

The key assign of PSDIS 1.2

Key	Explanation
General ones	
Ctrl-N	New compilation
Ctrl-O	You open
Ctrl-S	Superscription retention
Ctrl-C	Copying the range of choice. (In text editor and the like paste possibility)
Ctrl-F	Search
F3	Search under
Shift-F3	Search on
PageDown	It moves under page 1
PageUp	It moves on page 1
UP	Cursor movement.
DOWN	Cursor movement
Dialogue indication	
Ctrl-A	“Setting dialogue indication of disassemble method of presentation” (with the menu it has become Ctrl-D.)
Ctrl-W	The disassembler window is opened.
Ctrl-D	The hexadecimal window is opened
Ctrl-H	Indicating header information
Something related to register	
Ctrl-R	Modification of register value
R	It executes selective line and makes register value reflect
Method of presentation of line	
Shift-B	Method of presentation of not yet decisive line is designated as .byte
Shift-H	Method of presentation of not yet decisive line is designated as .half
Shift-W	Method of presentation of not yet decisive line is designated as .word
Shift-C	Method of presentation of not yet decisive line is designated as code
B	Deciding the method of presentation of cursor line as .byte.
H	Deciding the method of presentation of cursor line as .half.

W	Deciding the method of presentation of cursor line as .word.
C	Deciding the method of presentation of cursor line as code.
U	Method of presentation of cursor line is made not yet decisive.
S	Method of presentation of cursor line becomes the semicolon. Something you are disgusted, the [ya]

The movement with address

G	Appointing address, it jumps.
F10	Appointing address, it jumps. (The same as G)
Ctrl-G	Appointing the label, it jumps

Mark

SPACE	Marking cursor line.
Ctrl-UP	It moves to the mark above
Ctrl-DOWN	It moves to the mark position under

Correction of line

RETURN	Correction of line. Setting cursor to Label.
Shift-RETURN	Correction of line. Setting cursor to Comment.

Stack

INSERT	Cursor line push to stack
DELETE	The pop it does from stack, throws away.
LEFT	The pop doing from stack, it moves.
RIGHT	To push cursor line in stack, jump to reference line.

Range appointment

Shift-DOWN	Cursor movement. Selecting the moving range.
Shift-UP	Cursor movement. Selecting the moving range.
Shift-PageDown	Moving under page 1, selecting the moving range.
Shift-PageUp	Moving on page 1, selecting the moving range.
Shift-RIGHT	It pushes cursor line in stack jump to reference line. Selecting the moving range.
Shift-LEFT	The pop doing from stack, it moves. Selecting the moving range.

Obscurity

Ctrl-V	??? Unused?
F6	??? Unused?
Shift-F6	??? Unused?

Shift-F1

??? Unused?

The bug which I verified

- The operand of sllv, srlv and srav (the [te] you say, is the [tsu] [ke] which is) order is strange.
- It draws up the disassembler window start address as 00000000 and when the search and search result there is no correspondence being, it moves to 00000000, after that falls.
- “Indication” of the menu - the accelerator “of setting of the disassemble” is indicated Ctrl-D, but really Ctrl-A.

2000-08-19

Format of the saving file of PSDIS 1.2

File header 16bytes

After +00 [shigunechiya] "PsDis100" +08 unused zero

x8 from here, the chunk (like this it calls) it is enumeration with chunk like something of GIF, but it has lined up one at a time into the order of HEADER CHUNK, IMAGE CHUNK, ATTRIBUTE CHUNK, LABEL CHUNK, COMMENT CHUNK and REGISTER CHUNK. When the similar chunk plural was appointed, the data before went out. Regrettable.

Chunk

+00 WORD type; ASCII 2 byte +02 DWORD size; Byte several +06 chunk specific data of chunk specific data

HEADER CHUNK

Chunk type: 50h 48h (PH: PS-X EXE HEADER?)

+000 PS-X EXE HEADER that way +800

IMAGE CHUNK

Chunk type: 4dh 49h (MI: Memory Image?)

+00 DWORD addr; The first address +04 DWORD size; Byte several +08
 BYTE [] img; Memory image

ATTRIBUTE CHUNK

Chunk type: 41h 54h (AT: ATtribute?)

+00 DWORD addr; The first address +04 DWORD size; Byte several +08
 BYTE [] flg; Arrangement this

of the flag explanation a little with necessary something just. As for the size of flg it is not size and the [te], size/it becomes 4. With you say, or (size + 3) perhaps /4. As for the flag 1 bytes are allotted word (4bytes) vis-a-vis as for addr or size memory image vis-a-vis that similarly, the scope of the data has been shown.

As for meaning of the flag bit 0-2 has displayed the attribute of the address.

00: undef 01: byte 02: half 03: word 04: The semicolon 05 of semantic
 obscurity:

code bit 3 when being comment in the place where it corresponds becomes with 1. (It is the case that the place where it corresponds are 4 places, but when it is somewhere, it is like the flag stands.)When searching using, you think that the [ru] it is.

Whether or not bit 4-7 is a label, is. As known it is somewhere of 4 places to, 4 bits using, it increases this. For example the flag displaying 0x80010000, in case of the [ru], when the flag is 0x39,

As for xxxx x 180,010,000 attribute with byte,

xxxx 1xxx comment from 80010000 being in 80010003,

0011 xxxx the label being set by 80010000 and 80010001, the [ru].

With it means to say.

LABEL CHUNK

Chunk type: 4ch 42h (LB: LaBel?)

+00 BYTE [8] sig; By all means "PSDISL00" +08 {+00 DOWRD address;
 Address +04 BYTE length; Byte several +05 CHAR in character string
 string [length]; The character string} continues even the [hi] and
 others.

From +08 as for block count it is understood at size of the chunk. You think that you call hash, or throw into CMap and the [ru] it is are.

COMMENT CHUNK

Chunk type: 43h 4dh (CM: CoMment?) Structure is the same as LABEL.

REGISTER CHUNK

Chunk type: 52h 56h (RV: Register Value?)

```
+00 DWORD r00; zero +04 DWORD r01; at... +7c DWORD r31; ra
```

it tried writing the value other than zeros on zero by the way, but it was ignored (laughing

[te] with reason experiment program

Reading the .pef file of psdis, selfish comment it tried making the program which is attached. When I analyze sometimes, to be, because also the [tsu] having worked in the same way to be the wasteful [tsu] [chi], it automated, rubbing, it does. Method of using basic,

```
prepsdis <in> <out>
```

Is. The pef file is appointed to in. If possible, please appoint the sufficient pef file which takes in memory image. It is the intention of trying to move even with the file of the compilation being completed, but there is no self-confidence. out is name of the pef file which is written out. With in and out as the same it operates, but it cannot have responsibility.

When it is caetla something, gp becoming 1f800000 fixing, it increases, but in such case

```
prepsdis <in> <out> gp=1f800000
```

When it does no lever, well enough nice comment is acquired.

In habit of the WIN32 console program DLL of mfc is necessary in operation. Furthermore, the extent [gero] it may spit the source, because the [tsu] [chi] to be, it hides, (laughing.

That when you say, what kind of is comes out,

```
8001d0d4 a4600000 LOOP> sh zero and the $0000 (v1) >spuCh0 vol-L
8001d0d8 a4600002 sh zero and the $0002 (v1) >spuCh0 vol-R 8001d0dc
a4660004 sh a2 and the $0004 (v1) >spuCh0 pitch 8001d0e0 a4650006 sh
a1 and the $0006 (v1) >spuCh0 wavAddr 8001d0e4 a4600008 sh zero and
the $0008 (v1) >spuCh0 attack, decay, sus lv 8001d0e8 a460000a sh
zero and $000a (v1) >spuCh0 sus, rel rate 8001d0ec 24840001 addiu a0,
a0, #$ 999,999,995,904 d0f 28,820,018 slti v0, a0, #$ 188,001 d0f
```

```
41,440 fff7 bne v0, zero and $8001d0d4: (LOOP>) -8Λ it becomes with  
8001d0f 824,630,016 addiu v1, v1, #$0010 spuCh1 vol-L 8001d0fc  
3411ffff ori s1, zero, #$ffff 0000ffff 8001d 1003 c028006 lui
```

v0 and the #8006 lever. V1 at a time 0x10 increasing with the loop, it is [ku] reason, but as for indication with the meeting present [tsu] lever please pardon ch0 [tsu] [te]. Even in addition we deliver in bug full load. However the [mu] [tsu] [chi] [ya] it is slow, please try

By the way because (empty [inchiki] it is the ill-smelling program,) however you read, you do not adequately treat the report of bug.

hanimar@geocities.com