

例外ライブラリ

Copyright © 2000 Sony Computer Entertainment Inc.
All Rights Reserved.
SCEI Confidential

目次

構造体	3
sceExcepIOPEexceptionData	4
関数	5
sceExcepConsOpen.....	6
sceExcepConsClose	7
sceExcepConsPrintf.....	8
sceExcepConsLocate	9
sceExcepSetDebugEEHandler	10
sceExcepSetDebugIOPHandler	11

構造体

sceExcepIOPEXceptionData

IOP 例外データ構造体

構 造

```
typedef struct {  
    int reg[45];  
    int version;  
    int offset;  
    char module[32];  
} sceExcepIOPEXceptionData __attribute__((aligned(16)));
```

メ ャ ン バ

<i>reg</i>	例外が発生したときの IOP のレジスタ
<i>version</i>	例外が発生したモジュールのバージョン
<i>offset</i>	例外が発生したモジュールのアドレスのオフセット
<i>module</i>	例外が発生したモジュール名[31 文字+NULL 文字]

解 説

IOP で例外が発生したときに、それに関する情報を EE 側で受け取るための構造体です。

関 連 項 目

sceExcepSetDebugIOPHandler()

関数

sceExcepConsOpen

コンソールのオープン

形 式

```
void sceExcepConsOpen(  
    u_int gs_x,  
    u_int gs_y,  
    u_int ch_w,  
    u_int ch_h  
)
```

引 数

<i>gs_x</i>	コンソールの描画 X 座標 (GS のプリミティブ座標)
<i>gs_y</i>	コンソールの描画 Y 座標 (GS のプリミティブ座標)
<i>ch_w</i>	コンソールの幅
<i>ch_h</i>	コンソールの高さ

返 り 値

なし

解 説

コンソールをオープンします。同時にオープンできるコンソールは1つだけで、メモリ確保などは行われません。

ch_w, *ch_h* で指定できる最大サイズは 80 × 40 キャラクタです。

sceExcepConsClose

コンソールのクローズ

形 式

void **sceExcepConsClose**(void)

引 数

なし

返 り 値

なし

解 説

sceExcepConsOpen() で開いたコンソールをクローズします。

sceExcepConsPrintf

コンソールへの文字出力

形 式

```
void sceExcepConsPrintf(  
    const char * str,  
    ...  
)
```

引 数

<i>str</i>	フォーマット指定文字列
...	出力パラメータ

返 り 値

なし

解 説

コンソールへ文字列を出力します。フォーマット指定と出力パラメータの記述方法は標準ライブラリの `printf` と同様です。表示できる ASCII コードは、0x20(スペース)から 0x7e'~'です。'\n' は改行し、その他の文字コードはスペースに置き換えて表示されます。

この関数を実行すると、PATH3 経由で GIF へ描画パケットを出力します。

sceExcepConsLocate

カーソル位置の変更

形 式

```
void sceExcepConsLocate(  
    u_int x,  
    u_int y  
)
```

引 数

x	カーソル X 座標 ($0 \leq x \leq ch_w-1$)
y	カーソル Y 座標 ($0 \leq y \leq ch_h-1$)

返 り 値

なし

解 説

コンソール上のカーソル位置を変更します。

引数 x がコンソールの右端の座標（オープン時に指定した ch_w-1 ）より大きいときは ch_w-1 に設定されます。

同様に、引数 y がコンソールの下端の座標（オープン時に指定した ch_h-1 ）より大きいときは ch_h-1 に設定されます。

sceExcepSetDebugEEHandler

EE 側例外ハンドラの登録

形 式

```
void sceExcepSetDebugEEHandler(  
    void (*handler)(u_int stat,  
    u_int cause,  
    u_int epc,  
    u_int bva,  
    u_int bpa,  
    u_long128 *gpr)  
)
```

引 数

<i>handler</i>	例外ハンドラ関数のポインタ
----------------	---------------

解 説

"PlayStation 2" debugging station および "PlayStation 2" Development Tool DTL-T10000 でのみ、*handler* で指定された関数を例外ハンドラとして登録します。"PlayStation 2"では何も行わずにリターンします。

"PlayStation 2" debugging station および "PlayStation 2" Development Tool DTL-T10000 で
の動作は、EE カーネルライブラリの SetDebugHandler と同じです。例外ハンドラの詳細につ
いては SetDebugHandler の解説を参照してください。

handler の引数は、例外発生時の状態を表しています。

引数 *stat* は COP0 の status レジスタ

引数 *cause* は COP0 の cause レジスタ

引数 *epc* は例外が発生したプログラムカウンタ値

引数 *bva* はメモリアクセスや分岐アドレス不正時の論理アドレス値

引数 *bpa* はバスエラー終了の物理アドレス値

引数 *gpr* はサイズが 32 個の配列で GPR の値です。

sceExcepSetDebugIOPHandler

IOP 側例外ハンドラの登録

形 式

```
void sceExcepSetDebugIOPHandler(  
    char *module,  
    sceSifCmdHandler IOExceptionHandler,  
    sceExcepIOExceptionData *pIOExceptionData  
)
```

引 数

<i>module</i>	panicsys.irx へのフルパス
<i>IOExceptionHandler</i>	IOP 例外発生時に呼ばれる EE のハンドラ関数へのポインタ
<i>pIOExceptionData</i>	IOP 例外データ構造体へのポインタ

返 り 値

なし

解 説

"PlayStation 2" debugging station および "PlayStation 2" Development Tool DTL-T10000 のみ、*module* で指定された IOP モジュールをロードし、*IOExceptionHandler* で指定された関数を IOP 例外のハンドラとして登録します。"PlayStation 2"では何も行わずにリターンします。

module には"host0:" / "cdrom0:"などを適切に補った panicsys.irx のフルパスを指定してください。panicsys.irx は IOP で発生した例外をトラップする IOP 側のモジュールで、このモジュールからの通知を受けて、EE カーネルは登録されている IOP 例外ハンドラを呼び出します。その際に panicsys.irx から通知される例外情報が *pIOExceptionData* で示されるアドレスに格納されます。

備 考

panicsys.irx のロードに失敗した場合、その旨のメッセージを繰り返し表示します。