



Apresenta

**Informações sobre compressão LZ?? e
suas variantes nos jogos Phantasy Star
Generations 1 e 2 [PS2]**

Brasil, 06 de Março de 2009

Rev. 0

Informações sobre compressão LZ?? Usada no Jogo Phantasy Star Generations 1 [PS2]

Todo arquivo comprimido utiliza um header da seguinte forma.

CM(ASCII) = 2 bytes

(Tamanho descomprimido) = 4 bytes

(Tamanho comprimido) = 4bytes

Início dos dados = (até o tamanho comprimido indicado)

FLAG ID = (Logo vou explicar o que é)

Bem primeiramente a compressão foi um pouco demorada para descobrir, tendo que ler bastantes documentos informando sobre outras compressões sobre LZSS (que praticamente não achei uma que servisse até achar uma mas já tinha entendido-a na marra), LZ77, LZW, LZX, LZRW, LZW, Constant Mixing (porque achei que podia ser ela pelo CM no header) e Huffman (“Um insulto ao cérebro do ser humano”).

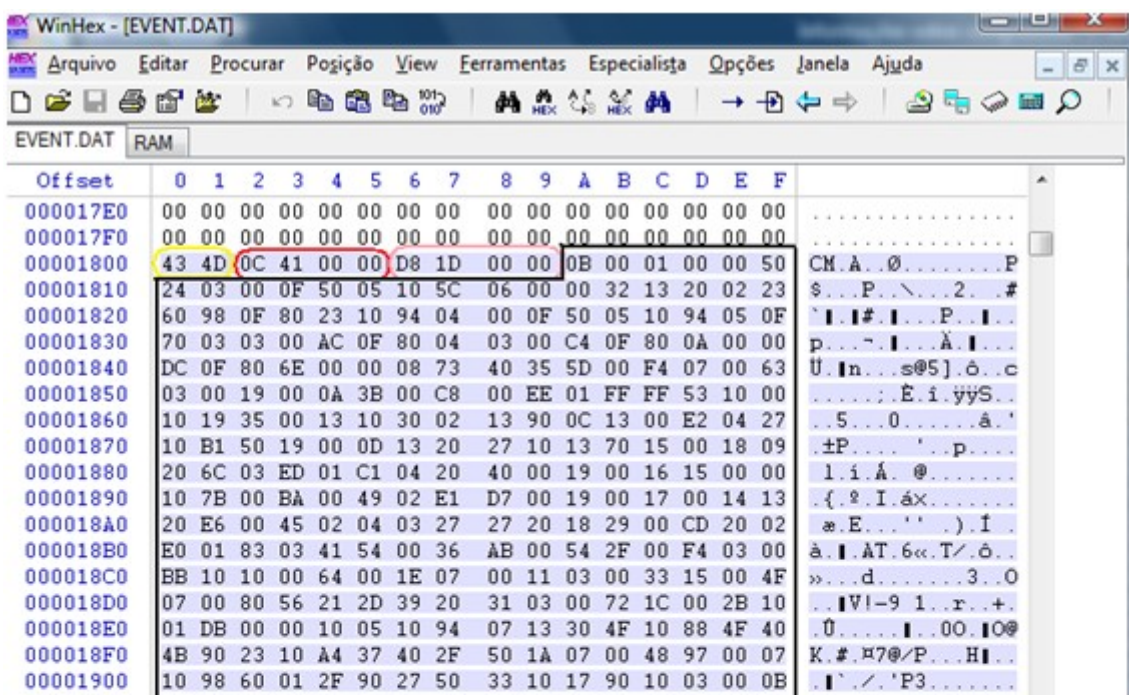
Neste documento vou demonstrar como a compressão funciona ilustradamente para um fácil entendimento sobre a compressão e sua lógica.



Olhe a estrutura do arquivo comprimido onde o início dele é o CM (434D)

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000017E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000017F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00001800	43	4D	0C	41	00	00	D8	1D	00	00	0B	00	01	00	00	50	CM A. @.....P
00001810	24	03	00	0F	50	05	10	5C	06	00	00	32	13	20	02	23	\$...P.\...2...#
00001820	60	98	0F	80	23	10	94	04	00	0F	50	05	10	94	05	0F	! .# ...P.
00001830	70	03	03	00	AC	0F	80	04	03	00	C4	0F	80	0A	00	00	p... ...Ä
00001840	DC	0F	80	6E	00	00	08	73	40	35	5D	00	F4	07	00	63	Ü n...s@5]..ô..c
00001850	03	00	19	00	0A	3B	00	C8	00	EE	01	FF	FF	53	10	00:É.i.ÿÿS..
00001860	10	19	35	00	13	10	30	02	13	90	0C	13	00	E2	04	27	..5...0.....ä..
00001870	10	B1	50	19	00	0D	13	20	27	10	13	70	15	00	18	09	..±P.....p.....
00001880	20	6C	03	ED	01	C1	04	20	40	00	19	00	16	15	00	00	l.i.Ä. @.....
00001890	10	7B	00	BA	00	49	02	E1	D7	00	19	00	17	00	14	13	{.º.I.áx.....
000018A0	20	E6	00	45	02	04	03	27	27	20	18	29	00	CD	20	02	æ.E.....')..Í..
000018B0	E0	01	83	03	41	54	00	36	AB	00	54	2F	00	F4	03	00	à.. ..AT.6<<.T/..ô..
000018C0	BB	10	10	00	64	00	1E	07	00	11	03	00	33	15	00	4F	>>..d.....3...0
000018D0	07	00	80	56	21	2D	39	20	31	03	00	72	1C	00	2B	10	.. V -9 1..r...+..
000018E0	01	DB	00	00	10	05	10	94	07	13	30	4F	10	88	4F	40	..Û..... ...00.. O@
000018F0	4B	90	23	10	A4	37	40	2F	50	1A	07	00	48	97	00	07	K.#.#7@/P...H
00001900	10	98	60	01	2F	90	27	50	33	10	17	90	10	03	00	0B	.. /..P3.....
00001910	50	4A	EF	41	7B	11	8E	10	0E	04	00	07	F5	40	0E	97	PJiA{.. ...ö@..
00001920	01	07	15	41	0E	CD	01	07	F9	40	0E	51	01	07	41	41	...A.Í..ù@.Q..AA
00001930	0E	C5	01	07	27	01	00	10	0F	03	00	13	CF	40	6F	90	..Ä.....I@o..
00001940	00	00	3C	05	20	12	23	22	40	00	00	FE	2E	42	3B	01	...<...#"@..p..B;..
00001950	12	7F	00	03	13	10	00	30	13	20	22	5B	42	16	22	080..."[B..."
00001960	00	22	04	00	0F	90	05	50	4F	10	04	3B	C0	B9	10	22PO...:Ä¹..
00001970	30	73	70	05	23	C0	1B	50	07	91	F1	30	07	30	C1	10	0sp.#Ä.P.ºñ0.OÄ..
00001980	07	10	0B	1F	21	00	30	1F	51	D7	90	21	5F	A1	53	F1	...!..0.Qx..! Sñ
00001990	53	B1	A1	11	53	51	1C	53	81	1D	3B	F1	63	80	6B	31	Sti..SQ.S...;ñc k1
000019A0	00	50	D3	50	33	11	80	D3	F0	63	61	00	60	1B	A0	63	..POP3.. Óöca...c
000019B0	10	E7	90	A5	10	73	10	97	13	A3	30	0B	54	2F	70	93	..ç.¥.s.. ..º0.T/p
000019C0	50	B7	30	0D	1F	F0	73	03	62	10	56	27	00	3C	93	00	P..0...ö.s..b..V'<
000019D0	01	10	23	90	CD	00	27	6B	F1	63	F1	2F	91	1D	11	4F	..#.Í..kñcñ/...0
000019E0	F1	4F	D1	20	97	80	89	11	4F	51	3D	12	4F	F1	8B	F2	ñON Ñ ..OQ=.Oñ ö
000019F0	1C	11	43	71	C0	62	20	5F	91	EA	09	00	20	15	00	E0	..CqÄb_'.é.....ä
00001A00	00	1B	B0	02	D7	24	00	30	20	01	E8	1B	80	4F	F2	67	...x\$.0..è.. Oög
00001A10	71	07	10	27	11	FB	F0	FB	F0	19	29	00	14	00	40	01	q...'.úäüä.)...@.
00001A20	C8	00	FF	FF	91	51	CB	13	19	79	00	1D	17	F0	17	00	É.ÿÿ'QE...y...ö..
00001A30	04	99	03	17	F0	19	00	1F	B1	01	40	01	F8	17	80	13ö...±..@..ö..
00001A40	10	64	13	00	26	23	05	2B	B0	20	D1	03	E2	00	B2	2B	..d..&#+.ºÑ.Ä.²+
00001A50	A0	20	2B	00	E2	01	00	13	90	21	C9	02	9E	01	27	B0	+.ä... !É.. ..'
00001A60	21	27	00	9E	01	27	90	37	36	B7	33	F7	10	10	2E	04	!76-3+....
00001A70	0F	50	3F	11	B8	1B	0F	70	7F	12	98	B6	01	0F	50	1F	..P?...p... ñ...P.
00001A80	11	24	3D	0F	70	27	11	F0	0F	80	0C	1F	10	3F	0F	70	..\$.=p'.ö... ...?..p
00001A90	95	12	D4	0F	80	B1	14	8C	6A	05	0F	50	B5	14	88	2FÓ ±.. j...Pp.. /
00001AA0	0F	70	AD	14	D0	34	0F	70	C9	24	39	0F	70	1F	19	16	..p-.Ä4..pÉ\$9..p...
00001AB0	B5	02	0F	50	69	12	7C	2A	0F	70	6D	12	44	2C	0F	70	µ..Pi.. *.pm..D...p
00001AC0	EB	10	8C	A4	03	0F	50	64	03	00	60	CB	45	3B	00	70	ë.. ñ..Pd...EE...p
00001AD0	00	88	0B	00	46	D7	20	04	07	20	B9	16	49	67	40	0BFx...'.Ig@.
00001AE0	16	DC	9E	04	2B	10	57	10	07	10	25	31	01	24	49	00	..Ü ..+..W...%1.\$I.
00001AF0	F7	F2	6B	15	38	01	47	E4	23	14	47	74	6F	94	03	10	÷òk..8.Gä#.Gto ..

Agora olhe o screen relacionando a estrutura do arquivo.



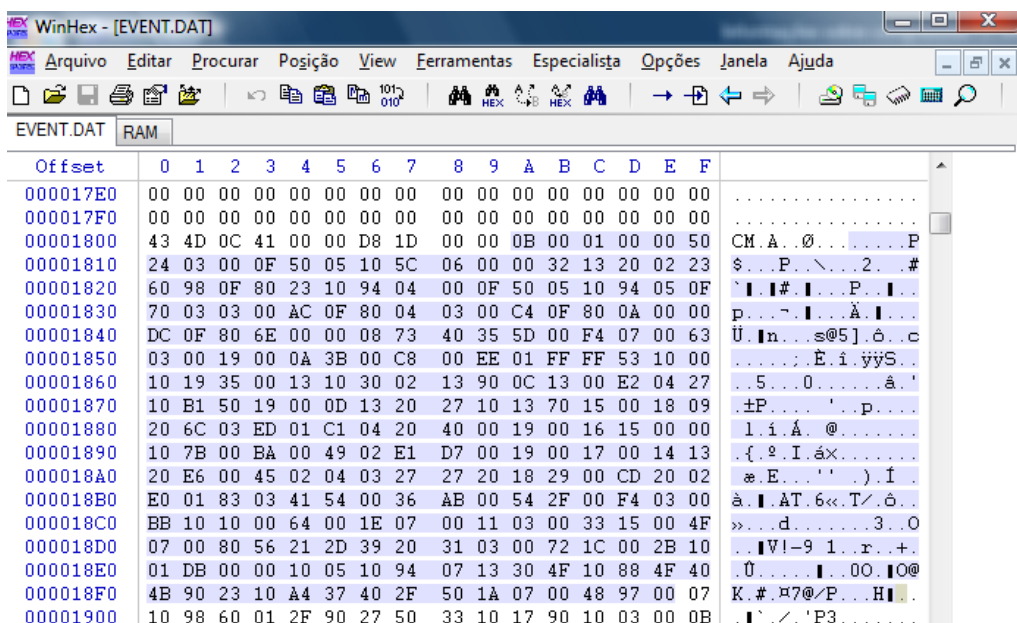
**O querido CM
CoMpressed?
Contant Mixing?
Sei lá...**

**Tamanho dos dados
após a descompressão**

**"A Manteiga do seu pão"
Depois eu explico o porquê.**

**E o preto é DADOS
DADOS**

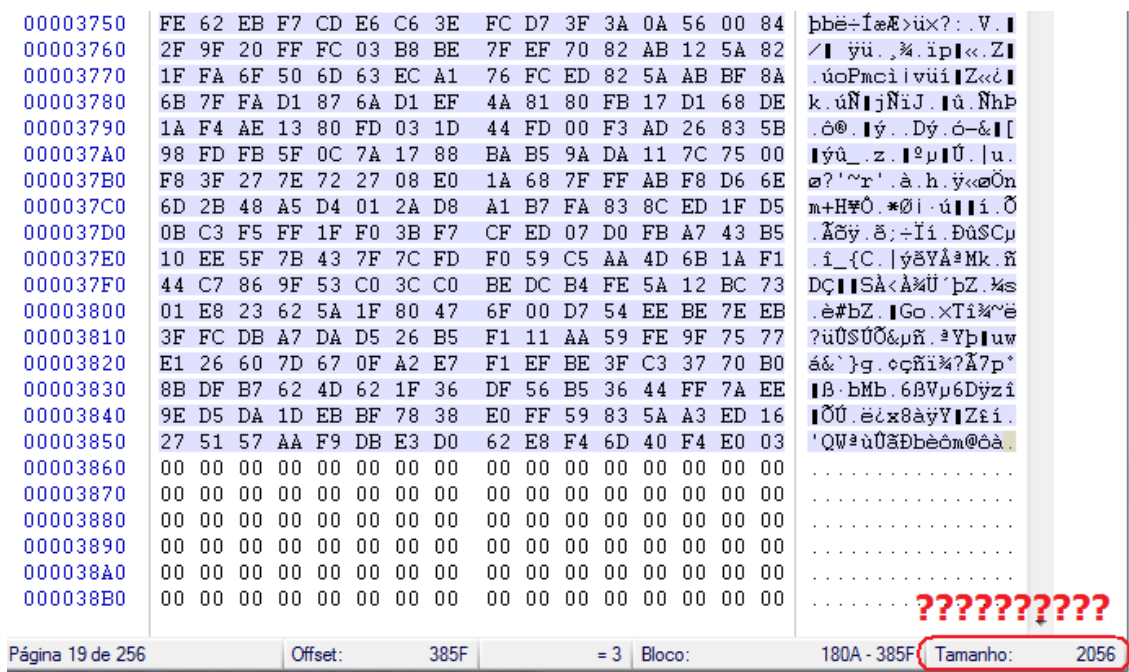
Bem, o **D81D0000** que invertendo, pelo fato que o processador do PS2 é Little Endian, é **0x1DD8** então deveria ser o tamanho do arquivo comprimido, então vamos ver se é comprovado.



Começo a contar a partir do 11º byte pelo fato que os 10 bytes sendo:

**CM = 2 BYTES
0C410000 = 4 BYTES
D81D0000 = 4 BYTES**

**Obviamente:
2+4+4 = 10
Então 11º é o início.**



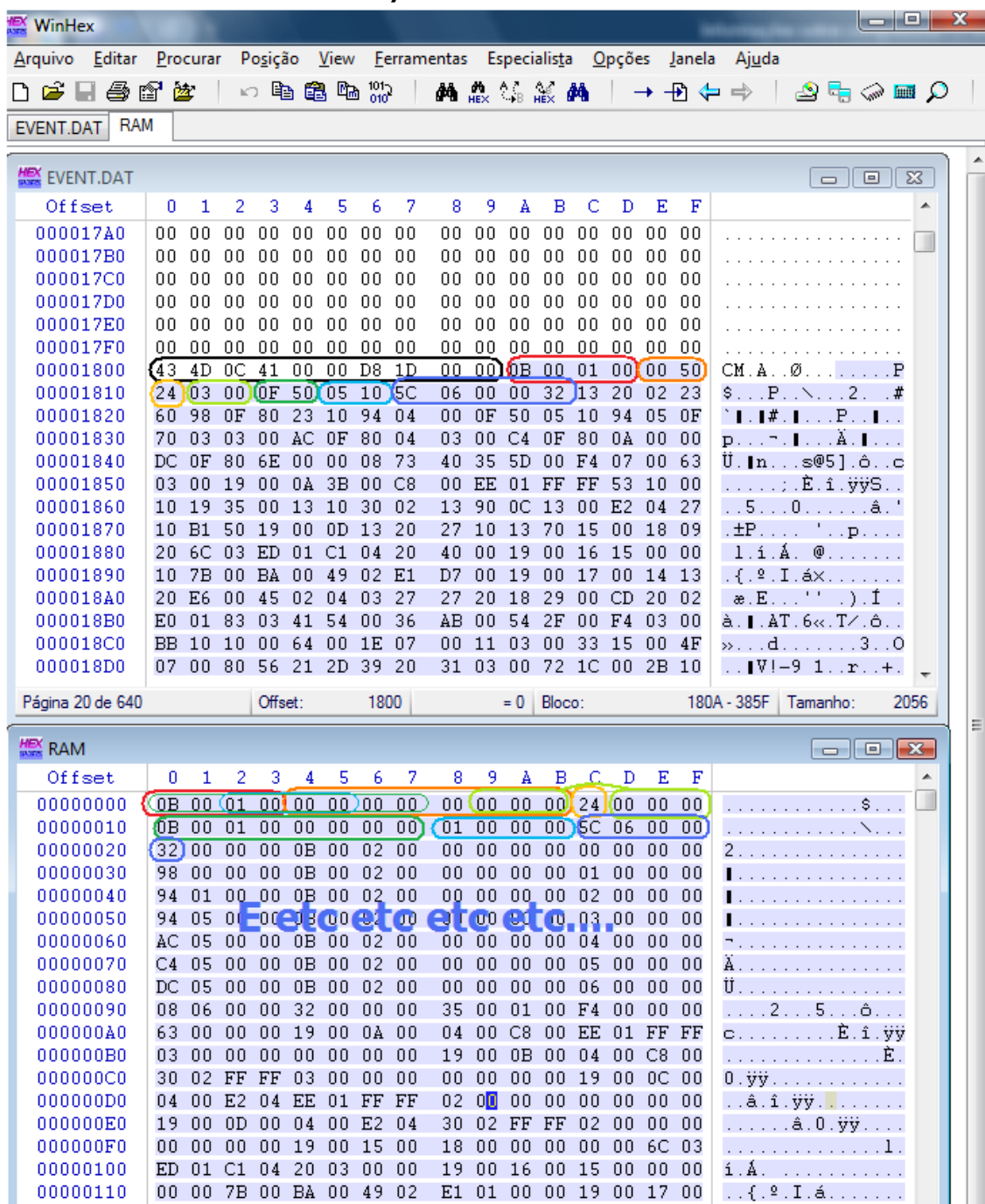
Chegando ao final do arquivo, fiquei estranhando com o seguinte fato:

0x1DD8 ≠ 0x2056

Isso é mais que óbvio.

Então vamos olhar pela RAM, já que o arquivo descomprimido está lá.

***Procedimento de dumpar (extrair) RAM” (não contém maiores detalhes neste documento).**



Assim sendo vou explicar direitinho como funciona a compressão:

0B 00 01 00 → Não posso pegar em nenhum outro lugar (Claro, está no começo do arquivo, né)

Sendo que tem 00 00 00 00 00 00 00 00 repetidos então lá vem a primeira

00 50 → 00 significa o que byte eu vou repetir - 50 quebrando o byte no meio ocasiona 5 sendo que a compressão já usa 2 bytes no arquivo então seria inútil usa o 0 ou 1, então seria 05 + 2 + 1(porque no HEX começa a contar do zero) 8 bytes.

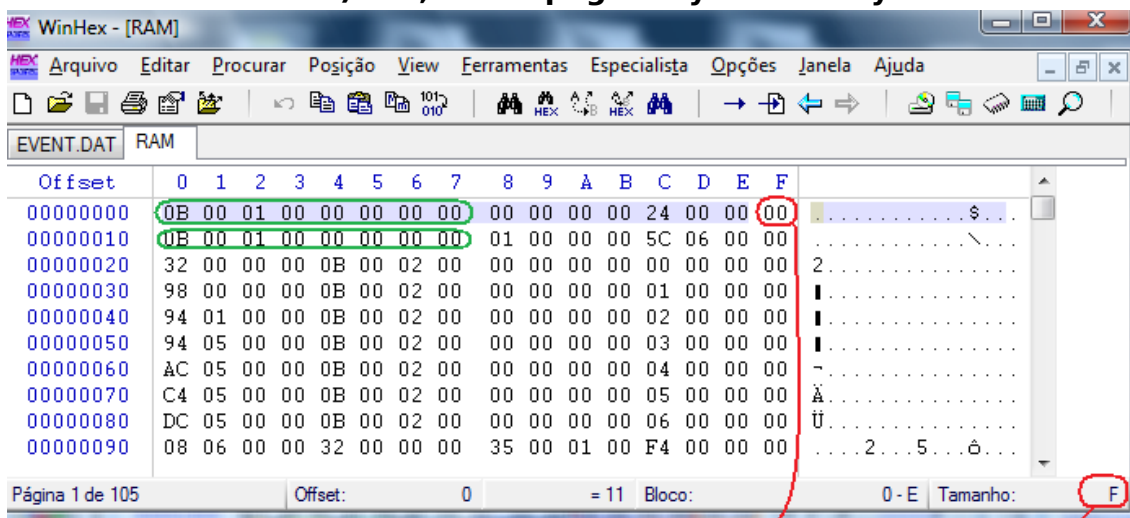
24 - sem comprimir

00 00 00 da para pegar “emprestado” um pouco atrás.

03 00 → sendo 00 = 00 + 3 (HEX começa a contar e tal) utiliza três bytes - o 03 serve para buscar o mais próximo 00 00 00 logo atrás.

0B 01 00 00 00 00 00 tem logo atrás então vamos pegar emprestado

0F50 → 50 quebrando o byte da 5 sendo 5 + 2 + 1(HEX começa a contar do zero e etc) = 8, então pega 8 bytes a 0F bytes atrás.



No LZ o zero conta também

OF

Com isso os outros dois nem preciso explicar, certo.

Mas olhando assim fica faltando alguma coisa. Onde estão os FLAGS? Praticamente não vejo nenhuma indicativa de uso de compressão ou não, então não poderia criar um LZ randomicamente então.

Lembre da **manteiga do seu pão** logo acima? Para te poupar o trabalho de subir umas páginas:

Informações sobre compressão LZ?? no jogo PSG 1 – Po.B.R.E. RomHackers

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000017E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000017F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00001800	43	4D	0C	41	00	00	D8	1D	00	00	0B	00	01	00	00	50	CM: A 00.....P
00001810	24	03	00	0F	50	05	10	5C	06	00	00	32	13	20	02	23	Sh: P: N: 2: #
00001820	60	98	0F	80	23	10	94	04	00	0F	50	05	10	94	05	0F	1: # I: P: I: #
00001830	70	03	03	00	AC	0F	80	04	03	00	C4	0F	80	0A	00	00	p: I: I: A: I: #
00001840	DC	0F	80	6E	00	00	08	73	40	35	5D	00	F4	07	00	63	U: In: s: #5: I: #
00001850	03	00	19	00	0A	3B	00	C8	00	EE	01	FF	FF	53	10	00E: i: yyS: ..
00001860	10	19	35	00	13	10	30	02	13	90	0C	13	00	E2	04	27	..S: ..0: ..: A: #
00001870	10	B1	50	19	00	0D	13	20	27	10	13	70	15	00	18	09	±P: ..: ..: P: ..: #
00001880	20	6C	03	ED	01	C1	04	20	40	00	19	00	16	15	00	00	1: i: A: #
00001890	10	7B	00	BA	00	49	02	E1	D7	00	19	00	17	00	14	13	(: # I: I: #K: ..: ..: #
000018A0	20	E6	00	45	02	04	03	27	27	20	18	29	00	CD	20	02	w: E: ..: ..: # I: #
000018B0	E0	01	83	03	41	54	00	36	AB	00	54	2F	00	F4	03	00	A: I: AT: 6: # I: #
000018C0	BB	10	10	00	64	00	1E	07	00	11	03	00	33	15	00	4F	»: ..: d: ..: ..: 3: ..: O
000018D0	07	00	80	56	21	2D	39	20	31	03	00	72	1C	00	2B	10	»: I: V: I: -9: 1: ..: ..: #
000018E0	01	DB	00	00	10	05	10	94	07	13	30	4F	10	88	4F	40	.0: ..: ..: I: ..: ..: 00: I: O: #
000018F0	4B	90	23	10	A4	37	40	2F	50	1A	07	00	48	97	00	07	R: #: B: 7: #/P: ..: ..: H: I: #
00001900	10	98	60	01	2F	90	27	50	33	10	17	90	10	03	00	0B	I: I: I: I: #P: 3: ..: ..: #

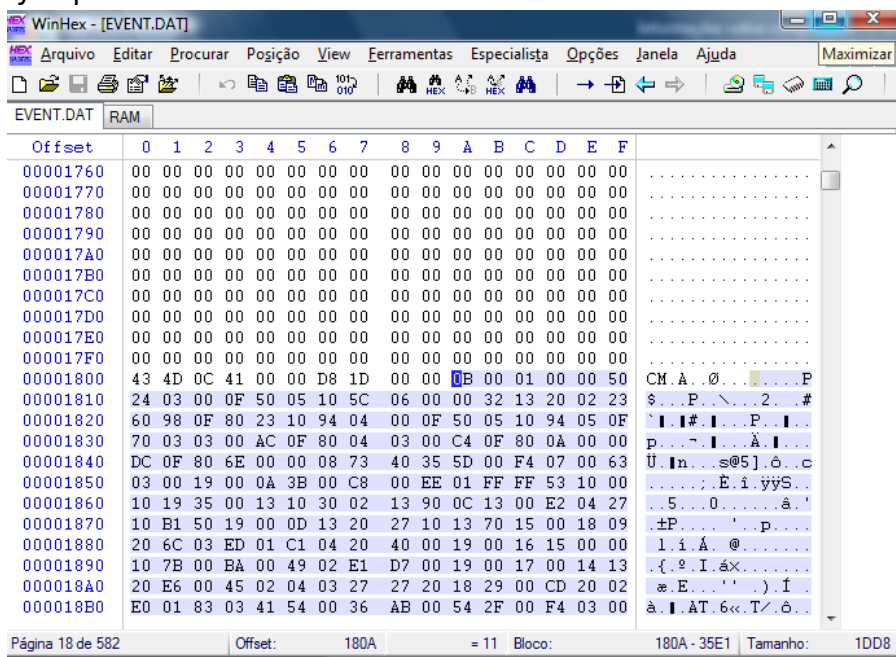
O querido CM
CoMpressed?
Contant Mixing?
Sei lá...

Tamanho dos dados
após a descompressão

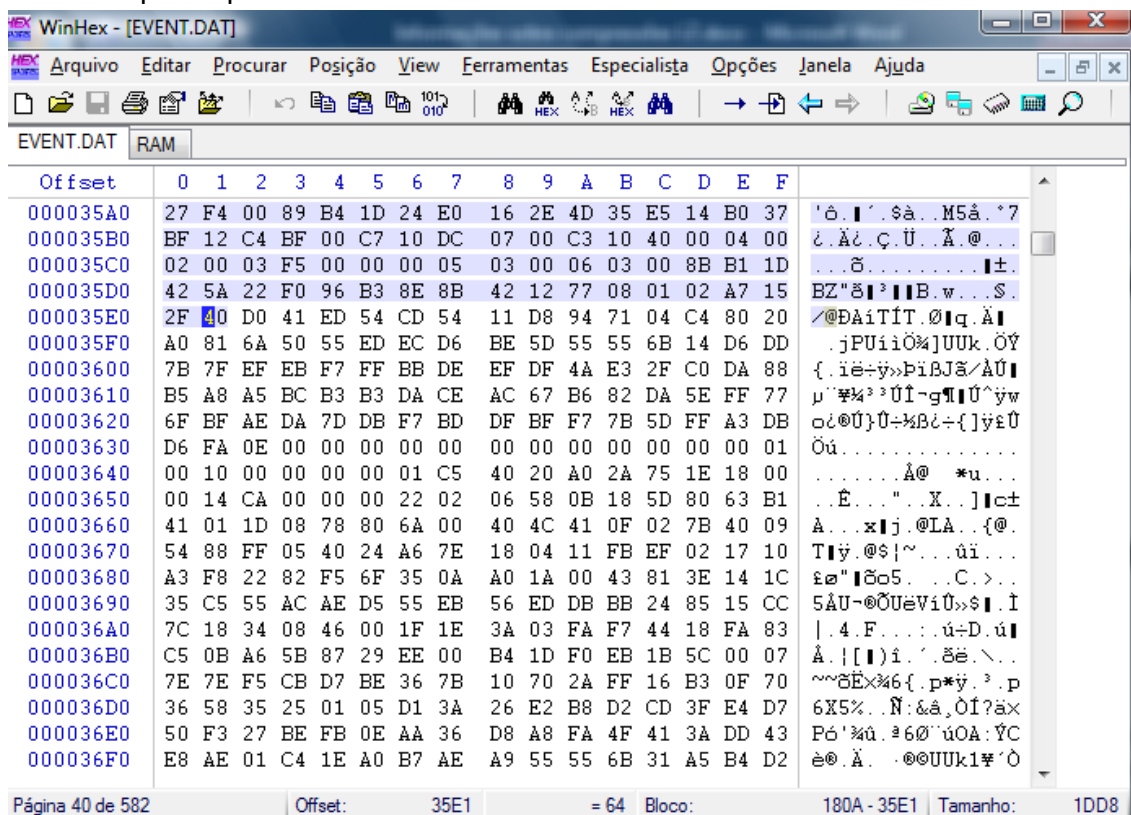
"A Manteiga do seu pão"
Depois eu explico o porquê.

E o preto é DADOS
DADOS

Olhe bem o valor, então o problema é bem mais simples que você imagina. Vamos contar o tamanho dos dados comprimidos, lembrando que é a partir do 11º byte por causa do header.



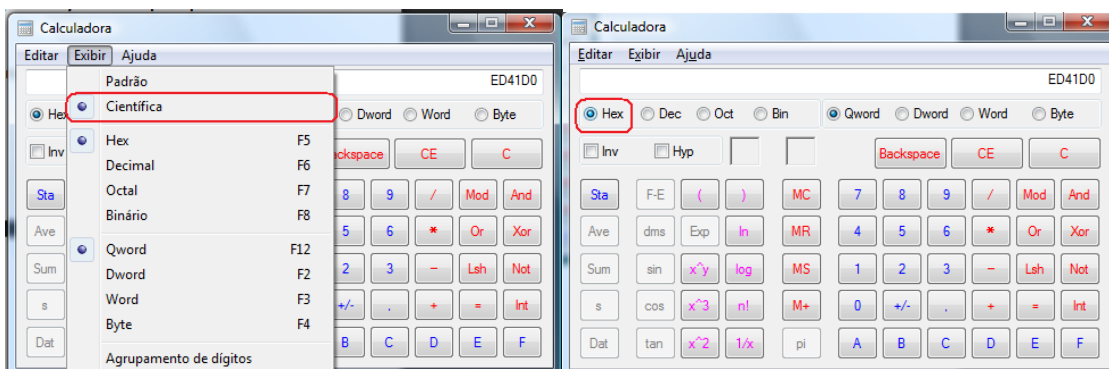
Parando por aqui dando o valor exato dos dados.



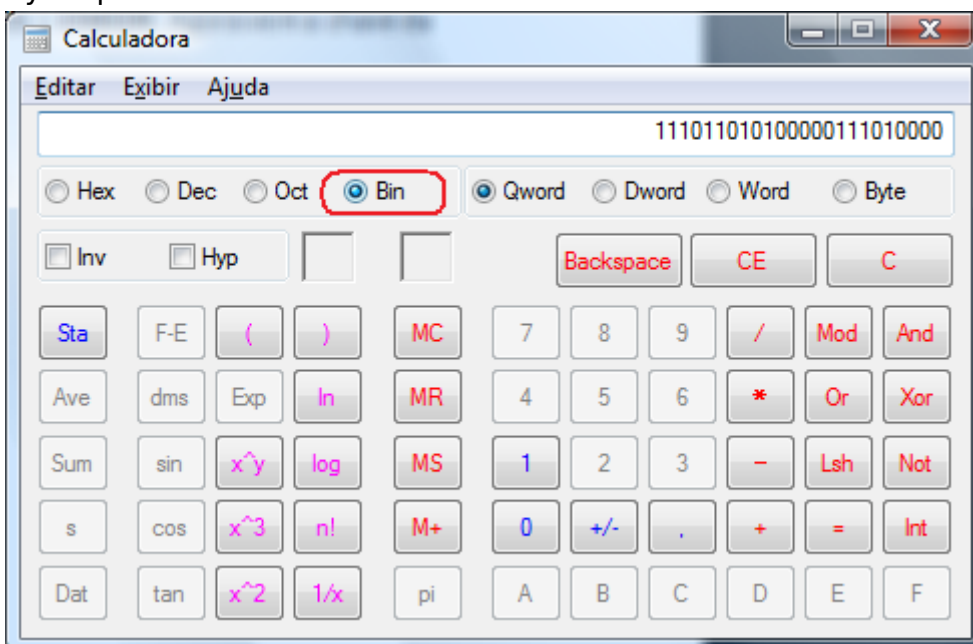
Então o que seria os outros bytes, não parece indicar quantos bytes devem pular para a próxima compressão LZUnknow?

Na verdade explica sim.

Primeiramente o PS2 é Little Endian, então temos que inverter a leitura dos dados. Vamos pegar um exemplo com apenas os três bytes **D0 41 ED**. Use a calculadora do seu sistema operacional e coloque em modo científica.



Mude o valor da calculadora para HEXADECIMAL. Como processador do PS2 e etc. lê os bytes e também bits ao contrário esteja ciente disso vamos colocar os bytes emprestados ao contrário **D041ED = ED41D0**. Agora vem a chave da descompressão, mude esses bytes para binários.



Agora explicando que:

0 = 1 byte sem compressão

E

1 = 2 bytes indicando uma compressão LZ

Assim lendo de trás para frente temos quatro zeros, certo? Assim são descomprimidos,

0B	00	01	00
0	0	0	0

Um **1** encontrado seguido de **0** e mais dois **1**:

LZ		DESCOMPRI	LZ	
00		MIDO	00	
00	50	24	03	00

Informações sobre compressão LZ?? no jogo PSG 1 - Po.B.R.E. RomHackers

1	0	1
----------	----------	----------

Brincando mais um pouco:

11000001

LZ		LZ						LZ	
0F	50	05	10	5C	06	00	00	32	13 20
1		1		0	0	0	0	0	1

Assim os bytes **0F 50**, **00 32** e **20 02** são compressões **LZ** assim indicando cada seção **LZ** por números binários indicado por **1**.

Isso explica o funcionamento da compressão usada nos jogos **Phantasy Star Generations 1 e 2 [PS2]**.

O jogo pode ser modificado sem a necessidade de um recompressor, mas teria que remanejar a ISO já que os dados descomprimidos ficariam obviamente, maiores que o original.



<http://www.romhackers.org/>

TUTORIAL FEITO E COLABORADO POR:

RodolfoRG

orakio_rob

Ignitz

gamer_boy

