111	24 11000	25 11001	26 11010	27 11011	28 11100
29 11101	30 11110	31 11111				

前置き終わり

続・
ハロー・ワールド
入門

坂井弘亮

(KOZOSプロジェクト)

TwitterID:kozossakai



「ハロー・ワールド入門」 とは何か

1年前の
OSC東京春の
LTでの発表

「ハロー・ワールド入門」

C言語の入門時に
必ずと言っていいほど書く
「ハロー・ワールド」

こんなのです

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hello, World!¥n");
```

```
    return 0;
```

```
}
```


実行結果

He l Lo, Wo r ld!

単に

「Hello, World!」
と出力するだけの
プログラム

たいていのプログラミング
入門書はこれで始まる

これを徹底的に
解析する, という話

1年前の
OSC東京春の
LTで言ったこと

マニアックな技術書を書
くのはもう卒業だ！

今年は入門書を書くぞ！

入門書と言えば
ハロー・ワールドだろう

じゃあハロー・ワールドだけに
ついて解説する本を書けば、
入門書の前に読むべき
スーパー入門書になるのでは？

そして今年

続・
ハロー・ワールド
入門

坂井弘亮

(KOZOSプロジェクト)

TwitterID:kozossakai



そもそも
きっかけ

こんなことを
言われたことが
ある

「ハロー・ワールド」について
レポートを書けという
課題が出れば、
5枚でも6枚でも書ける！
(要するにそれほど奥が深い)

これを聞いたときの
正直な感想

いやいや
無理でしょ

足りないでしょ

50枚は必要でしょ

ということで
書いてみた

「ハロー "Hello, World"」

ハ
ロ
ー

"Hello, World"

たった7行のCプログラムから解き明かす
OSと標準ライブラリのシゴトとしくみ

坂井弘亮 [著]

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

(当初の予定)

ハロー・ワールドについて
いろんな視点から説明した
150ページくらいの薄い本

(できあがったもの)

ハロー・ワールドについて
徹底的に解析した
450ページくらいの
薄くない本(またか)

注意

意

C言語によるハロー・ワールド
の入門です

ハロー・ワールドによるC言語
の入門ではありません

つまり
K&Rの前に読むべき
スーパー入門書

ハロー・ワールドの
入門書なので

この本には
リスト1しか
ありません

利用しているOSS
(たいへんありがたい)

Linuxカーネル glibc
FreeBSD Newlib
GDB GCC binutils
CentOS

内

容

(第1章)

CentOS環境でハロー・ワールドをビルドして 逆アセンブル結果や実行ファイルの解析結果を眺めてみたりする

(第2章)

printf()の動作を知るために
デバッガで動的解析して シ
ステムコールの呼出しまでを
追う

(第3章)

Linuxカーネルのソースコード
を読んで, Linuxのシステム
コール処理を探る

(第4章)

標準ライブラリのソースコードを読んでシステムコールの呼出し処理を知る. ついでにビルドして動的解析で処理を追う

(第5章)

main()関数が呼ばれる前の
スタートアップの処理と,
main()関数から戻った後の
終了処理を探る

(第6章)

標準ライブラリのソースコードを読んでprintf()の実装を知る。FreeBSDやNewlibの実装も見てみる

こんなのが延々と続く，
(自己評価で)最高に面白い本

他にも実行ファイル解析
共有ライブラリ
最適化
可変長引数などの話もあり

Linux/x86だけでなく
FreeBSDやARMなど
他実装／アーキの話題もあり
(このへんが特色)

どうもありがとう
ございました

