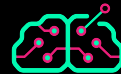


A Arte da Engenharia Reversa

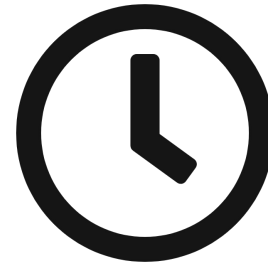
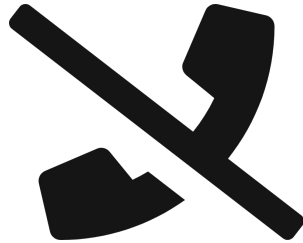


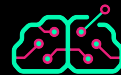
Aproveite esse curso!

- Não se permita sair com dúvidas.
- Aproveite a presença do instrutor ou instrutora. É uma das vantagens de cursos ao vivo sobre livros, por exemplo.
- Ponha o celular em modo avião ou silencioso.
- Faça os exercícios no horário proposto. Não deixe para “depois”.
- Não acredite cegamente no que o instrutor ou a instrutora diz. Faça seus testes.
- Tenha um caderno ou algum aplicativo de tomar notas sempre à mão.



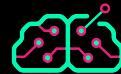
Importante





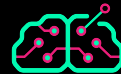
Apresentação da Turma





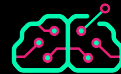
Agenda

- A Engenharia Reversa e Suas Aplicações
- Falando em Binário e Hexadecimal
- Arquivos
- Arquivos Binários
- Cadeias de Texto
 - ASCII, UNICODE e C Strings
- Arquivos Executáveis
 - Cabeçalhos e Campos
 - Segmentos e Seções

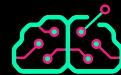


Agenda

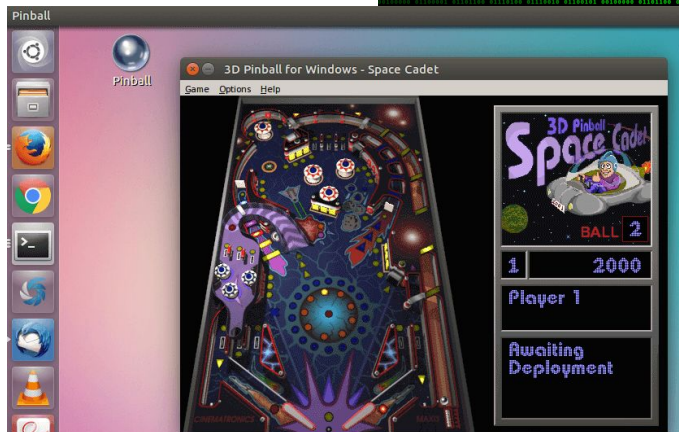
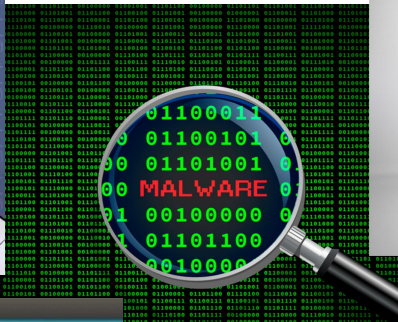
- Introdução à Assembly x86
 - Arquiteturas
 - Registradores
 - Instruções Básicas
 - Funções e Pilha
- Disassembly e Debugging
 - Opcodes, mnemônicos e instruções
 - Breakpoints de software, memória e hardware
 - Patching

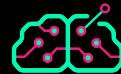


A Engenharia Reversa e Suas Aplicações

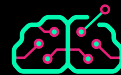


A Engenharia Reversa e Suas Aplicações



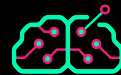


Falando Em Binário e Hexadecimal



Sistemas de Numeração

- O que é um número?



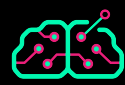
Decimal

- Dez símbolos
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Contagem
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, **9**, 10...





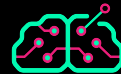
Decimal - Passo a passo

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109



Um sistema quaternário

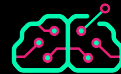


Binário

- Dois símbolos
 - 0, 1

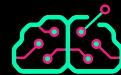
Dígito binário -> **binary digit** -> bit

- Contagem
 - 0, **1**, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010...



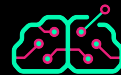
Octal

- Oito símbolos
 - 0, 1, 2, 3, 4, 5, 6, 7
- Contagem:
 - 0, 1, 2, 3, 4, 5, 6, **7**, 10 ... 17, 20



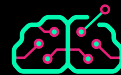
Hexadecimal

- Dezesseis símbolos
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Contagem:
 - ..., D, E, **F**, 10, 11, 12, ...



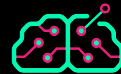
Representações em diferentes programas

	C	IDA	x64dbg	Hiew	Python	bc*	VB
Decimal	10	10	.10	10t	10	10	10
Binário	0b1010	?	-	?	0b1010	ibase=2; 1010	-
Hexa	0xa	Ah / 0xa	a	A	0xa	ibase=16; A	&HA
Octal	012	?	-	?	0o	ibase=8; 12	&o12



Lab 01

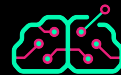
Sistemas de Numeração



Lab 01 - Sistemas de numeração

Faça este lab sem a ajuda do computador.

1. Sem efetuar nenhuma conversão, conte de zero a doze em binário.
2. Efetue as seguintes operações e escreva o resultado em hexa:
 - a) $0xf + 1$
 - b) $0b1010 + 1$
 - c) $0x19 + 2$
 - d) $0x61 - 0x20$



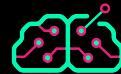
Lab 01 - Sistemas de numeração

3. Analise o seguinte programa em C:

```
int esp = 0;
for (int i=0; i<8; i++) {
    printf("%x\n", esp);
    esp = esp + 4;
}
```

Que valor (em hexa) da variável `esp` em cada iteração do loop?

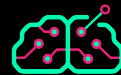
i	0	1	2	3	4	5	6	7
esp								



Lab 01 - Sistemas de numeração

4. (Extra) Converta:

- a) 0x1e8 para decimal.
- b) 127 para binário.
- c) 0b111100001111000011001010 para hexadecimal.
- d) 0x1ff para octal.

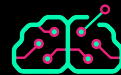


Lab 01 - Sistemas de numeração - Respostas

1. Sem efetuar nenhuma conversão, conte de zero a doze em binário.

Lembrar da regra: quando os símbolos acabam, se utiliza um a mais

decimal	0	1	2	3	4	5	6	7	8	9	10	11	12
binário	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100



Lab 01 - Sistemas de numeração - Respostas

2. Efetue as seguintes operações e escreva o resultado em hexa:

a) $0xf + 1$

- f é o último dígito hexa, então somando 1, teremos **10** (que é igual a 16 em decimal)

b) $0b1010 + 1$

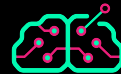
- Já sabemos, pelo exercício 1, que 1010 em binário é igual a 10 em decimal. Se é 10, no sistema hexadecimal, é A
- Em hexadecimal, $A + 1 = \mathbf{B}$

c) $0x19 + 2$

- Em hexa, depois do 9, vem A. Então, se temos $19 + 1$, vamos para 1A. E $1A + 1 = \mathbf{1B}$

d) $0x61 - 0x20$

- $61 - 20 = \mathbf{41}$ em hexa



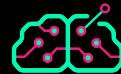
Lab 01 - Sistemas de numeração - Respostas

3. Analise o seguinte programa em C:

```
int esp = 0;
for (int i=0; i<8; i++) {
    printf("%x\n", esp);
    esp = esp + 4;
}
```

Que valor (em hexa) da variável `esp` em cada iteração do loop?

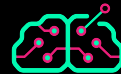
i	0	1	2	3	4	5	6	7
esp	0	4	8	c	10	14	18	1c



○ Byte

- Unidade de medida na computação. 00000000 até 11111111, 0 até 255, 0 até 0xff
- Comumente 8 bits.

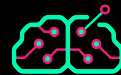
Medida	Tamanho (Intel)	Nomenclatura Intel
<i>nibble</i>	4 bits	
<i>byte</i>	8 bits	BYTE
<i>word</i>	16 bits	WORD
<i>double word</i>	32 bits	DWORD
<i>quad word</i>	64 bits	QWORD



O Byte

- A cada 4 bits, temos um dígito hexa
- Exemplo: 00001111 em hexa
 0 F

Então temos 0x0F



Bytes em arquivos

Criando um arquivo:

```
$ echo -n 'ab' > arquivo.txt
```

Visualizando o arquivo:

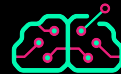
```
$ xxd -g1 arquivo.txt  
00000000: 61 62
```

```
$ xxd -bg1 arquivo.txt  
00000000: 01100001 01100010
```

Tamanho do arquivo:

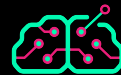
```
$ wc -c arquivo.txt  
2 arquivo.txt
```

```
$ ls -l arquivo.txt  
-rw-r--r-- 1 kali kali 2 Jun  9 21:01 arquivo.txt
```



Números Negativos

- Complemento de dois
 - Tome o número em binário.
 - Complete-o com os zeros à esquerda, se necessário (é preciso saber o tamanho do número em *bits*).
 - Inverta os *bits*.
 - Some uma unidade ao resultado final.
- Com um byte podemos representar:
 - De -128 a 127 (decimal)



Números Negativos

Número 10 em binário:

```
>>> 0b1010  
10
```

Zeros à esquerda:

```
>>> 0b00001010  
10
```

Inverter os bits:

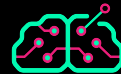
```
>>> 0b11110101  
245
```

Somar uma unidade:

```
>>> 0b11110110  
246
```

Testando:

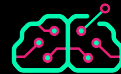
```
>>> import ctypes  
>>> ctypes.c_byte(0b11110110).value  
-10
```



Cálculos com Binários

- Conjunção (E / AND)

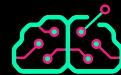
x	y	x & y
0	0	0
0	1	0
1	0	0
1	1	1



Cálculos com Binários

- Disjunção (OR / OU)

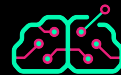
x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1



Cálculos com Binários

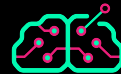
- Disjunção Exclusiva (XOR)

x	y	$x \wedge y$
0	0	0
0	1	1
1	0	1
1	1	0



Lab 02

Operações matemáticas



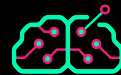
Lab 02 - Operações matemáticas

1. Converta -1 em 32-bits para hexadecimal (use complemento de dois).

2. Calcule:

a) $5 \mid 7$, com resposta em binário e decimal.

b) $9 \& 1$, com resposta em hexadecimal.



Lab 02 - Operações matemáticas - Respostas

1. Converta -1 em 32-bits para hexadecimal (use complemento de dois).

- 00000000000000000000000000000001
- 11111111111111111111111111111110
- 11111111111111111111111111111111
- **FFFFFFFF**

2. Calcule:

a) $5 \mid 7$, com resposta em binário e decimal.

$$\begin{array}{r} 1 \ 0 \ 1 \qquad 5 \\ 1 \ 1 \ 1 \qquad 7 \\ \hline 1 \ 1 \ 1 \qquad 7 \end{array}$$

b) $9 \& 1$, com resposta em hexadecimal.

$$\begin{array}{r} 1 \ 0 \ 0 \ 1 \\ 0 \ 0 \ 0 \ 1 \\ \hline 0 \ 0 \ 0 \ 1 \end{array}$$