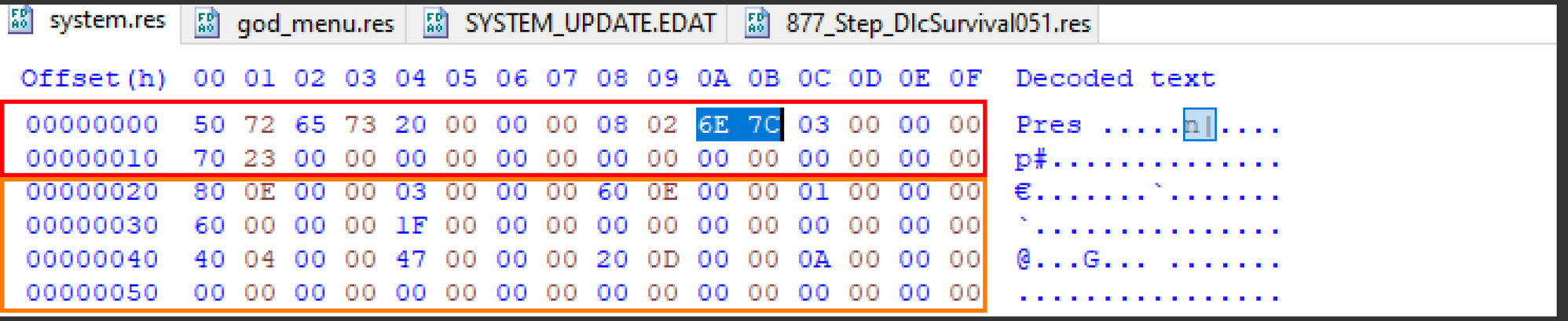


[Start Point]
~ 0x00: Magic/Header: 4 bytes (Pres)
~ 0x04: 4 bytes, little endian offset (0x20). jumps to table offset
~ 0x08: byte, group table count. (always set to **08**)
~ 0x09: some kind of value?. changes on different RES archives.
~ 0x0A: 2 bytes checksum?. not sure what to do with it. (but i guess it could be a a MD5 hash due to **libmd5.prx** being used)
~ 0x0C: 4 bytes, seems to be stuck at 3, must be a version of that RES archive?
~ 0x10: 4 bytes little endian offset, jumps you to where certain chunks of data are stored (related to ToC)
~ 0x14->0x1F: lengthy zeroes, 4 bytes each. not really useful mostly. but can contain other data like **0x10** out of nowhere so keep an eye out.

[Table Offsets] ranging: *0x20-0x5F*
~ 0x20: empty usually on some other .res archive, but it represent as ToC offset (`00 15 00 00` -> 0x1500 for example)
~ 0x24: empty usually on some other .res archive, but it represent as count for ToC (`03 00 00 00` -> 3 for example)

- Table Offsets always uses 16 bytes. though sometimes the other 8 bytes will be an empty table, be creative when reading them.
- Remember to multiply the Table Counts by 32 bytes *(count*32)* as ToC uses 32 bytes, if 0x1500 has 3 counts of TOC. then multiply that to get 96 bytes (or 60 in hex interpretation). This will accurately pinpoint where the Table of `0x1500` ends. (use addition to check where the table ends, which **0x1500** ends at **0x1560**. basically: 0x1500+60=0x1560)

[Table of Contents] (ToC) [32 bytes in size]

~ File Offset: a little endian offset. brings you to a area where a offset chunk is stored.
~ Size/Compressed Size: defines the size of that area starting with File Offset. Size can vary if it's ZLIB compressed or not based on it's header (blz2)
~ Name Offset Table: a little endian offset. brings you to a area where the name is located. Which is a name table on how i call it.
~ Name Element: a name element (that's how i call it). can vary from 1 and 3. (can go above 5 on PSVITA SIDE)
~ Zeroes: bunch of zeroes, length of 12 (0xC)
~ Size (Decompressed): Game's expected decompressed size of that Compressed Size. Value can match with Size if it's not compressed by ZLIB.

[Identifying External File Data]
in 0x940. you see that the 4th byte has `40`. let's call this Address Types. If you encounter any of these values, it means that the file source is external. here's the following lists for these values:
~ 00 (0x0) = none
~ 30 (0x3) = Nosets (for PSvita, usually exists in folders `data_[value]`)
~ 40 (0x4) = package.rdp
~ 50 (0x5) = data.rdp
~ 60 (0x6) = patch.rdp
If you encounter any of these. You will need to do multiplications.

~ for **0xC**, it's a current/internal file. so it's inside of the **.res** file being read. as for **0xD**. it seem to be a **non-dlc** type of file, as the file already exists in between **data.rdp** or **package.rdp**, and has not been modified inside of **patch.rdp** and to be loaded when needed.

```
000007C0  10 12 00 C0 01 00 00 00 20 12 00 00 01 00 00 00  ...À.... .......
000007D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000007E0  40 12 00 C0 0F 00 00 00 80 12 00 00 01 00 00 00  @..À....€.......
000007F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000800  A0 12 00 C0 01 00 00 00 B0 12 00 00 01 00 00 00    ..À....°.......
00000810  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000820  D0 12 00 C0 01 00 00 00 E0 12 00 00 01 00 00 00  Ð..À....à.......
00000830  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000840  00 13 00 C0 03 00 00 00 10 13 00 00 01 00 00 00  ...À............
00000850  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000860  30 13 00 C0 08 00 00 00 50 13 00 00 01 00 00 00  0..À....P.......
00000870  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000880  80 13 00 C0 08 00 00 00 A0 13 00 00 01 00 00 00  €..À.... .......
00000890  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000008A0  D0 13 00 C0 08 00 00 00 F0 13 00 00 01 00 00 00  Ð..À....ð.......
000008B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000008C0  00 00 00 00 00 00 00 00 20 14 00 00 01 00 00 00  ........ .......
000008D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000008E0  40 14 00 C0 A8 00 00 00 F0 14 00 00 01 00 00 00  @..À¨...ð.......
000008F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000900  10 15 00 C0 12 00 00 00 30 15 00 00 01 00 00 00  ...À....0.......
00000910  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000920  50 15 00 C0 02 00 00 00 60 15 00 00 01 00 00 00  P..À....`.......
00000930  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000940  F5 78 05 40 E4 7E 19 00 70 15 00 00 03 00 00 00  õx.@ä~..p.......
00000950  00 00 00 00 00 00 00 00 00 00 00 00 E4 7E 19 00  ............ä~..
```

Obtaining Absolute Offset (for external files)
~ Reassign 0x940's File Offset into Big Endian order, you'll get something like this: "400578F5"/"0x400578F5".
~ Remove or Change "4" to "0" and multiply the new offset by 800: "400578F5" * "800" = "2BC7A800"/"0x2BC7A800"
~ The true/absolute offset is "2BC7A800"

***.rtbl* files have this structures but spreads out of nowhere with lengthy paddings keep an eye out.**

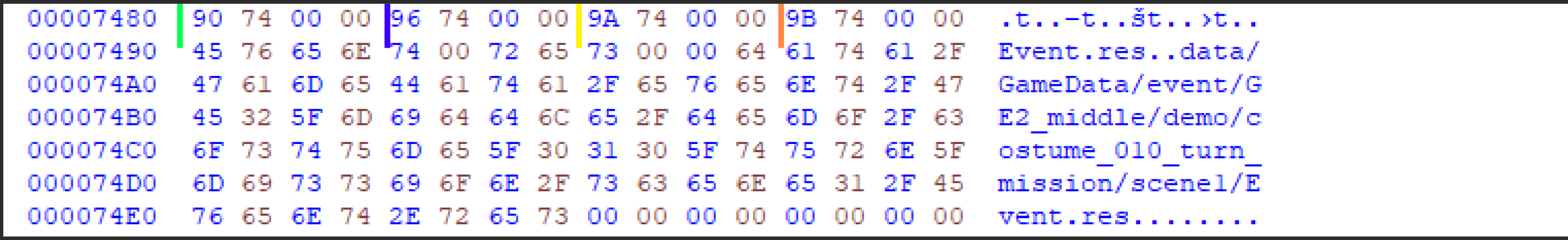
[Name Tables] (PSP SIDE)

Note: If Name Element is less than three (sometimes it could be 1). it will rely only on Name Offset. There's only **1**, **3,** and **4** value on the PSP side, so if its **4**, then there's **4** tables despite not having a number **2** value.

~ 1 = Name Offset: Little Endian offset, this is an offset that jumps you to the File Name.

~ 2 = Extension Offset: Little Endian offset. this offset jumps you to the File Extension.

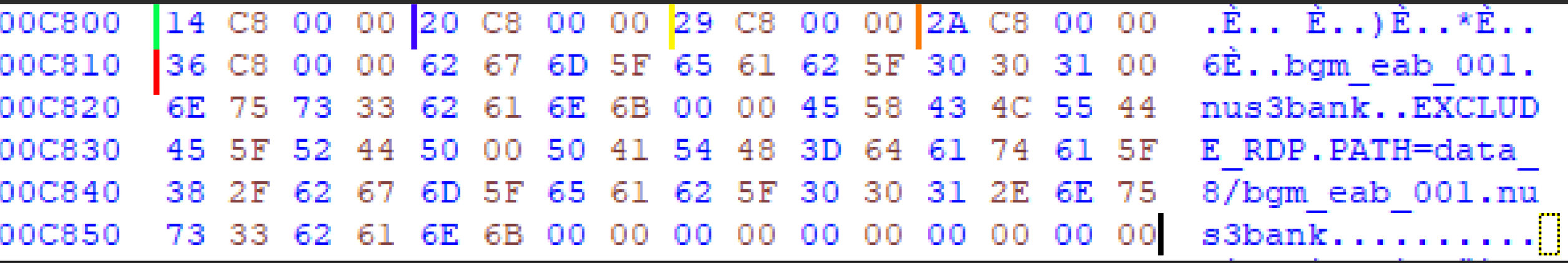
~ 3 and 4 = Path: an offset that jumps you to the zeroes. but if its non-zero, then it is a path (folder name). usually can form with `/` or not. 4th table Path usually shares the same value is 3rd table but with a +1 offset.



[Name Tables] (PSVITA SIDE)

Note: continuation of Name Tables (PSP Side). this is for the value **5**. value **3** is treated different here if value is above **4**.

~ 5 = Instruction Directory: an offset that jumps you to a instruction of that file's directory. A exclusion file that is outside of the RDP file



[ZLIB Compression] *BLZ2 only, i have no clue much on BLZ4*

0x00: Header 4 bytes. “blz2"

0x04: Compressed file size, 2 bytes.

the rest: compressed data.

Padding: adjustment for another file chunk or compressed data

[decompression sample code (python)]

```
def blz_decompress(data, csize, dsize):
    data = BytesIO(data)
    magic = data.read(4)
    if magic != b"blz2":
        raise ValueError("Data is not in BLZ2 format.")

    decom = b""
    if dsize >= 0xFFFF:
        size = int.from_bytes(data.read(2), "little")
        ekor = zlib.decompress(data.read(size), -15)
        while data.tell() < csize:
            size = int.from_bytes(data.read(2), "little")
            decom += zlib.decompress(data.read(size), -15)
        return decom + ekor
    else:
        size = int.from_bytes(data.read(2), "little")
        decom = zlib.decompress(data.read(size), -15)
        return decom
```