

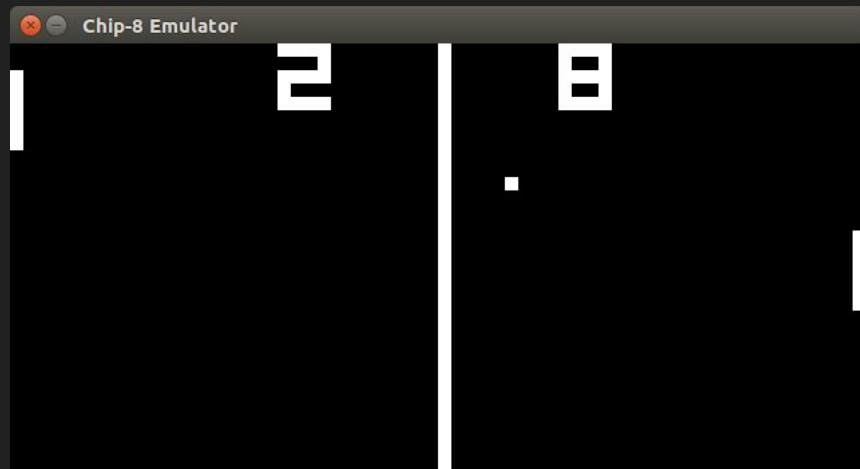
JAK NAPISAĆ WŁASNY EMULATOR

Czym jest emulator?



Jaki jest cel pisania własnego
emulatora?

CHIP-8

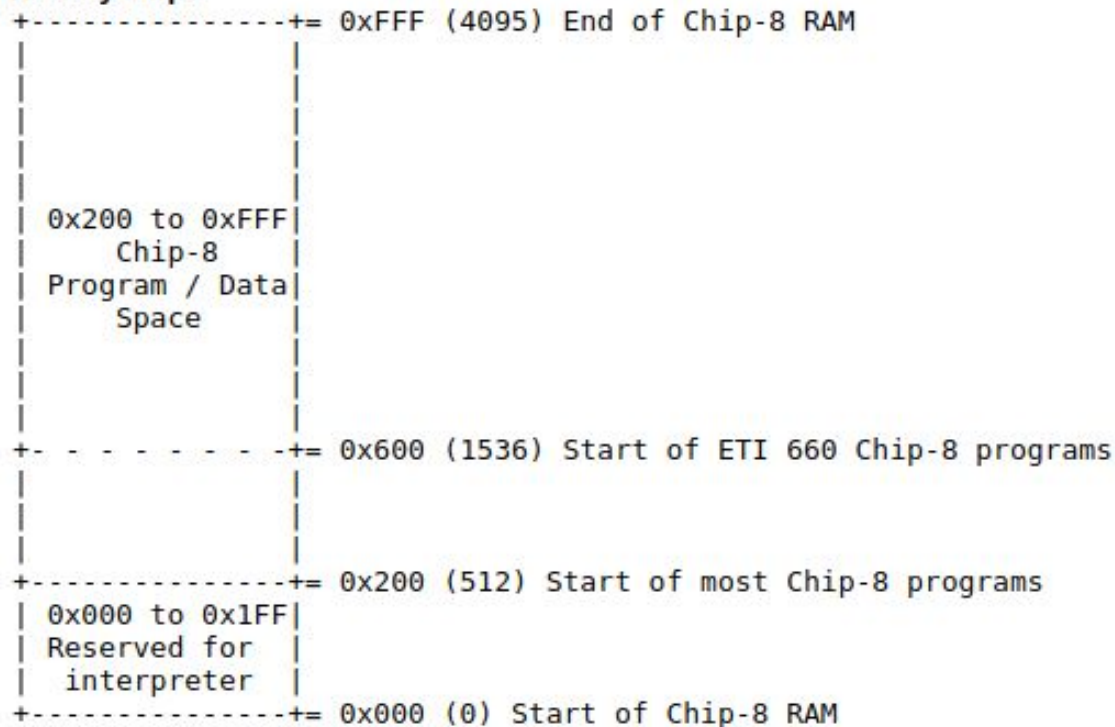


Budowa emulatora CHIP-8

1. Pamięć
2. CPU
3. Urządzenia wejścia-wyjścia

Pamięć - 4KB

Memory Map:



Zawartość pliku z programem

/home/rafzby/Pulpit/ROMs/PONG2.c8 - Bless

PONG2.c8 ✕

00000000	22	F6	6B	0C	6C	3F	6D	0C	A2	EA	DA	B6	DC	D6	6E	00	22	D4	"	k.l?m.....n."
00000012	66	03	68	02	60	60	F0	15	F0	07	30	00	12	1A	C7	17	77	08	f.h.`.....0.....w.	
00000024	69	FF	A2	F0	D6	71	A2	EA	DA	B6	DC	D6	60	01	E0	A1	7B	FE	i....q.....`...{.	
00000036	60	04	E0	A1	7B	02	60	1F	8B	02	DA	B6	60	0C	E0	A1	7D	FE	`...{.`.....`...}	
00000048	60	0D	E0	A1	7D	02	60	1F	8D	02	DC	D6	A2	F0	D6	71	86	84	`...}.`.....q..	
0000005a	87	94	60	3F	86	02	61	1F	87	12	46	00	12	78	46	3F	12	82	..`?...a...F..xF?..	
0000006c	47	1F	69	FF	47	00	69	01	D6	71	12	2A	68	02	63	01	80	70	G.i.G.i..q.*h.c..p	
0000007e	80	B5	12	8A	68	FE	63	0A	80	70	80	D5	3F	01	12	A2	61	02	...h.c..p..?..a.	
00000090	80	15	3F	01	12	BA	80	15	3F	01	12	C8	80	15	3F	01	12	C2	..?.....?.....?	
000000a2	60	20	F0	18	22	D4	8E	34	22	D4	66	3E	33	01	66	03	68	FE	` .."...4".f>3.f.h.	
000000b4	33	01	68	02	12	16	79	FF	49	FE	69	FF	12	C8	79	01	49	02	3.h...y.I.i...y.I.	
000000c6	69	01	60	04	F0	18	76	01	46	40	76	FE	12	6C	A2	F2	FE	33	i.`...v.F@v..l...3	
000000d8	F2	65	F1	29	64	14	65	00	D4	55	74	15	F2	29	D4	55	00	EE	.e.)d.e..Ut..).U..	
000000ea	80	80	80	80	80	80	80	00	00	00	00	00	6B	20	6C	00	A2	EAk l...	
000000fc	DB	C1	7C	01	3C	20	12	FC	6A	00	00	EE							..l.< ..j...	

Signed 8 bit:	34	Signed 32 bit:	586574604	Hexadecimal:	22 F6 6B 0C	✕
Unsigned 8 bit:	34	Unsigned 32 bit:	586574604	Decimal:	034 246 107 012	
Signed 16 bit:	8950	Float 32 bit:	6,679177E-18	Octal:	042 366 153 014	
Unsigned 16 bit:	8950	Float 64 bit:	2,94145332167706E-140	Binary:	00100010 11110110 01	
<input type="checkbox"/> Show little endian decoding		<input type="checkbox"/> Show unsigned as hexadecimal		ASCII Text:	"?k _{asc}	
Offset: 0x0 / 0x107				Selection: None	INS	


```

6 public class Memory {
7     private char[] memory;
8
9
10    public Memory(int size) {
11        memory = new char[size];
12    }
13
14    public char readByte(int address) throws MemoryReadException {
15        if (address < 0 || address >= memory.length) {
16            throw new MemoryReadException("Attempt to read data from wrong memory address.");
17        }
18
19        return memory[address];
20    }
21
22    public char readOpcode(int address) throws MemoryReadException {
23        return (char) ((readByte(address) << 8) | readByte(address + 1));
24    }
25
26    public void writeByte(int address, char value) throws MemoryWriteException {
27        if (address < 0 || address >= memory.length) {
28            throw new MemoryWriteException("Attempt to write date to the wrong memory address.");
29        }
30
31        memory[address] = (char) (value & 0xFF);
32    }
33
34    @Override
35    public String toString() {
36        StringBuilder builder = new StringBuilder();
37
38        for (int i = 0; i < memory.length; i++) {
39            try {
40                builder.append(Integer.toHexString(i))
41                    .append(": ")
42                    .append(Integer.toHexString(readByte(i)))
43                    .append("\n");
44            } catch (MemoryReadException e) {
45                e.printStackTrace();
46            }
47        }
48
49        return builder.toString();
50    }
51 }

```

0x - Prefix stosowany przy zapisie liczb szesnastkowych w różnych językach programowania.

MEMORY

```

-----+
0x200: 22|
0x201: F6|
0x202: 6B|
0x203: 0C|
-----+

```

readOpcode(200)

0x22 = 0010 0010

0010 0010 << 8
0010 0010 0000 0000

0xF6 = 1111 0110

0010 0010 0000 0000 | 1111 0110

```

0010 0010 0000 0000 |
                      1111 0110
-----

```

0010 0010 1111 0110

opcode = 0010 0010 1111 0110 = 0x22F6

CPU - Central Processing Unit

- 35 instrukcji.
- 16 rejestrów ogólnego przeznaczenia (1B, 0x00 - 0xFF), jeden zarezerwowany na flagę przeniesienia (ang. carry flag) o adresie 0xF.
- Rejestr I - służący do przechowywania adresu pamięci do późniejszego użytku. Nie ma możliwości jego modyfikacji.
- Licznik instrukcji.
- Stos (16 elementów).
- Delay Timer.
- Sound Timer.

Urządzenia wejścia-wyjścia

- Wyświetlacz o rozmiarach 64x32px
- Klawiatura z 16 klawiszami

Wyświetlacz

- Zwykły JPanel
- 2 kolory pikseli - czarny i biały
- 16-elementowa czcionka (0-F)
- Dla lepszego odbioru ekran został powiększony do rozmiaru 640x320px
- Wartość każdego piksela pomnożona przez 10

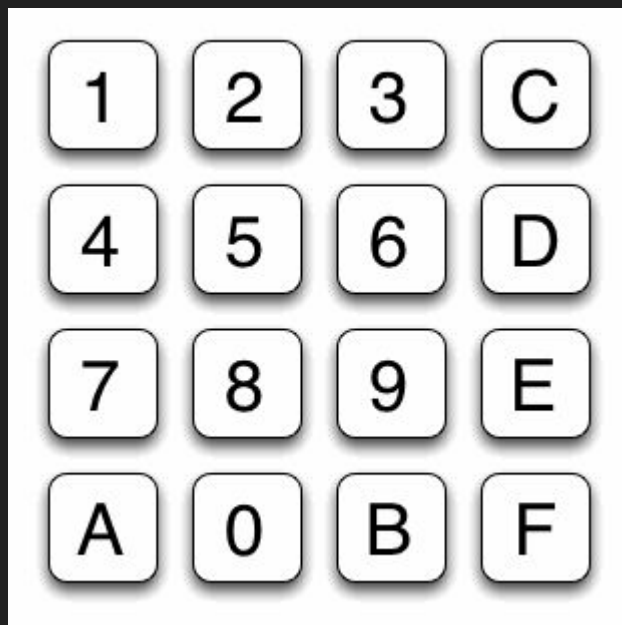
Czcionka

- 16 znaków zapisanych w pamięci RAM zaczynając od adresu 0x50

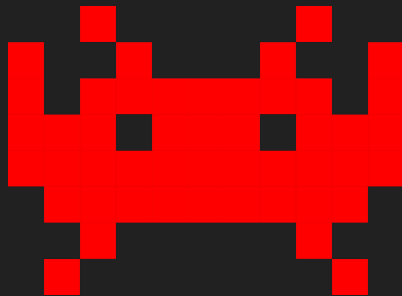
"0"	Binary	Hex	"1"	Binary	Hex
****	11110000	0xF0	*	00100000	0x20
* *	10010000	0x90	**	01100000	0x60
* *	10010000	0x90	*	00100000	0x20
* *	10010000	0x90	*	00100000	0x20
****	11110000	0xF0	***	01110000	0x70
"2"	Binary	Hex	"3"	Binary	Hex
****	11110000	0xF0	****	11110000	0xF0
*	00010000	0x10	*	00010000	0x10
****	11110000	0xF0	****	11110000	0xF0
*	10000000	0x80	*	00010000	0x10
****	11110000	0xF0	****	11110000	0xF0
"4"	Binary	Hex	"5"	Binary	Hex
* *	10010000	0x90	****	11110000	0xF0
* *	10010000	0x90	*	10000000	0x80
****	11110000	0xF0	****	11110000	0xF0
*	00010000	0x10	*	00010000	0x10
*	00010000	0x10	****	11110000	0xF0
"6"	Binary	Hex	"7"	Binary	Hex
****	11110000	0xF0	****	11110000	0xF0
*	10000000	0x80	*	00010000	0x10
****	11110000	0xF0	*	00100000	0x20
* *	10010000	0x90	*	01000000	0x40
****	11110000	0xF0	*	01000000	0x40

"8"	Binary	Hex	"9"	Binary	Hex
****	11110000	0xF0	****	11110000	0xF0
* *	10010000	0x90	* *	10010000	0x90
****	11110000	0xF0	****	11110000	0xF0
* *	10010000	0x90	*	00010000	0x10
****	11110000	0xF0	****	11110000	0xF0
"A"	Binary	Hex	"B"	Binary	Hex
****	11110000	0xF0	***	11100000	0xE0
* *	10010000	0x90	* *	10010000	0x90
****	11110000	0xF0	***	11100000	0xE0
* *	10010000	0x90	* *	10010000	0x90
* *	10010000	0x90	***	11100000	0xE0
"C"	Binary	Hex	"D"	Binary	Hex
****	11110000	0xF0	***	11100000	0xE0
*	10000000	0x80	* *	10010000	0x90
*	10000000	0x80	* *	10010000	0x90
*	10000000	0x80	* *	10010000	0x90
****	11110000	0xF0	***	11100000	0xE0
"E"	Binary	Hex	"F"	Binary	Hex
****	11110000	0xF0	****	11110000	0xF0
*	10000000	0x80	*	10000000	0x80
****	11110000	0xF0	****	11110000	0xF0
*	10000000	0x80	*	10000000	0x80
****	11110000	0xF0	*	10000000	0x80

Klawiatura



DEMO



Zasoby

<http://devernay.free.fr/hacks/chip8/C8TECH10.HTM>

<http://mattmik.com/files/chip8/mastering/chip8.html>

<https://chris.beams.io/posts/git-commit/>

<https://en.wikipedia.org/wiki/CHIP-8>

<https://github.com/rafzby/chip8-emulator>

https://helion.pl/ksiazki/zrozumiec-programowanie-gynvael-coldwind,e_1ott.htm

Kontakt

rafal.zbytniewski@gmail.com

<https://www.linkedin.com/in/rafzby/>

<https://github.com/rafzby/>

Q&A

Thanks!

