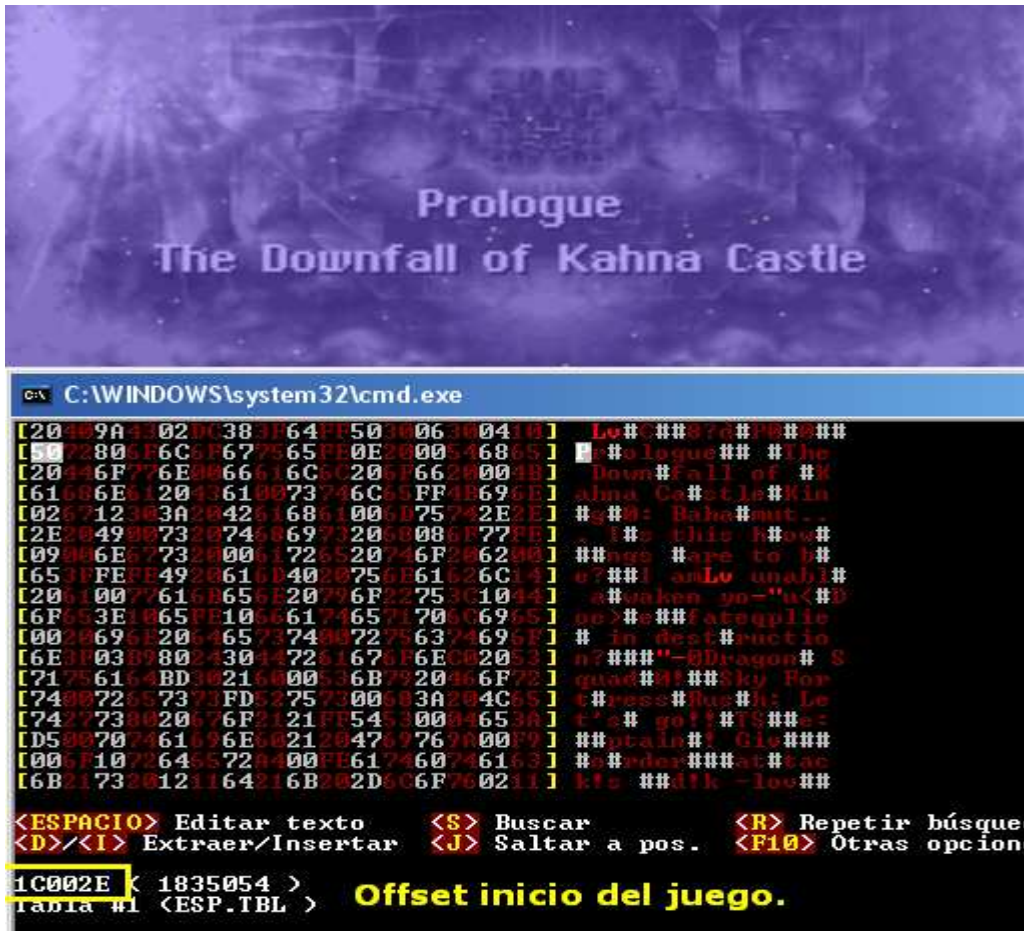


PUNTEROS LZSS EN BAHAMUT LAGOON. By Darke.

INDICADORES.

Bueno, vamos a intentar comprender la compresión LZSS del Bahamut Lagoon, a ver si consigo explicársela.

Empecemos con la imagen del principio del texto del juego, en el que podéis ver también el Offset del inicio del juego, os aconsejo que tengáis el editor hexadecimal abierto y vayáis mirando a la vez que lo explico.



Bien, si veis la palabra Prologue, en el editor aparece como Pr<80>ologue, a este byte que nos encontramos en mitad del texto, pero que realmente no modifica en nada el texto (como podéis observar en la imagen del juego) lo vamos a llamar **indicador**. No se si su nombre técnico y real es ese, yo me referiré a él con ese nombre.

El **indicador** lo que hace es indicar, del texto que tiene en adelante, que es texto sin compresión y donde hay un puntero LZSS, el cual si es texto comprimido. ¿Y cómo

nos lo indica? Pues bien, si pasamos ese byte, <80> a binario con la calculadora del Windows, nos da que:

80 en hexadecimal = 10000000 bits

Recordad que un byte son 8 bits, si la calculadora nos mostrase “1001”, para nosotros sería realmente “00001001”, es decir, con 8 dígitos. Pues bien, este número hay que leerlo de derecha a izquierda, de modo que primero hay 7 ceros y después un 1, esto quiere decir que los 7 bytes siguientes son letras sin comprimir y el octavo byte es un **puntero LZSS**. Nos vamos al editor, contamos y comprobamos.

Pr<80>ologue<FE><0E><20><00>The...
1234 5 6 7 8

El byte nº 7 es como si fuese texto, en concreto es el salto de línea, pero no es ningún puntero ni nada por el estilo.

IMPORTANTE: ¡Los punteros LZSS aunque son dos bytes, se contabilizan como uno sólo!

Luego explicaremos como funciona el puntero LZSS, antes vamos a comprender del todo a los indicadores.

Bien, cada 8 bytes, SIEMPRE, habrá un indicador para saber si los 8 bytes siguientes son texto descomprimido o puntero LZSS.

Si os fijais, justo después del octavo byte, efectivamente viene otro indicador, en este caso es el <00>. Si lo pasamos a binario sería 00000000. Es decir, que los 8 bytes siguientes son todo de texto descomprimido, no hay ningún puntero. Comprobémoslo.

<00>The_Down<00>fall_of_<00>Kahna_Ca<00>stle...
1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8

¿Veis? Cada 8 bytes hay un indicador que dice que es lo que hay adelante.

PUNTEROS LZSS

Bien ahora vamos a explicar lo que son los **punteros LZSS**. El primer puntero LZSS que nos hemos encontrado no es un buen ejemplo para explicaroslo, ya que apunta hacia antes de donde comienza el texto, aunque su función es la misma, lo único que hace es copiar unos cuantos “espacios” seguidos para mantener centrada la palabra Prologue en el juego. Vamos hacia el siguiente puntero, ¿como sabemos que es un puntero? Porque nos lo indica el indicador xD, si continuamos por donde estábamos antes...

Ca<00>stle<FF>Kin<02>g<12><30>:_Baha<00>mut...

8 bytes
1
2
345678

BIN: 0 1 000000

El indicador <02> pasado a binario es “00000010” pero recordemos que se lee de derecha a izquierda, por tanto hay un primer byte que es texto, luego un puntero LZSS que aunque se compone de dos bytes se considera como uno solo, y luego otros 6 bytes de texto hasta el próximo indicador. Ahora vamos a analizar ese puntero LZSS.



Donde debería aparecer según el juego "King Kahna" aparecen estos códigos:

Kin<02>g<12><30>:

El indicador ya sabemos lo que hace, los dos bytes del puntero LZSS tienen la siguiente función:

El primero <12> hace referencia a cuantos bytes hacia atrás debe irse para buscar el texto que indica ese puntero, 12 en hexadecimal, lo pasamos a decimal para contar con mayor facilidad y son 18, si desde el puntero <12> contamos hacia atrás 18 bytes **PERO IMPORTANTE TENER EN CUENTA: ¡¡Cuando cuentas hacia atrás no se cuentan los indicadores y si pasamos por un puntero LZSS se contará como si el texto ya estuviese descomprimido!!** Comprobémoslo.

```
The_Down<00>fall_of_<00>Kahna_Ca<00>stle<FF>Kin<02>g<12><30>:
      18      17|15|13|11|          9876  5  432      1
              16 14 12 10
```

Bien, hemos llegado a la posición correcta, ya que lo que se debe de copiar con el puntero LZSS para que se coherente con lo que aparece en el juego debe de ser “_Kahna”. ¿Pero como sabe cuantas letras debe copiar?¿porque ha copiado 6 y no 8? Bien, esa indicación la da el segundo de los bytes del puntero LZSS.

El segundo byte del puntero LZSS <30> se debe leer invirtiendo los caracteres del byte, es decir, tenemos <30> lo transformamos a <03> que si lo pasamos a decimal sería “3”. **IMPORTANTE:** A este valor que obtenemos, hay que sumarle +3 **SIEMPRE**. ¿Por qué? Pues porque si un texto es menor de 3 caracteres, no tiene sentido realizar esta compresión, para eso escribes directamente el texto y ya está.

Entonces tenemos 3 +3 = 6, hay que copiar el byte en el que nos hemos posicionado (el marcado con 18) y los 5 siguientes, 6 en total. Es decir que en este caso copiará “_Kahna” que efectivamente es lo que debe de aparecer.

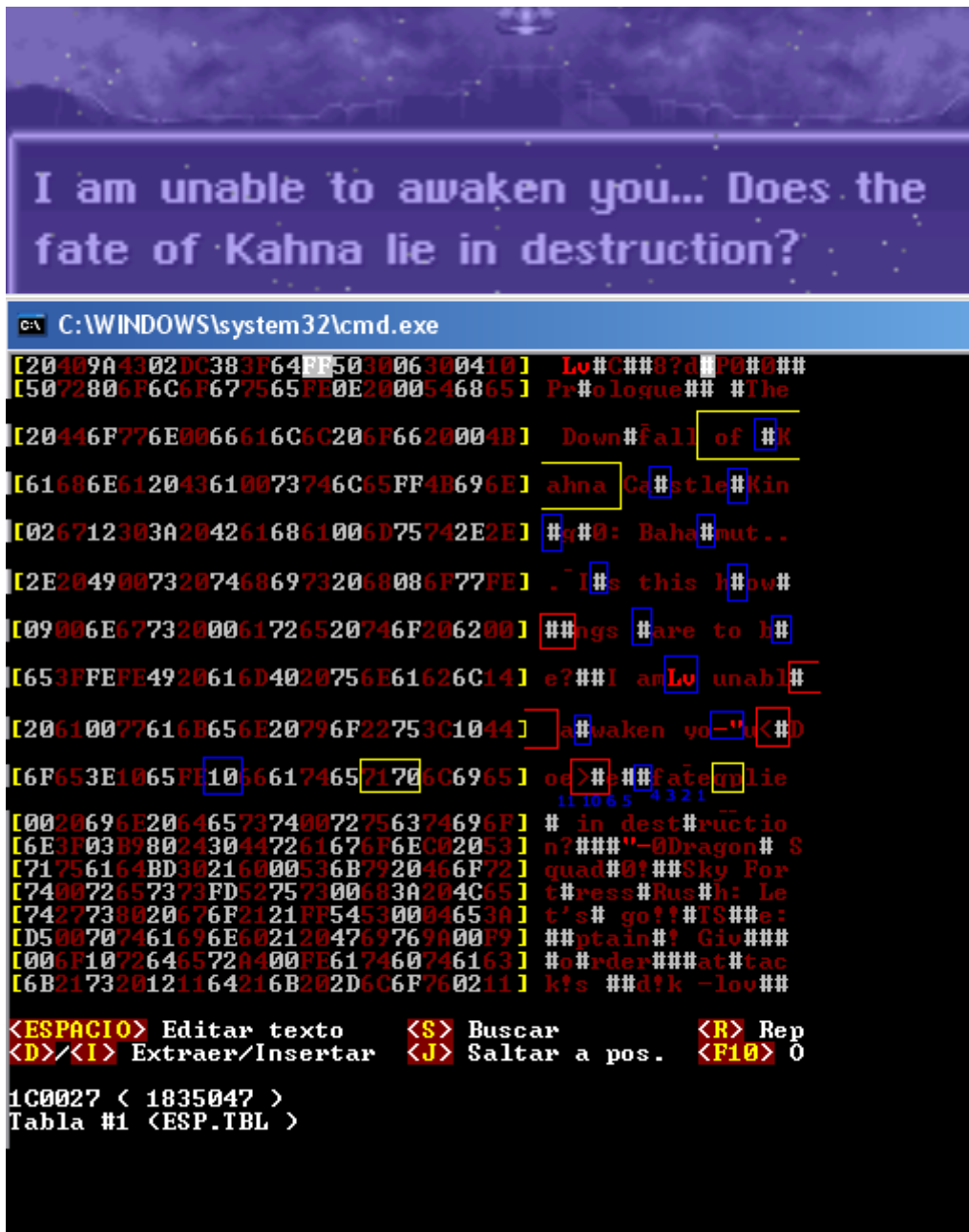
CON ESTO YA LO SABES TODO PERO HAGAMOS UN EJEMPLO MÁS COMPLEJO PARA QUE NO HAYA DUDAS.

Ahora vamos a irnos a un puntero que aparece un poco más adelante. Aquí no lo voy a explicar tanto, intentad comprenderlo con lo explicado anteriormente.

El puntero LZSS es <71><70>

<71> -> Buscará texto 113 bytes atrás, saltándose los indicadores (señalados en azul) y contando a los punteros LZSS como si estuviesen descomprimidos, es decir, si el segundo byte del puntero es <20>, contará al puntero completo (a los 2 bytes) como 5 bytes (<20> = <02> = 2 +3 = 5)

<70> -> Copiará 10 bytes (<70> = <07> = 7 + 3 = 10)



Si lo hacéis con cuidado comprobareis que el primer puntero te marca al primer espacio de “_of_Kahna_” y el segundo puntero te dirá que copies 10 bytes, estos 10 bytes se cuentan igual que los otros, saltándose los indicadores y considerando los punteros LZSS que encuentre por el camino como texto ya descomprimido.

AGRADECIMIENTOS: Esto no habría sido posible de no ser gracias a la ayuda brindada por SkyBladeCloud, que me ayudo a terminar de comprender la compresión de este juego.

TUTORIAL HECHO POR DARKE.