SunOS リファレンスマニュアル 9 : DDI/DKI カーネル関数

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817–3842–10
2003 年 12 月

原典: *man pages section 9: DDI and DKI Kernel Functions*

Part No: 817-0702-10

Revision A

Adobe PostScript

030908@6671

目次

はじめに

概要

SunOS リファレンスマニュアルは、初めて SunOS を使用するユーザーやすでにある程度の知識を持っているユーザーのどちらでも対応できるように解説されています。このマニュアルを構成するマニュアルページは一般に参照マニュアルとして作られており、チュートリアルな要素は含んでいません。それぞれのコマンドを実行すると、どのような結果が得られるかについて、詳しく説明されています。なお、各マニュアルページの内容はオンラインでも参照することができます。

このマニュアルは、マニュアルページの内容によっていくつかのセクションに分かれています。各セクションについて以下に簡単に説明します。

- セクション 1 は、オペレーティングシステムで使えるコマンドを説明します。
- セクション 1M は、システム保守や管理用として主に使われるコマンドを説明します。
- セクション 2 は、すべてのシステムコールについて説明します。ほとんどのシステムコールに 1 つまたは複数のエラーがあります。エラーの場合、通常ありえない戻り値が返されます。
- セクション 3 は、さまざまなライブラリ中の関数について説明します。ただし、UNIX システムプリミティブを直接呼び出す関数については、セクション 2 で説明しています。
- セクション 4 は、各種ファイルの形式について説明します。また、ファイル形式を宣言する C 構造体を適用できる場合には随時説明しています。
- セクション 5 は、文字セットテーブルなど他のセクションには該当しないものについて説明します。
- セクション 7 は、特殊なハードウェア周辺装置またはデバイスドライバに関するさまざまな特殊ファイルについて説明します。STREAMS ソフトウェアドライバ、モジュール、またはシステムコールの STREAMS 汎用セットについても説明します。

- セクション 9 は、カーネル環境でデバイスドライバを記述するのに必要な参照情報を提供します。ここでは、デバイスドライバインタフェース (DDI) とドライバ/カーネルインタフェース (DKI) という 2 つのデバイスドライバインタフェース仕様について説明します。
- セクション 9F は、デバイスドライバが使用できるカーネル関数について説明します。

以下に、このマニュアルの項目を表記されている順に説明します。ほとんどのマニュアルページが下記の項目からなる共通の書式で書かれていますが、必要でない項目については省略されています。たとえば、記述すべきバグがコマンドにない場合などは、「使用上の留意点」という項目はありません。各マニュアルページの詳細は各セクションの `intro` を、マニュアルページの一般的な情報については `man`(1) を参照してください。

名前　コマンドや関数の名称と概略が示されています。

形式　コマンドや関数の構文が示されています。標準パスにコマンドやファイルが存在しない場合は、フルパス名が示されます。字体は、コマンド、オプションなどの定数にはボールド体 (bold) を、引数、パラメータ、置換文字などの変数にはイタリック体 (Italic) または <日本語訳> を使用しています。オプションと引数の順番は、アルファベット順です。特別な指定が必要な場合を除いて、1 文字の引数、引数のついたオプションの順に書かれています。

以下の文字がそれぞれの項目で使われています。

[ ]　このかっこに囲まれたオプションや引数は省略できます。このかっこが付いていない場合には、引数を必ず指定する必要があります。

. . .　省略符号。前の引数に変数を付けたり、引数を複数指定したりできることを意味します (例: ‘ `filename` . . .’)。

|　区切り文字 (セパレータ)。この文字で分割されている引数のうち 1 つだけを指定できます。

{ }　この大かっこに囲まれた複数のオプションや引数は省略できます。かっこ内を 1 組として扱います。

プロトコル　この項が使われているのは、プロトコルが記述されているファイルを示すサブセクション 3R だけです。パス名は常にボールド体 (bold) で示されています。

機能説明　コマンドの機能とその動作について説明します。実行時の詳細を説明していますが、オプションの説明や使用例はここでは示されていません。対話形式のコマンド、サブコマンド、リクエスト、マクロ、関数などに関しては「使用法」で説明します。

IOCTL　セクション 7 だけに使用される項です。`ioctl`(2) システムコールへのパラメータは `ioctl` と呼ばれ、適切なパラメータを持つデバイスクラスのマニュアルページだけに記載されています。特定の

デバイスに関する ioctl は、(そのデバイスのマニュアルページに) アルファベット順に記述されています。デバイスの特定のクラスに関する ioctl は、mtio(7I) のように io で終わる名前が付いているデバイスクラスのマニュアルページに記載されています。

オプション　各オプションがどのように実行されるかを説明しています。「形式」で示されている順に記述されています。オプションの引数はこの項目で説明され、必要な場合はデフォルト値を示します。

オペランド　コマンドのオペランドを一覧表示し、各オペランドがコマンドの動作にどのように影響を及ぼすかを説明しています。

出力　コマンドによって生成される出力 (標準出力、標準エラー、または出力ファイル) を説明しています。

戻り値　値を返す関数の場合、その値を示し、値が返される時の条件を説明しています。関数が 0 や -1 のような一定の値だけを返す場合は、値と説明の形で示され、その他の場合は各関数の戻り値について簡単に説明しています。void として宣言された関数はこの項では扱いません。

エラー　失敗の場合、ほとんどの関数はその理由を示すエラーコードを errno 変数の中に設定します。この項ではエラーコードをアルファベット順に記述し、各エラーの原因となる条件について説明します。同じエラーの原因となる条件が複数ある場合は、エラーコードの下にそれぞれの条件を別々のパラグラフで説明しています。

使用法　この項では、使用する際の手がかりとなる説明が示されています。特定の決まりや機能、詳しい説明の必要なコマンドなどが示されています。組み込み機能については、以下の小項目で説明しています。

コマンド
修飾子
変数
式
入力文法

使用例　コマンドや関数の使用例または使用方法を説明しています。できるだけ実際に入力するコマンド行とスクリーンに表示される内容を例にしています。例の中には必ず example% のプロンプトが出てきます。スーパーユーザーの場合は example# のプロンプトになります。例では、その説明、変数置換の方法、戻り値が示され、それらのほとんどが「形式」、「機能説明」、「オプション」、「使用法」の項からの実例となっています。

環境　コマンドや関数が影響を与える環境変数を記述し、その影響について簡単に説明しています。

終了ステータス	コマンドが呼び出しプログラムまたはシェルに返す値と、その状態を説明しています。通常、正常終了には 0 が返され、0 以外の値はそれぞれのエラー状態を示します。
ファイル	マニュアルページが参照するファイル、関連ファイル、およびコマンドが作成または必要とするファイルを示し、各ファイルについて簡単に説明しています。
属性	属性タイプとその対応する値を定義することにより、コマンド、ユーティリティ、およびデバイスドライバの特性を一覧しています。詳細は `attributes`(5) を参照してください。
関連項目	関連するマニュアルページ、当社のマニュアル、および一般の出版物が示されています。
診断	エラーの発生状況と診断メッセージが示されています。メッセージはボールド体 (bold) で、変数はイタリック体 (Italic) または <日本語訳> で示されており、C ロケール時の表示形式です。
警告	作業に支障を与えるような現象について説明しています。診断メッセージではありません。
注意事項	それぞれの項に該当しない追加情報が示されています。マニュアルページの内容とは直接関係のない事柄も参照用に扱っています。ここでは重要な情報については説明していません。
使用上の留意点	すでに発見されているバグについて説明しています。可能な場合は対処法も示しています。

SunOS リファレンスマニュアル 9 : DDI/DKI カーネル関数

名前 | Intro, intro – DDI/DKI 関数の序章

機能説明 | このセクションでは、デバイスドライバで使用可能なカーネル関数について説明します。デバイスドライバインタフェースの概要については、Intro(9E) を参照してください。

このセクションでは、各デバイスドライバ関数の情報を次の項目別に記載しています。

- 「名前」 - 関数の目的を簡単に示します。
- 「形式」 - ソースコードに含まれる関数のエントリポイントの構文を示します。必要なヘッダーには、#include 指示子を示します。
- 「インタフェースレベル」 - すべてのアーキテクチャ依存関係について説明します。
- 「引数」 - 関数を呼び出すのに必要なすべての引数について説明します。
- 「機能説明」 - 関数に関する概要情報を記載します。
- 「戻り値」 - 関数を呼び出した結果の戻り値とメッセージについて説明します。
- 「コンテキスト」 - 関数を呼び出すことのできるドライバコンテキスト (ユーザー、カーネル、割り込み、および高レベルな割り込み) を示します。
- ユーザースレッドにより直接呼び出された場合は、ドライバ関数にユーザーコンテキストがあります。ドライバの read(9E) エントリポイントを read(2) システムコールから呼び出した場合も、ユーザーコンテキストがあります。
- カーネルのほかの部分から呼び出された場合は、ドライバ関数にカーネルコンテキストがあります。ブロックデバイスドライバでは、デバイスにページを書き込むために、strategy(9E) エントリポイントがページデーモンによって呼び出される可能性があります。ページデーモンは現在のユーザースレッドとは関係がないので、この場合、strategy(9E) がカーネルコンテキストを持ちます。
- 割り込みコンテキスト はカーネルコンテキストですが、同時にそれに関連する割り込みレベルも持ちます。ドライバ割り込みルーチンは、割り込みコンテキストを持ちます。

  注 – 割り込みコンテキストでも入手できる mutex をユーザーコンテキストまたはカーネルコンテキストで入手した場合、この mutex を持つユーザーコンテキストスレッドまたはカーネルコンテキストスレッドは、この mutex を所有している限り、割り込みコンテキストによって強制されるすべての制限に影響されます。ドライバで mutex を適切に処理する方法について詳しくは、 mutex(9F) のマニュアルページを参照してください。
- 高レベルの割り込みコンテキストは、割り込みコンテキストの、より制限の厳しい形式です。ddi_intr_hilevel(9F) で、割り込みが高レベルの場合、ddi_add_intr(9F) を使ってその割り込みに追加されたドライバ割り込みルーチンは、高レベルの割り込みコンテキストで実行します。このような割り込みルーチンは、ddi_trigger_softintr(9F)、mutex_enter(9F) 、および mutex_exit(9F) を呼び出すことしかできません。さらに、mutex_enter(9F) と mutex_exit(9F) は、ddi_get_iblock_cookie(9F) で返された ddi_iblock_cookieで初期化される mutex 上でしか呼び出せない可能性があり

ます。

- 「関連項目」 - 「使用法」とソースから参照される関連する関数で、さらに詳細な情報を参照できます。
- 「使用例」 - ドライバコードで関数を使用する方法を示します。

すべてのドライバには、 <sys/ddi.h> および <sys/sunddi.h> が、ドライバに含まれる最後のファイルとしてこの順序で含まれていなければなりません。

STREAMS カーネル関数の概要

次の表は、このセクションで説明する STREAMS 関数を示します。

ルーチン	種類
adjmsg	DDI/DKI
allocb	DDI/DKI
backq	DDI/DKI
bcanput	DDI/DKI
bcanputnext	DDI/DKI
bufcall	DDI/DKI
canput	DDI/DKI
canputnext	DDI/DKI
clrbuf	DDI/DKI
copyb	DDI/DKI
copymsg	DDI/DKI
datamsg	DDI/DKI
dupb	DDI/DKI
dupmsg	DDI/DKI
enableok	DDI/DKI
esballoc	DDI/DKI
esbbcall	DDI/DKI
flushband	DDI/DKI
flushq	DDI/DKI
freeb	DDI/DKI
freemsg	DDI/DKI
freezestr	DDI/DKI

ルーチン	種類
getq	DDI/DKI
insq	DDI/DKI
linkb	DDI/DKI
msgdsize	DDI/DKI
msgpullup	DDI/DKI
mt-streams	Solaris DDI
noenable	DDI/DKI
OTHERQ	DDI/DKI
pullupmsg	DDI/DKI
put	DDI/DKI
putbq	DDI/DKI
putctl	DDI/DKI
putctl1	DDI/DKI
putnext	DDI/DKI
putnextctl	DDI/DKI
putq	DDI/DKI
qbufcall	Solaris DDI
qenable	DDI/DKI
qprocson	DDI/DKI
qprocsoff	DDI/DKI
qreply	DDI/DKI
qsize	DDI/DKI
qtimeout	Solaris DDI
qunbufcall	Solaris DDI
quntimeout	Solaris DDI
qwait	Solaris DDI
qwait_sig	Solaris DDI
qwriter	Solaris DDI
RD	DDI/DKI

ルーチン	種類
rmvb	DDI/DKI
rmvq	DDI/DKI
SAMESTR	DDI/DKI
strlog	DDI/DKI
strqget	DDI/DKI
strqset	DDI/DKI
testb	DDI/DKI
unbufcall	DDI/DKI
unfreezestr	DDI/DKI
unlinkb	DDI/DKI
WR	DDI/DKI

次の表は、STREAMS 固有ではない関数を示します。

ルーチン	種類
ASSERT	DDI/DKI
anocancel	Solaris DDI
aphysio	Solaris DDI
bcmp	DDI/DKI
bcopy	DDI/DKI
biodone	DDI/DKI
bioclone	Solaris DDI
biofini	Solaris DDI
bioinit	Solaris DDI
biomodified	Solaris DDI
biosize	Solaris DDI
bioerror	Solaris DDI
bioreset	Solaris DDI
biowait	DDI/DKI
bp_mapin	DDI/DKI

ルーチン	種類
bp_mapout	DDI/DKI
btop	DDI/DKI
btopr	DDI/DKI
bzero	DDI/DKI
cmn_err	DDI/DKI
copyin	DDI/DKI
copyout	DDI/DKI
cv_broadcast	Solaris DDI
cv_destroy	Solaris DDI
cv_init	Solaris DDI
cv_signal	Solaris DDI
cv_timedwait	Solaris DDI
cv_wait	Solaris DDI
cv_wait_sig	Solaris DDI
ddi_add_intr	Solaris DDI
ddi_add_softintr	Solaris DDI
ddi_btop	Solaris DDI
ddi_btopr	Solaris DDI
ddi_copyin	Solaris DDI
ddi_copyout	Solaris DDI
ddi_create_minor_node	Solaris DDI
ddi_dev_is_sid	Solaris DDI
ddi_dev_nintrs	Solaris DDI
ddi_dev_nregs	Solaris DDI
ddi_dev_regsize	Solaris DDI
ddi_device_copy	Solaris DDI
ddi_device_zero	Solaris DDI
ddi_devmap_segmap	Solaris DDI
ddi_dma_addr_bind_handle	Solaris DDI

ルーチン	種類
ddi_dma_addr_setup	Solaris DDI
ddi_dma_alloc_handle	Solaris DDI
ddi_dma_buf_bind_handle	Solaris DDI
ddi_dma_buf_setup	Solaris DDI
ddi_dma_burstsizes	Solaris DDI
ddi_dma_coff	Solaris SPARC DDI
ddi_dma_curwin	Solaris SPARC DDI
ddi_dma_devalign	Solaris DDI
ddi_dma_free	Solaris DDI
ddi_dma_free_handle	Solaris DDI
ddi_dma_getwin	Solaris DDI
ddi_dma_htoc	Solaris SPARC DDI
ddi_dma_mem_alloc	Solaris DDI
ddi_dma_mem_free	Solaris DDI
ddi_dma_movwin	Solaris SPARC DDI
ddi_dma_nextcookie	Solaris DDI
ddi_dma_nextseg	Solaris DDI
ddi_dma_nextwin	Solaris DDI
ddi_dma_numwin	Solaris DDI
ddi_dma_segtocookie	Solaris DDI
ddi_dma_set_sbus64	Solaris DDI
ddi_dma_setup	Solaris DDI
ddi_dma_sync	Solaris DDI
ddi_dma_unbind_handle	Solaris DDI
ddi_dmae	Solaris x86 DDI
ddi_dmae_1stparty	Solaris x86 DDI
ddi_dmae_alloc	Solaris x86 DDI
ddi_dmae_disable	Solaris x86 DDI
ddi_dmae_enable	Solaris x86 DDI

ルーチン	種類
ddi_dmae_getattr	Solaris x86 DDI
ddi_dmae_getcnt	Solaris x86 DDI
ddi_dmae_getlim	Solaris x86 DDI
ddi_dmae_prog	Solaris x86 DDI
ddi_dmae_release	Solaris x86 DDI
ddi_dmae_stop	Solaris x86 DDI
ddi_enter_critical	Solaris DDI
ddi_exit_critical	Solaris DDI
ddi_ffs	Solaris DDI
ddi_fls	Solaris DDI
ddi_get16	Solaris DDI
ddi_get32	Solaris DDI
ddi_get64	Solaris DDI
ddi_get8	Solaris DDI
ddi_get_cred	Solaris DDI
ddi_get_driver_private	Solaris DDI
ddi_get_iblock_cookie	Solaris DDI
ddi_get_instance	Solaris DDI
ddi_get_name	Solaris DDI
ddi_get_parent	Solaris DDI
ddi_get_soft_iblock_cookie	Solaris DDI
ddi_get_soft_state	Solaris DDI
ddi_getb	Solaris DDI
ddi_getl	Solaris DDI
ddi_getll	Solaris DDI
ddi_getlongprop	Solaris DDI
ddi_getlongprop_buf	Solaris DDI
ddi_getprop	Solaris DDI
ddi_getproplen	Solaris DDI

ルーチン	種類
ddi_getw	Solaris DDI
ddi_intr_hilevel	Solaris DDI
ddi_io_get16	Solaris DDI
ddi_io_get32	Solaris DDI
ddi_io_get8	Solaris DDI
ddi_io_getb	Solaris DDI
ddi_io_getl	Solaris DDI
ddi_io_getw	Solaris DDI
ddi_io_put16	Solaris DDI
ddi_io_put32	Solaris DDI
ddi_io_put8	Solaris DDI
ddi_io_putb	Solaris DDI
ddi_io_putl	Solaris DDI
ddi_io_putw	Solaris DDI
ddi_io_rep_get16	Solaris DDI
ddi_io_rep_get32	Solaris DDI
ddi_io_rep_get8	Solaris DDI
ddi_io_rep_getb	Solaris DDI
ddi_io_rep_getl	Solaris DDI
ddi_io_rep_getw	Solaris DDI
ddi_io_rep_put16	Solaris DDI
ddi_io_rep_put32	Solaris DDI
ddi_io_rep_put8	Solaris DDI
ddi_io_rep_putb	Solaris DDI
ddi_io_rep_putl	Solaris DDI
ddi_io_rep_putw	Solaris DDI
ddi_iomin	Solaris DDI
ddi_iopb_alloc	Solaris DDI
ddi_iopb_free	Solaris DDI

ルーチン	種類
ddi_map_regs	Solaris DDI
ddi_mapdev	Solaris DDI
ddi_mapdev_intercept	Solaris DDI
ddi_mapdev_nointercept	Solaris DDI
ddi_mapdev_set_device_acc_attr	Solaris DDI
ddi_mem_alloc	Solaris DDI
ddi_mem_free	Solaris DDI
ddi_mem_get16	Solaris DDI
ddi_mem_get32	Solaris DDI
ddi_mem_get64	Solaris DDI
ddi_mem_get8	Solaris DDI
ddi_mem_getb	Solaris DDI
ddi_mem_getl	Solaris DDI
ddi_mem_getll	Solaris DDI
ddi_mem_getw	Solaris DDI
ddi_mem_put16	Solaris DDI
ddi_mem_put32	Solaris DDI
ddi_mem_put64	Solaris DDI
ddi_mem_put8	Solaris DDI
ddi_mem_putb	Solaris DDI
ddi_mem_putl	Solaris DDI
ddi_mem_putll	Solaris DDI
ddi_mem_putw	Solaris DDI
ddi_mem_rep_get16	Solaris DDI
ddi_mem_rep_get32	Solaris DDI
ddi_mem_rep_get64	Solaris DDI
ddi_mem_rep_get8	Solaris DDI
ddi_mem_rep_getb	Solaris DDI
ddi_mem_rep_getl	Solaris DDI

ルーチン	種類
ddi_mem_rep_getll	Solaris DDI
ddi_mem_rep_getw	Solaris DDI
ddi_mem_rep_put16	Solaris DDI
ddi_mem_rep_put32	Solaris DDI
ddi_mem_rep_put64	Solaris DDI
ddi_mem_rep_put8	Solaris DDI
ddi_mem_rep_putb	Solaris DDI
ddi_mem_rep_putl	Solaris DDI
ddi_mem_rep_putll	Solaris DDI
ddi_mem_rep_putw	Solaris DDI
ddi_mmap_get_model	Solaris DDI
ddi_model_convert_from	Solaris DDI
ddi_node_name	Solaris DDI
ddi_peek16	Solaris DDI
ddi_peek32	Solaris DDI
ddi_peek64	Solaris DDI
ddi_peek8	Solaris DDI
ddi_peekc	Solaris DDI
ddi_peekd	Solaris DDI
ddi_peekl	Solaris DDI
ddi_peeks	Solaris DDI
ddi_poke16	Solaris DDI
ddi_poke32	Solaris DDI
ddi_poke64	Solaris DDI
ddi_poke8	Solaris DDI
ddi_pokec	Solaris DDI
ddi_poked	Solaris DDI
ddi_pokel	Solaris DDI
ddi_pokes	Solaris DDI

ルーチン	種類
ddi_prop_create	Solaris DDI
ddi_prop_exists	Solaris DDI
ddi_prop_free	Solaris DDI
ddi_prop_get_int	Solaris DDI
ddi_prop_lookup	Solaris DDI
ddi_prop_lookup_byte_array	Solaris DDI
ddi_prop_lookup_int_array	Solaris DDI
ddi_prop_lookup_string	Solaris DDI
ddi_prop_lookup_string_array	Solaris DDI
ddi_prop_modify	Solaris DDI
ddi_prop_op	Solaris DDI
ddi_prop_remove	Solaris DDI
ddi_prop_remove_all	Solaris DDI
ddi_prop_undefine	Solaris DDI
ddi_prop_update	Solaris DDI
ddi_prop_update_byte_array	Solaris DDI
ddi_prop_update_int	Solaris DDI
ddi_prop_update_int_array	Solaris DDI
ddi_prop_update_string	Solaris DDI
ddi_prop_update_string_array	Solaris DDI
ddi_ptob	Solaris DDI
ddi_put16	Solaris DDI
ddi_put32	Solaris DDI
ddi_put64	Solaris DDI
ddi_put8	Solaris DDI
ddi_putb	Solaris DDI
ddi_putl	Solaris DDI
ddi_putll	Solaris DDI
ddi_putw	Solaris DDI

ルーチン	種類
ddi_regs_map_free	Solaris DDI
ddi_regs_map_setup	Solaris DDI
ddi_remove_intr	Solaris DDI
ddi_remove_minor_node	Solaris DDI
ddi_remove_softintr	Solaris DDI
ddi_rep_get16	Solaris DDI
ddi_rep_get32	Solaris DDI
ddi_rep_get64	Solaris DDI
ddi_rep_get8	Solaris DDI
ddi_rep_getb	Solaris DDI
ddi_rep_getl	Solaris DDI
ddi_rep_getll	Solaris DDI
ddi_rep_getw	Solaris DDI
ddi_rep_put16	Solaris DDI
ddi_rep_put32	Solaris DDI
ddi_rep_put64	Solaris DDI
ddi_rep_put8	Solaris DDI
ddi_rep_putb	Solaris DDI
ddi_rep_putl	Solaris DDI
ddi_rep_putll	Solaris DDI
ddi_rep_putw	Solaris DDI
ddi_report_dev	Solaris DDI
ddi_root_node	Solaris DDI
ddi_segmap	Solaris DDI
ddi_segmap_setup	Solaris DDI
ddi_set_driver_private	Solaris DDI
ddi_slaveonly	Solaris DDI
ddi_soft_state	Solaris DDI
ddi_soft_state_fini	Solaris DDI

ルーチン	種類
ddi_soft_state_free	Solaris DDI
ddi_soft_state_init	Solaris DDI
ddi_soft_state_zalloc	Solaris DDI
ddi_trigger_softintr	Solaris DDI
ddi_umem_alloc	Solaris DDI
ddi_umem_free	Solaris DDI
ddi_unmap_regs	Solaris DDI
delay	DDI/DKI
devmap_default_access	Solaris DDI
devmap_devmem_setup	Solaris DDI
devmap_do_ctxmgt	Solaris DDI
devmap_load	Solaris DDI
devmap_set_ctx_timeout	Solaris DDI
devmap_setup	Solaris DDI
devmap_umem_setup	Solaris DDI
devmap_unload	Solaris DDI
disksort	Solaris DDI
drv_getparm	DDI/DKI
drv_hztousec	DDI/DKI
drv_priv	DDI/DKI
drv_usectohz	DDI/DKI
drv_usecwait	DDI/DKI
free_pktiopb	Solaris DDI
freerbuf	DDI/DKI
get_pktiopb	Solaris DDI
geterror	DDI/DKI
getmajor	DDI/DKI
getminor	DDI/DKI
getrbuf	DDI/DKI

ルーチン	種類
hat_getkpfnum	DKI のみ
inb	Solaris x86 DDI
inl	Solaris x86 DDI
inw	Solaris x86 DDI
kmem_alloc	DDI/DKI
kmem_free	DDI/DKI
kmem_zalloc	DDI/DKI
kstat_create	Solaris DDI
kstat_delete	Solaris DDI
kstat_install	Solaris DDI
kstat_named_init	Solaris DDI
kstat_queue	Solaris DDI
kstat_runq_back_to_waitq	Solaris DDI
kstat_runq_enter	Solaris DDI
kstat_runq_exit	Solaris DDI
kstat_waitq_enter	Solaris DDI
kstat_waitq_exit	Solaris DDI
kstat_waitq_to_runq	Solaris DDI
makecom_g0	Solaris DDI
makecom_g0_s	Solaris DDI
makecom_g1	Solaris DDI
makecom_g5	Solaris DDI
makedevice	DDI/DKI
max	DDI/DKI
min	DDI/DKI
minphys	Solaris DDI
mod_info	Solaris DDI
mod_install	Solaris DDI
mod_remove	Solaris DDI

ルーチン	種類
mutex_destroy	Solaris DDI
mutex_enter	Solaris DDI
mutex_exit	Solaris DDI
mutex_init	Solaris DDI
mutex_owned	Solaris DDI
mutex_tryenter	Solaris DDI
nochpoll	Solaris DDI
nodev	DDI/DKI
nulldev	DDI/DKI
numtos	Solaris DDI
outb	Solaris x86 DDI
outl	Solaris x86 DDI
outw	Solaris x86 DDI
pci_config_get16	Solaris DDI
pci_config_get32	Solaris DDI
pci_config_get64	Solaris DDI
pci_config_get8	Solaris DDI
pci_config_getb	Solaris DDI
pci_config_getl	Solaris DDI
pci_config_getw	Solaris DDI
pci_config_put16	Solaris DDI
pci_config_put32	Solaris DDI
pci_config_put64	Solaris DDI
pci_config_put8	Solaris DDI
pci_config_putb	Solaris DDI
pci_config_putl	Solaris DDI
pci_config_putw	Solaris DDI
pci_config_setup	Solaris DDI
pci_config_teardown	Solaris DDI

ルーチン	種類
physio	Solaris DDI
pollwakeup	DDI/DKI
proc_ref	Solaris DDI
proc_signal	Solaris DDI
proc_unref	Solaris DDI
ptob	DDI/DKI
repinsb	Solaris x86 DDI
repinsd	Solaris x86 DDI
repinsw	Solaris x86 DDI
repoutsb	Solaris x86 DDI
repoutsd	Solaris x86 DDI
repoutsw	Solaris x86 DDI
rmalloc	DDI/DKI
rmalloc_wait	DDI/DKI
rmallocmap	DDI/DKI
rmallocmap_wait	DDI/DKI
rmfree	DDI/DKI
rmfreemap	DDI/DKI
rw_destroy	Solaris DDI
rw_downgrade	Solaris DDI
rw_enter	Solaris DDI
rw_exit	Solaris DDI
rw_init	Solaris DDI
rw_read_locked	Solaris DDI
rw_tryenter	Solaris DDI
rw_tryupgrade	Solaris DDI
scsi_abort	Solaris DDI
scsi_alloc_consistent_buf	Solaris DDI
scsi_cname	Solaris DDI

ルーチン	種類
scsi_destroy_pkt	Solaris DDI
scsi_dmafree	Solaris DDI
scsi_dmaget	Solaris DDI
scsi_dname	Solaris DDI
scsi_errmsg	Solaris DDI
scsi_free_consistent_buf	Solaris DDI
scsi_hba_attach	Solaris DDI
scsi_hba_attach_setup	Solaris DDI
scsi_hba_detach	Solaris DDI
scsi_hba_fini	Solaris DDI
scsi_hba_init	Solaris DDI
scsi_hba_lookup_capstr	Solaris DDI
scsi_hba_pkt_alloc	Solaris DDI
scsi_hba_pkt_free	Solaris DDI
scsi_hba_probe	Solaris DDI
scsi_hba_tran_alloc	Solaris DDI
scsi_hba_tran_free	Solaris DDI
scsi_ifgetcap	Solaris DDI
scsi_ifsetcap	Solaris DDI
scsi_init_pkt	Solaris DDI
scsi_log	Solaris DDI
scsi_mname	Solaris DDI
scsi_pktalloc	Solaris DDI
scsi_pktfree	Solaris DDI
scsi_poll	Solaris DDI
scsi_probe	Solaris DDI
scsi_resalloc	Solaris DDI
scsi_reset	Solaris DDI
scsi_reset_notify	Solaris DDI

ルーチン	種類
scsi_resfree	Solaris DDI
scsi_rname	Solaris DDI
scsi_slave	Solaris DDI
scsi_sname	Solaris DDI
scsi_sync_pkt	Solaris DDI
scsi_transport	Solaris DDI
scsi_unprobe	Solaris DDI
scsi_unslave	Solaris DDI
sema_destroy	Solaris DDI
sema_init	Solaris DDI
sema_p	Solaris DDI
sema_p_sig	Solaris DDI
sema_tryp	Solaris DDI
sema_v	Solaris DDI
sprintf	Solaris DDI
stoi	Solaris DDI
strchr	Solaris DDI
strcmp	Solaris DDI
strcpy	Solaris DDI
strlen	Solaris DDI
strncmp	Solaris DDI
strncpy	Solaris DDI
swab	DDI/DKI
timeout	DDI/DKI
uiomove	DDI/DKI
untimeout	DDI/DKI
ureadc	DDI/DKI
uwritec	DDI/DKI
va_arg	Solaris DDI

ルーチン	種類
va_end	Solaris DDI
va_start	Solaris DDI
vcmn_err	DDI/DKI
vsprintf	Solaris DDI

関連項目　Intro(9E)、mutex(9F)

関数の一覧　名前
　説明

ASSERT(9F)
　式を検証する

IOC_CONVERT_FROM(9F)
　M_IOCTL の内容を変換する必要があるかどうか決定する

OTHERQ(9F)
　キューのパートナーキューへのポインタを入手する

RD(9F)
　読み取りキューへのポインタを入手する

SAMESTR(9F)
　次のキューが同じストリームに存在するかどうかテストする

SIZEOF_PTR(9F)
　STRUCT_DECL(9F) を参照

SIZEOF_STRUCT(9F)
　STRUCT_DECL(9F) を参照

STRUCT_BUF(9F)
　STRUCT_DECL(9F) を参照

STRUCT_DECL(9F)
　32 ビットアプリケーションのデータアクセスマクロ

STRUCT_FADDR(9F)
　STRUCT_DECL(9F) を参照

STRUCT_FGET(9F)
　STRUCT_DECL(9F) を参照

STRUCT_FGETP(9F)
　STRUCT_DECL(9F) を参照

STRUCT_FSET(9F)
　STRUCT_DECL(9F) を参照

STRUCT_FSETP(9F)
　STRUCT_DECL(9F) を参照

STRUCT_HANDLE(9F)
  STRUCT_DECL(9F) を参照

STRUCT_INIT(9F)
  STRUCT_DECL(9F) を参照

STRUCT_SET_HANDLE(9F)
  STRUCT_DECL(9F) を参照

STRUCT_SIZE(9F)
  STRUCT_DECL(9F) を参照

WR(9F)
  このモジュールまたはドライバの書き込みキューへのポインタを入手する

adjmsg(9F)
  メッセージからバイトをトリミングする

allocb(9F)
  メッセージブロックを割り当てる

anocancel(9F)
  非同期の入出力要求の取り消しを防ぐ

aphysio(9F)
  非同期の物理入出力を実行する

assert(9F)
  ASSERT(9F) を参照

backq(9F)
  現在のキューの後に控えているキューへのポインタを入手する

bcanput(9F)
  指定した優先順位の帯域でフロー制御をテストする

bcanputnext(9F)
  canputnext(9F) を参照

bcmp(9F)
  2 つのバイト配列を比較する

bcopy(9F)
  カーネル中のアドレス位置間でデータをコピーする

bioclone(9F)
  ほかのバッファーを複製する

biodone(9F)
  バッファーの入出力転送の後でバッファーを解放し、ブロックされたスレッドに通知する

bioerror(9F)
  バッファーのヘッダー内のエラーを示す

biofini(9F)
バッファー構造の初期化を解除する

bioinit(9F)
バッファー構造を初期化する

biomodified(9F)
バッファーが修正されたか検査する

bioreset(9F)
入出力が完了した後で、プライベートバッファーのヘッダーを再利用する

biosize(9F)
バッファー構造のサイズを戻す

biowait(9F)
ブロック入出力の完了を保留しているプロセスを中断する

bp_mapin(9F)
仮想アドレス空間を割り当てる

bp_mapout(9F)
仮想アドレス空間の割り当てを解除する

btop(9F)
バイト単位のサイズをページ単位のサイズに変換する (下位に丸める)

btopr(9F)
バイト単位のサイズをページ単位のサイズに変換する (上位に丸める)

bufcall(9F)
バッファーが使用可能になったら、関数を呼び出す

bzero(9F)
指定されたバイト数の分だけメモリーをクリアする

canput(9F)
メッセージキューに空きがあるかテストする

canputnext(9F)
次のモジュールのメッセージキューに空きがあるかテストする

clrbuf(9F)
バッファーの内容を消去する

cmn_err(9F)
エラーメッセージを表示するか、またはシステムを混乱させる

condvar(9F)
変数のルーチンを調整する

copyb(9F)
メッセージブロックをコピーする

copyin(9F)
ユーザープログラムからドライバのバッファーにデータをコピーする

copymsg(9F)
　メッセージをコピーする

copyout(9F)
　ドライバからユーザープログラムにデータをコピーする

csx_AccessConfigurationRegister(9F)
　PC カード構成レジスタを読み取る、または書き込む

csx_CS_DDI_Info(9F)
　DDI 情報を取得する

csx_ConvertSize(9F)
　デバイスのサイズを変換する

csx_ConvertSpeed(9F)
　デバイスの速度を変換する

csx_DeregisterClient(9F)
　カードサービスリストからクライアントを削除する

csx_DupHandle(9F)
　アクセスハンドルを複製する

csx_Error2Text(9F)
　エラーの戻りコードをテキスト文字列に変換する

csx_Event2Text(9F)
　イベントをテキスト文字列に変換する

csx_FreeHandle(9F)
　アクセスハンドルを解放する

csx_Get16(9F)
　csx_Get8(9F) を参照

csx_Get32(9F)
　csx_Get8(9F) を参照

csx_Get64(9F)
　csx_Get8(9F) を参照

csx_Get8(9F)
　デバイスアドレスからデータを読み取る

csx_GetEventMask(9F)
　csx_SetEventMask(9F) を参照

csx_GetFirstClient(9F)
　最初のクライアントまたは次のクライアントを戻す

csx_GetFirstTuple(9F)
　カード情報構造のタプルを戻す

csx_GetHandleOffset(9F)
　現在のアクセスハンドルオフセットを戻す

csx_GetMappedAddr(9F)
マップされた仮想アドレスを戻す

csx_GetNextClient(9F)
csx_GetFirstClient(9F) を参照

csx_GetNextTuple(9F)
csx_GetFirstTuple(9F) を参照

csx_GetStatus(9F)
PC カードおよびそのソケットの現在の状態を戻す

csx_GetTupleData(9F)
タプルのデータ部分を戻す

csx_MakeDeviceNode(9F)
クライアントにかわって、マイナーノードを作成および削除する

csx_MapLogSocket(9F)
クライアントハンドルに関連した物理ソケット番号を戻す

csx_MapMemPage(9F)
PC カード上のメモリー領域をマッピングする

csx_ModifyConfiguration(9F)
ソケットおよび PC カード構成レジスタを修正する

csx_ModifyWindow(9F)
ウィンドウ属性を修正する

csx_ParseTuple(9F)
汎用タプル構文解析プログラム

csx_Parse_CISTPL_BATTERY(9F)
バッテリー交換日付タプルを構文解析する

csx_Parse_CISTPL_BYTEORDER(9F)
バイト順序タプルを構文解析する

csx_Parse_CISTPL_CFTABLE_ENTRY(9F)
16 ビットカード構成表エントリタプルを構文解析する

csx_Parse_CISTPL_CONFIG(9F)
構成タプルを構文解析する

csx_Parse_CISTPL_DATE(9F)
カード初期化日付タプルを構文解析する

csx_Parse_CISTPL_DEVICE(9F)
デバイス情報タプルを構文解析する

csx_Parse_CISTPL_DEVICEGEO(9F)
デバイスジオメトリタプルを構文解析する

csx_Parse_CISTPL_DEVICEGEO_A(9F)
デバイスジオメトリ A タプルを構文解析する

csx_Parse_CISTPL_DEVICE_A(9F)
  csx_Parse_CISTPL_DEVICE(9F) を参照

csx_Parse_CISTPL_DEVICE_OA(9F)
  csx_Parse_CISTPL_DEVICE(9F) を参照

csx_Parse_CISTPL_DEVICE_OC(9F)
  csx_Parse_CISTPL_DEVICE(9F) を参照

csx_Parse_CISTPL_FORMAT(9F)
  データ記録書式タプルを構文解析する

csx_Parse_CISTPL_FUNCE(9F)
  関数拡張タプルを構文解析する

csx_Parse_CISTPL_FUNCID(9F)
  関数識別タプルを構文解析する

csx_Parse_CISTPL_GEOMETRY(9F)
  ジオメトリタプルを構文解析する

csx_Parse_CISTPL_JEDEC_A(9F)
  csx_Parse_CISTPL_JEDEC_C(9F) を参照

csx_Parse_CISTPL_JEDEC_C(9F)
  JEDEC 識別子タプルを構文解析する

csx_Parse_CISTPL_LINKTARGET(9F)
  リンクターゲットタプルを構文解析する

csx_Parse_CISTPL_LONGLINK_A(9F)
  Long Link A および Long Link C タプルを構文解析する

csx_Parse_CISTPL_LONGLINK_C(9F)
  csx_Parse_CISTPL_LONGLINK_A(9F) を参照

csx_Parse_CISTPL_LONGLINK_MFC(9F)
  マルチ関数タプルを構文解析する

csx_Parse_CISTPL_MANFID(9F)
  製造元識別子タプルを構文解析する

csx_Parse_CISTPL_ORG(9F)
  データ組織タプルを構文解析する

csx_Parse_CISTPL_SPCL(9F)
  特定の目的のタプルを構文解析する

csx_Parse_CISTPL_SWIL(9F)
  ソフトウェアインタリーブタプルを構文解析する

csx_Parse_CISTPL_VERS_1(9F)
  レベル 1 バージョンまたは製品情報タプルを構文解析する

csx_Parse_CISTPL_VERS_2(9F)
  レベル 2 バージョンまたは製品情報タプルを構文解析する

csx_Put16(9F)
  csx_Put8(9F) を参照

csx_Put32(9F)
  csx_Put8(9F) を参照

csx_Put64(9F)
  csx_Put8(9F) を参照

csx_Put8(9F)
  デバイスレジスタに書き込む

csx_RegisterClient(9F)
  クライアントを登録する

csx_ReleaseConfiguration(9F)
  PC カード構成およびソケット構成を解放する

csx_ReleaseIO(9F)
  csx_RequestIO(9F) を参照

csx_ReleaseIRQ(9F)
  csx_RequestIRQ(9F) を参照

csx_ReleaseSocketMask(9F)
  csx_RequestSocketMask(9F) を参照

csx_ReleaseWindow(9F)
  csx_RequestWindow(9F) を参照

csx_RemoveDeviceNode(9F)
  csx_MakeDeviceNode(9F) を参照

csx_RepGet16(9F)
  csx_RepGet8(9F) を参照

csx_RepGet32(9F)
  csx_RepGet8(9F) を参照

csx_RepGet64(9F)
  csx_RepGet8(9F) を参照

csx_RepGet8(9F)
  デバイスレジスタから繰り返し読み取る

csx_RepPut16(9F)
  csx_RepPut8(9F) を参照

csx_RepPut32(9F)
  csx_RepPut8(9F) を参照

csx_RepPut64(9F)
  csx_RepPut8(9F) を参照

csx_RepPut8(9F)
  デバイスレジスタに繰り返し書き込む

csx_RequestConfiguration(9F)
PC カードおよびソケットを構成する

csx_RequestIO(9F)
クライアント用の入出力リソースを要求する、または解放する

csx_RequestIRQ(9F)
IRQ リソースを要求する、または解放する

csx_RequestSocketMask(9F)
クライアントのクライアントイベントマスクを設定する、または消去する

csx_RequestWindow(9F)
ウィンドウリソースを要求する、または解放する

csx_ResetFunction(9F)
PC カード上の関数をリセットする

csx_SetEventMask(9F)
クライアント用のクライアントイベントマスクを設定する、または戻す

csx_SetHandleOffset(9F)
現在のアクセスハンドルオフセットを設定する

csx_ValidateCIS(9F)
カード情報構造 (CIS) の妥当性検査を行う

cv_broadcast(9F)
condvar(9F) を参照

cv_destroy(9F)
condvar(9F) を参照

cv_init(9F)
condvar(9F) を参照

cv_signal(9F)
condvar(9F) を参照

cv_timedwait(9F)
condvar(9F) を参照

cv_timedwait_sig(9F)
condvar(9F) を参照

cv_wait(9F)
condvar(9F) を参照

cv_wait_sig(9F)
condvar(9F) を参照

datamsg(9F)
メッセージがデータメッセージかどうかをテストする

ddi_add_intr(9F)
ハードウェア割り込み処理ルーチン

ddi_add_softintr(9F)
ソフトウェア割り込み処理ルーチン

ddi_binding_name(9F)
ドライバの割り当て名を戻す

ddi_btop(9F)
ページサイズ変換

ddi_btopr(9F)
ddi_btop(9F) を参照

ddi_check_acc_handle(9F)
データアクセスおよび DMA ハンドルを検査する

ddi_check_dma_handle(9F)
ddi_check_acc_handle(9F) を参照

ddi_copyin(9F)
データをドライババッファーにコピーする

ddi_copyout(9F)
データをドライバからコピーする

ddi_create_minor_node(9F)
このデバイスのマイナーノードを作成する

ddi_dev_is_needed(9F)
デバイスのコンポーネントが必要であることをシステムに通知する

ddi_dev_is_sid(9F)
デバイスが自己識別形式であるかどうか表示する

ddi_dev_nintrs(9F)
デバイスが持つ割り込み指定数を戻す

ddi_dev_nregs(9F)
デバイスが持つレジスタセット数を戻す

ddi_dev_regsize(9F)
デバイスのレジスタのサイズを戻す

ddi_dev_report_fault(9F)
ハードウェア障害を報告する

ddi_device_copy(9F)
あるデバイスレジスタからほかのデバイスレジスタにデータをコピーする

ddi_device_zero(9F)
デバイスを 0 値で満たす

ddi_devid_compare(9F)
デバイス ID のカーネルインタフェース

ddi_devid_free(9F)
ddi_devid_compare(9F) を参照

ddi_devid_init(9F)
  ddi_devid_compare(9F) を参照

ddi_devid_register(9F)
  ddi_devid_compare(9F) を参照

ddi_devid_sizeof(9F)
  ddi_devid_compare(9F) を参照

ddi_devid_str_decode(9F)
  ddi_devid_compare(9F) を参照

ddi_devid_str_encode(9F)
  ddi_devid_compare(9F) を参照

ddi_devid_str_free(9F)
  ddi_devid_compare(9F) を参照

ddi_devid_unregister(9F)
  ddi_devid_compare(9F) を参照

ddi_devid_valid(9F)
  ddi_devid_compare(9F) を参照

ddi_devmap_segmap(9F)
  devmap_setup(9F) を参照

ddi_dma_addr_bind_handle(9F)
  アドレスを DMA ハンドルに割り当てる

ddi_dma_addr_setup(9F)
  仮想アドレスで使用するのに、より簡単な DMA 設定

ddi_dma_alloc_handle(9F)
  DMA ハンドルを割り当てる

ddi_dma_buf_bind_handle(9F)
  システムバッファーを DMA ハンドルに割り当てる

ddi_dma_buf_setup(9F)
  バッファー構造で使用するのに、より簡単な DMA 設定

ddi_dma_burstsizes(9F)
  DMA マッピングに許可されるバーストサイズを検出する

ddi_dma_coff(9F)
  DMA クッキーを DMA ハンドル内のオフセットに変換する

ddi_dma_curwin(9F)
  現在の DMA ウィンドウのオフセットおよびサイズを報告する

ddi_dma_devalign(9F)
  DMA マッピング整列および最小の転送サイズを見つける

ddi_dma_free(9F)
  システムの DMA リソースを解放する

ddi_dma_free_handle(9F)
DMA ハンドルを解放する

ddi_dma_get_attr(9F)
デバイスの DMA 属性構造を DMA ハンドルから入手する

ddi_dma_getwin(9F)
新しい DMA ウィンドウをアクティブにする

ddi_dma_htoc(9F)
DMA ハンドルを DMA アドレスクッキーに変換する

ddi_dma_mem_alloc(9F)
DMA 転送用のメモリーを割り当てる

ddi_dma_mem_free(9F)
以前割り当てられていたメモリーを解放する

ddi_dma_movwin(9F)
現在の DMA ウィンドウをシフトする

ddi_dma_nextcookie(9F)
後続の DMA クッキーを抽出する

ddi_dma_nextseg(9F)
次の DMA セグメントを取得する

ddi_dma_nextwin(9F)
次の DMA ウィンドウを取得する

ddi_dma_numwin(9F)
DMA ウィンドウの数を抽出する

ddi_dma_segtocookie(9F)
DMA セグメントを DMA アドレスクッキーに変換する

ddi_dma_set_sbus64(9F)
SBus 上で 64 ビット転送を許可する

ddi_dma_setup(9F)
DMA リソースを設定する

ddi_dma_sync(9F)
CPU とメモリーの入出力ビューを同期化する

ddi_dma_unbind_handle(9F)
DMA ハンドル内のアドレスを割り当て解除する

ddi_dmae(9F)
システム DMA エンジン関数

ddi_dmae_1stparty(9F)
ddi_dmae(9F) を参照

ddi_dmae_alloc(9F)
ddi_dmae(9F) を参照

ddi_dmae_disable(9F)
  ddi_dmae(9F) を参照

ddi_dmae_enable(9F)
  ddi_dmae(9F) を参照

ddi_dmae_getattr(9F)
  ddi_dmae(9F) を参照

ddi_dmae_getcnt(9F)
  ddi_dmae(9F) を参照

ddi_dmae_getlim(9F)
  ddi_dmae(9F) を参照

ddi_dmae_prog(9F)
  ddi_dmae(9F) を参照

ddi_dmae_release(9F)
  ddi_dmae(9F) を参照

ddi_dmae_stop(9F)
  ddi_dmae(9F) を参照

ddi_driver_major(9F)
  ドライバのメジャーデバイス番号を戻す

ddi_driver_name(9F)
  ドライバの正規化名を戻す

ddi_enter_critical(9F)
  制御のクリティカルな範囲に入る、およびそこから出る

ddi_exit_critical(9F)
  ddi_enter_critical(9F) を参照

ddi_ffs(9F)
  ロング正数の最初の (最後の) ビットセットを検索する

ddi_fls(9F)
  ddi_ffs(9F) を参照

ddi_get16(9F)
  ddi_get8(9F) を参照

ddi_get32(9F)
  ddi_get8(9F) を参照

ddi_get64(9F)
  ddi_get8(9F) を参照

ddi_get8(9F)
  マッピングされたメモリーアドレス、デバイスレジスタ、または割り当てられた DMA メモリーアドレスからデータを読み取る

ddi_get_cred(9F)
  呼び出し元の資格構造体へのポインタを戻す

ddi_get_devstate(9F)
  デバイスの状態を検査する

ddi_get_driver_private(9F)
  デバイスのプライベートデータ領域のアドレスを取得する、または設定する

ddi_get_iblock_cookie(9F)
  ddi_add_intr(9F) を参照

ddi_get_instance(9F)
  デバイスのインスタンス番号を取得する

ddi_get_kt_did(9F)
  現在のスレッドの識別子を取得する

ddi_get_lbolt(9F)
  lbolt の値を戻す

ddi_get_name(9F)
  ddi_binding_name(9F) を参照

ddi_get_parent(9F)
  デバイス情報構造体の親を検索する

ddi_get_pid(9F)
  プロセスを戻す

ddi_get_soft_iblock_cookie(9F)
  ddi_add_softintr(9F) を参照

ddi_get_soft_state(9F)
  ddi_soft_state(9F) を参照

ddi_get_time(9F)
  現在の時刻を秒単位で戻す

ddi_getb(9F)
  ddi_get8(9F) を参照

ddi_getiminor(9F)
  カーネル内部のマイナー番号を外部の dev_t から取得する

ddi_getl(9F)
  ddi_get8(9F) を参照

ddi_getll(9F)
  ddi_get8(9F) を参照

ddi_getlongprop(9F)
  ddi_prop_op(9F) を参照

ddi_getlongprop_buf(9F)
  ddi_prop_op(9F) を参照

ddi_getprop(9F)
  ddi_prop_op(9F) を参照

ddi_getproplen(9F)
  ddi_prop_op(9F) を参照

ddi_getw(9F)
  ddi_get8(9F) を参照

ddi_in_panic(9F)
  システムがパニック状態にあるかどうか判定する

ddi_intr_hilevel(9F)
  割り込みハンドラの種類を示す

ddi_io_get16(9F)
  ddi_io_get8(9F) を参照

ddi_io_get32(9F)
  ddi_io_get8(9F) を参照

ddi_io_get8(9F)
  入出力領域中のマッピングされたデバイスレジスタからデータを読み取る

ddi_io_getb(9F)
  ddi_io_get8(9F) を参照

ddi_io_getl(9F)
  ddi_io_get8(9F) を参照

ddi_io_getw(9F)
  ddi_io_get8(9F) を参照

ddi_io_put16(9F)
  ddi_io_put8(9F) を参照

ddi_io_put32(9F)
  ddi_io_put8(9F) を参照

ddi_io_put8(9F)
  入出力領域中のマッピングされたデバイスレジスタにデータを書き込む

ddi_io_putb(9F)
  ddi_io_put8(9F) を参照

ddi_io_putl(9F)
  ddi_io_put8(9F) を参照

ddi_io_putw(9F)
  ddi_io_put8(9F) を参照

ddi_io_rep_get16(9F)
  ddi_io_rep_get8(9F) を参照

ddi_io_rep_get32(9F)
  ddi_io_rep_get8(9F) を参照

ddi_io_rep_get8(9F)
入出力領域中のマッピングされたデバイスレジスタから複数のデータを読み取る

ddi_io_rep_getb(9F)
ddi_io_rep_get8(9F) を参照

ddi_io_rep_getl(9F)
ddi_io_rep_get8(9F) を参照

ddi_io_rep_getw(9F)
ddi_io_rep_get8(9F) を参照

ddi_io_rep_put16(9F)
ddi_io_rep_put8(9F) を参照

ddi_io_rep_put32(9F)
ddi_io_rep_put8(9F) を参照

ddi_io_rep_put8(9F)
入出力領域中のマッピングされたデバイスレジスタに複数のデータを書き込む

ddi_io_rep_putb(9F)
ddi_io_rep_put8(9F) を参照

ddi_io_rep_putl(9F)
ddi_io_rep_put8(9F) を参照

ddi_io_rep_putw(9F)
ddi_io_rep_put8(9F) を参照

ddi_iomin(9F)
DMA の最小整列サイズおよび最小転送サイズを見つける

ddi_iopb_alloc(9F)
連続しないアクセス用メモリーを割り当てる、および解放する

ddi_iopb_free(9F)
ddi_iopb_alloc(9F) を参照

ddi_log_sysevent(9F)
ドライバ用システムイベントを記録する

ddi_map_regs(9F)
レジスタをマップする、またはマップ解除する

ddi_mapdev(9F)
ドライバ制御のデバイスマッピングを作成する

ddi_mapdev_intercept(9F)
ユーザーアクセスのドライバ通知を制御する

ddi_mapdev_nointercept(9F)
ddi_mapdev_intercept(9F) を参照

ddi_mapdev_set_device_acc_attr(9F)
マッピング用のデバイス属性を設定する

ddi_mem_alloc(9F)
連続するアクセス用メモリーを割り当てる、または解放する

ddi_mem_free(9F)
ddi_mem_alloc(9F) を参照

ddi_mem_get16(9F)
ddi_mem_get8(9F) を参照

ddi_mem_get32(9F)
ddi_mem_get8(9F) を参照

ddi_mem_get64(9F)
ddi_mem_get8(9F) を参照

ddi_mem_get8(9F)
メモリー空間または割り当てられた DMA メモリー内のマッピングされたデバイスからデータを読み取る

ddi_mem_getb(9F)
ddi_mem_get8(9F) を参照

ddi_mem_getl(9F)
ddi_mem_get8(9F) を参照

ddi_mem_getll(9F)
ddi_mem_get8(9F) を参照

ddi_mem_getw(9F)
ddi_mem_get8(9F) を参照

ddi_mem_put16(9F)
ddi_mem_put8(9F) を参照

ddi_mem_put32(9F)
ddi_mem_put8(9F) を参照

ddi_mem_put64(9F)
ddi_mem_put8(9F) を参照

ddi_mem_put8(9F)
メモリー空間または割り当てられた DMA メモリー内のマッピングされたデバイスにデータを書き込む

ddi_mem_putb(9F)
ddi_mem_put8(9F) を参照

ddi_mem_putl(9F)
ddi_mem_put8(9F) を参照

ddi_mem_putll(9F)
ddi_mem_put8(9F) を参照

ddi_mem_putw(9F)
ddi_mem_put8(9F) を参照

ddi_mem_rep_get16(9F)
  ddi_mem_rep_get8(9F) を参照

ddi_mem_rep_get32(9F)
  ddi_mem_rep_get8(9F) を参照

ddi_mem_rep_get64(9F)
  ddi_mem_rep_get8(9F) を参照

ddi_mem_rep_get8(9F)
  メモリー空間または割り当てられた DMA メモリー内のマッピングされたデバイスから複数のデータを読み取る

ddi_mem_rep_getb(9F)
  ddi_mem_rep_get8(9F) を参照

ddi_mem_rep_getl(9F)
  ddi_mem_rep_get8(9F) を参照

ddi_mem_rep_getll(9F)
  ddi_mem_rep_get8(9F) を参照

ddi_mem_rep_getw(9F)
  ddi_mem_rep_get8(9F) を参照

ddi_mem_rep_put16(9F)
  ddi_mem_rep_put8(9F) を参照

ddi_mem_rep_put32(9F)
  ddi_mem_rep_put8(9F) を参照

ddi_mem_rep_put64(9F)
  ddi_mem_rep_put8(9F) を参照

ddi_mem_rep_put8(9F)
  メモリー空間または割り当てられた DMA メモリー内のマッピングされたデバイスに複数のデータを書き込む

ddi_mem_rep_putb(9F)
  ddi_mem_rep_put8(9F) を参照

ddi_mem_rep_putl(9F)
  ddi_mem_rep_put8(9F) を参照

ddi_mem_rep_putll(9F)
  ddi_mem_rep_put8(9F) を参照

ddi_mem_rep_putw(9F)
  ddi_mem_rep_put8(9F) を参照

ddi_mmap_get_model(9F)
  現在のスレッドのデータモデルタイプを戻す

ddi_model_convert_from(9F)
  データモデルタイプの不一致を判定する

ddi_no_info(9F)
  getinfo(9E) のスタブ

ddi_node_name(9F)
  dev_info のノード名を戻す

ddi_peek(9F)
  位置から値を読み取る

ddi_peek16(9F)
  ddi_peek(9F) を参照

ddi_peek32(9F)
  ddi_peek(9F) を参照

ddi_peek64(9F)
  ddi_peek(9F) を参照

ddi_peek8(9F)
  ddi_peek(9F) を参照

ddi_peekc(9F)
  ddi_peek(9F) を参照

ddi_peekd(9F)
  ddi_peek(9F) を参照

ddi_peekl(9F)
  ddi_peek(9F) を参照

ddi_peeks(9F)
  ddi_peek(9F) を参照

ddi_poke(9F)
  位置に値を書き込む

ddi_poke16(9F)
  ddi_poke(9F) を参照

ddi_poke32(9F)
  ddi_poke(9F) を参照

ddi_poke64(9F)
  ddi_poke(9F) を参照

ddi_poke8(9F)
  ddi_poke(9F) を参照

ddi_pokec(9F)
  ddi_poke(9F) を参照

ddi_poked(9F)
  ddi_poke(9F) を参照

ddi_pokel(9F)
  ddi_poke(9F) を参照

ddi_pokes(9F)
  ddi_poke(9F) を参照

ddi_prop_create(9F)
  リーフデバイスドライバのプロパティを作成、削除、または修正する

ddi_prop_exists(9F)
  プロパティが存在するかどうか検査する

ddi_prop_free(9F)
  ddi_prop_lookup(9F) を参照

ddi_prop_get_int(9F)
  整数プロパティを検索する

ddi_prop_get_int64(9F)
  ddi_prop_get_int(9F) を参照

ddi_prop_lookup(9F)
  プロパティ情報を検索する

ddi_prop_lookup_byte_array(9F)
  ddi_prop_lookup(9F) を参照

ddi_prop_lookup_int64_array(9F)
  ddi_prop_lookup(9F) を参照

ddi_prop_lookup_int_array(9F)
  ddi_prop_lookup(9F) を参照

ddi_prop_lookup_string(9F)
  ddi_prop_lookup(9F) を参照

ddi_prop_lookup_string_array(9F)
  ddi_prop_lookup(9F) を参照

ddi_prop_modify(9F)
  ddi_prop_create(9F) を参照

ddi_prop_op(9F)
  リーフデバイスドライバ用のプロパティ情報を取得する

ddi_prop_remove(9F)
  ddi_prop_create(9F) を参照

ddi_prop_remove_all(9F)
  ddi_prop_create(9F) を参照

ddi_prop_undefine(9F)
  ddi_prop_create(9F) を参照

ddi_prop_update(9F)
  プロパティを更新する

ddi_prop_update_byte_array(9F)
  ddi_prop_update(9F) を参照

ddi_prop_update_int(9F)
  ddi_prop_update(9F) を参照

ddi_prop_update_int64(9F)
  ddi_prop_update(9F) を参照

ddi_prop_update_int64_array(9F)
  ddi_prop_update(9F) を参照

ddi_prop_update_int_array(9F)
  ddi_prop_update(9F) を参照

ddi_prop_update_string(9F)
  ddi_prop_update(9F) を参照

ddi_prop_update_string_array(9F)
  ddi_prop_update(9F) を参照

ddi_ptob(9F)
  ddi_btop(9F) を参照

ddi_put16(9F)
  ddi_put8(9F) を参照

ddi_put32(9F)
  ddi_put8(9F) を参照

ddi_put64(9F)
  ddi_put8(9F) を参照

ddi_put8(9F)
  マッピングされたメモリーアドレス、デバイスレジスタ、または割り当てられた DMA メモリーアドレスにデータを書き込む

ddi_putb(9F)
  ddi_put8(9F) を参照

ddi_putl(9F)
  ddi_put8(9F) を参照

ddi_putll(9F)
  ddi_put8(9F) を参照

ddi_putw(9F)
  ddi_put8(9F) を参照

ddi_regs_map_free(9F)
  以前マッピングされたレジスタアドレス空間を解放する

ddi_regs_map_setup(9F)
  レジスタアドレス空間用のマッピングを設定する

ddi_remove_intr(9F)
  ddi_add_intr(9F) を参照

ddi_remove_minor_node(9F)
  この dev_info 用のマイナーノードを削除する

ddi_remove_softintr(9F)
  ddi_add_softintr(9F) を参照

ddi_removing_power(9F)
  DDI_SUSPEND を実行すると電源がデバイスから削除されるかどうか検査する

ddi_rep_get16(9F)
  ddi_rep_get8(9F) を参照

ddi_rep_get32(9F)
  ddi_rep_get8(9F) を参照

ddi_rep_get64(9F)
  ddi_rep_get8(9F) を参照

ddi_rep_get8(9F)
  マッピングされたメモリーアドレス、デバイスレジスタ、または割り当てられた DMA メモリーアドレスからデータを読み取る

ddi_rep_getb(9F)
  ddi_rep_get8(9F) を参照

ddi_rep_getl(9F)
  ddi_rep_get8(9F) を参照

ddi_rep_getll(9F)
  ddi_rep_get8(9F) を参照

ddi_rep_getw(9F)
  ddi_rep_get8(9F) を参照

ddi_rep_put16(9F)
  ddi_rep_put8(9F) を参照

ddi_rep_put32(9F)
  ddi_rep_put8(9F) を参照

ddi_rep_put64(9F)
  ddi_rep_put8(9F) を参照

ddi_rep_put8(9F)
  マッピングされたメモリーアドレス、デバイスレジスタ、または割り当てられた DMA メモリーアドレスにデータを書き込む

ddi_rep_putb(9F)
  ddi_rep_put8(9F) を参照

ddi_rep_putl(9F)
  ddi_rep_put8(9F) を参照

ddi_rep_putll(9F)
  ddi_rep_put8(9F) を参照

ddi_rep_putw(9F)
  ddi_rep_put8(9F) を参照

ddi_report_dev(9F)
  デバイスを通知する

ddi_root_node(9F)
  dev_info ツリーのルートを取得する

ddi_segmap(9F)
  seg_dev を使用してユーザーマッピングを設定する

ddi_segmap_setup(9F)
  ddi_segmap(9F) を参照

ddi_set_driver_private(9F)
  ddi_get_driver_private(9F) を参照

ddi_slaveonly(9F)
  デバイスが、スレーブのみがアクセスできる位置にインストールされているかどうかを知らせる

ddi_soft_state(9F)
  ドライバソフトウェア状態ユーティリティルーチン

ddi_soft_state_fini(9F)
  ddi_soft_state(9F) を参照

ddi_soft_state_free(9F)
  ddi_soft_state(9F) を参照

ddi_soft_state_init(9F)
  ddi_soft_state(9F) を参照

ddi_soft_state_zalloc(9F)
  ddi_soft_state(9F) を参照

ddi_trigger_softintr(9F)
  ddi_add_softintr(9F) を参照

ddi_umem_alloc(9F)
  ページ揃えのカーネルメモリーを割り当てる、および解放する

ddi_umem_free(9F)
  ddi_umem_alloc(9F) を参照

ddi_umem_iosetup(9F)
  アプリケーションメモリーに対し、入出力要求を設定する

ddi_umem_lock(9F)
  メモリーページをロックする、およびロック解除する

ddi_umem_unlock(9F)
  ddi_umem_lock(9F) を参照

ddi_unmap_regs(9F)
ddi_map_regs(9F) を参照

delay(9F)
指定した数のクロックティックを遅延実行する

devmap_default_access(9F)
デフォルトのドライバメモリーアクセス関数

devmap_devmem_setup(9F)
ドライバメモリーマッピングパラメータを設定する

devmap_do_ctxmgt(9F)
マッピング上のデバイスコンテキストスイッチングを実行する

devmap_load(9F)
devmap_unload(9F) を参照

devmap_set_ctx_timeout(9F)
コンテキスト管理コールバック用のタイムアウト値を設定する

devmap_setup(9F)
devmap フレームワークを使用して、デバイスメモリーへのユーザーマッピングを設定する

devmap_umem_setup(9F)
devmap_devmem_setup(9F) を参照

devmap_unload(9F)
メモリーアドレス変換の妥当性検査を制御する

disksort(9F)
バッファー用単一方向エレベータ式シークソート

drv_getparm(9F)
カーネル状態情報を抽出する

drv_hztousec(9F)
クロックティックをマイクロ秒に変換する

drv_priv(9F)
ドライバ特権を判定する

drv_usectohz(9F)
マイクロ秒をクロックティックに変換する

drv_usecwait(9F)
指定した間隔でビジー待機する

dupb(9F)
メッセージブロック記述子を複製する

dupmsg(9F)
メッセージを複製する

enableok(9F)
サービス用にキューを再スケジューリングする

esballoc(9F)
呼び出し元が提供するバッファーを使用してメッセージブロックを割り当てる

esbbcall(9F)
バッファーが使用可能になったら関数を呼び出す

flushband(9F)
指定した優先順位の帯域用のメッセージをフラッシュする

flushq(9F)
キューからメッセージを削除する

free_pktiopb(9F)
get_pktiopb(9F) を参照

freeb(9F)
メッセージブロックを解放する

freemsg(9F)
メッセージ内のすべてのメッセージブロックを解放する

freerbuf(9F)
raw バッファーのヘッダーを解放する

freezestr(9F)
ストリームの状態をフリーズ、およびフリーズ解除する

get_pktiopb(9F)
iopb マップ内の SCSI パケットを割り当てる、または解放する

geterror(9F)
入出力エラーを戻す

gethrtime(9F)
高解像度時間を取得する

getmajor(9F)
メジャーデバイス番号を取得する

getminor(9F)
マイナーデバイス番号を取得する

getq(9F)
キューから次のメッセージを取得する

getrbuf(9F)
raw バッファーのヘッダーを取得する

gld(9F)
汎用 LAN ドライバサービスルーチン

gld_intr(9F)
gld(9F) を参照

gld_mac_alloc(9F)
  gld(9F) を参照

gld_mac_free(9F)
  gld(9F) を参照

gld_recv(9F)
  gld(9F) を参照

gld_register(9F)
  gld(9F) を参照

gld_sched(9F)
  gld(9F) を参照

gld_unregister(9F)
  gld(9F) を参照

hat_getkpfnum(9F)
  カーネルアドレスからページフレーム番号を取得する

id32_alloc(9F)
  32 ビットドライバ ID 管理ルーチン

id32_free(9F)
  id32_alloc(9F) を参照

id32_lookup(9F)
  id32_alloc(9F) を参照

inb(9F)
  入出力ポートから読み取る

inl(9F)
  inb(9F) を参照

insq(9F)
  メッセージをキューに挿入する

inw(9F)
  inb(9F) を参照

kmem_alloc(9F)
  カーネルメモリーを割り当てる

kmem_cache_alloc(9F)
  kmem_cache_create(9F) を参照

kmem_cache_create(9F)
  カーネルメモリーキャッシュ割り当て処理

kmem_cache_destroy(9F)
  kmem_cache_create(9F) を参照

kmem_cache_free(9F)
  kmem_cache_create(9F) を参照

kmem_free(9F)
  kmem_alloc(9F) を参照

kmem_zalloc(9F)
  kmem_alloc(9F) を参照

kstat_create(9F)
  新しい kstat を作成、および初期化する

kstat_delete(9F)
  システムから kstat を削除する

kstat_install(9F)
  完全に初期化された kstat をシステムに追加する

kstat_named_init(9F)
  名前の付いた kstat を初期化する

kstat_named_setstr(9F)
  kstat_named_init(9F) を参照

kstat_queue(9F)
  入出力 kstat 統計を更新する

kstat_runq_back_to_waitq(9F)
  kstat_queue(9F) を参照

kstat_runq_enter(9F)
  kstat_queue(9F) を参照

kstat_runq_exit(9F)
  kstat_queue(9F) を参照

kstat_waitq_enter(9F)
  kstat_queue(9F) を参照

kstat_waitq_exit(9F)
  kstat_queue(9F) を参照

kstat_waitq_to_runq(9F)
  kstat_queue(9F) を参照

linkb(9F)
  2 つのメッセージブロックを連結する

makecom(9F)
  SCSI コマンド用パケットを作成する

makecom_g0(9F)
  makecom(9F) を参照

makecom_g0_s(9F)
  makecom(9F) を参照

makecom_g1(9F)
  makecom(9F) を参照

makecom_g5(9F)
makecom(9F) を参照

makedevice(9F)
メジャー番号およびマイナー番号からデバイス番号を作成する

max(9F)
2 つの整数のうち、大きい方を戻す

min(9F)
2 つの整数のうち、小さい方を戻す

minphys(9F)
physio(9F) を参照

mkiocb(9F)
カーネル中の M_IOCTL メッセージのために STREAMS ioctl ブロックを割り当てる

mod_info(9F)
mod_install(9F) を参照

mod_install(9F)
読み込み可能なモジュールを追加、削除、または照会する

mod_remove(9F)
mod_install(9F) を参照

msgdsize(9F)
メッセージ中のバイト数を戻す

msgpullup(9F)
メッセージ中のバイトを連結する

mt-streams(9F)
STREAMS マルチスレッド処理

mutex(9F)
相互排他ロックルーチン

mutex_destroy(9F)
mutex(9F) を参照

mutex_enter(9F)
mutex(9F) を参照

mutex_exit(9F)
mutex(9F) を参照

mutex_init(9F)
mutex(9F) を参照

mutex_owned(9F)
mutex(9F) を参照

mutex_tryenter(9F)
  mutex(9F) を参照

nochpoll(9F)
  ポーリング不可のデバイス用エラー戻り関数

nodev(9F)
  エラー戻り関数

noenable(9F)
  キューがスケジューリングされるのを防ぐ

nulldev(9F)
  ゼロ戻り関数

numtos(9F)
  stoi(9F) を参照

nvlist_add_boolean(9F)
  nvlist_t に名前と値の新しいペアを追加する

nvlist_add_byte(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_byte_array(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_int16(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_int16_array(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_int32(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_int32_array(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_int64(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_int64_array(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_string(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_string_array(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_uint16(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_uint16_array(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_uint32(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_uint32_array(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_uint64(9F)
  nvlist_add_boolean(9F) を参照

nvlist_add_uint64_array(9F)
  nvlist_add_boolean(9F) を参照

nvlist_alloc(9F)
  名前と値のペアのリストを管理する

nvlist_dup(9F)
  nvlist_alloc(9F) を参照

nvlist_free(9F)
  nvlist_alloc(9F) を参照

nvlist_lookup_boolean(9F)
  インタフェース名が示す名前と種類の一致が見つかったら、データ値を抽出する

nvlist_lookup_byte(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_byte_array(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_int16(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_int16_array(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_int32(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_int32_array(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_int64(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_int64_array(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_string(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_string_array(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_uint16(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_uint16_array(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_uint32(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_uint32_array(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_uint64(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_lookup_uint64_array(9F)
  nvlist_lookup_boolean(9F) を参照

nvlist_next_nvpair(9F)
  名前と値のペアに関するデータを戻す

nvlist_pack(9F)
  nvlist_alloc(9F) を参照

nvlist_remove(9F)
  名前と値のペアを削除する

nvlist_remove_all(9F)
  nvlist_remove(9F) を参照

nvlist_size(9F)
  nvlist_alloc(9F) を参照

nvlist_unpack(9F)
  nvlist_alloc(9F) を参照

nvpair_name(9F)
  nvlist_next_nvpair(9F) を参照

nvpair_type(9F)
  nvlist_next_nvpair(9F) を参照

nvpair_value_byte(9F)
  名前と値のペアから値を抽出する

nvpair_value_byte_array(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_int16(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_int16_array(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_int32(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_int32_array(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_int64(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_int64_array(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_string(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_string_array(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_uint16(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_uint16_array(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_uint32(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_uint32_array(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_uint64(9F)
  nvpair_value_byte(9F) を参照

nvpair_value_uint64_array(9F)
  nvpair_value_byte(9F) を参照

otherq(9F)
  OTHERQ(9F) を参照

outb(9F)
  入出力ポートに書き込む

outl(9F)
  outb(9F) を参照

outw(9F)
  outb(9F) を参照

pci_config_get16(9F)
  pci_config_get8(9F) を参照

pci_config_get32(9F)
  pci_config_get8(9F) を参照

pci_config_get64(9F)
  pci_config_get8(9F) を参照

pci_config_get8(9F)
  いろいろなサイズの単一データを PCI ローカルバス構成空間から読み取る、または書き込む

pci_config_getb(9F)
  pci_config_get8(9F) を参照

pci_config_getl(9F)
  pci_config_get8(9F) を参照

pci_config_getll(9F)
  pci_config_get8(9F) を参照

pci_config_getw(9F)
  pci_config_get8(9F) を参照

pci_config_put16(9F)
  pci_config_get8(9F) を参照

pci_config_put32(9F)
  pci_config_get8(9F) を参照

pci_config_put64(9F)
  pci_config_get8(9F) を参照

pci_config_put8(9F)
  pci_config_get8(9F) を参照

pci_config_putb(9F)
  pci_config_get8(9F) を参照

pci_config_putl(9F)
  pci_config_get8(9F) を参照

pci_config_putll(9F)
  pci_config_get8(9F) を参照

pci_config_putw(9F)
  pci_config_get8(9F) を参照

pci_config_setup(9F)
  PCI ローカルバス構成空間へのアクセスを可能にするために、リソースを設定する、または破壊する

pci_config_teardown(9F)
  pci_config_setup(9F) を参照

pci_report_pmcap(9F)
  PCI デバイスの電源管理能力を報告する

pci_restore_config_regs(9F)
  pci_save_config_regs(9F) を参照

pci_save_config_regs(9F)
  PCI 構成レジスタを保存および復元する

physio(9F)
  物理入出力を実行する

pm_busy_component(9F)
電源管理用デバイスコンポーネント利用度を制御する

pm_create_components(9F)
電源管理可能なコンポーネントを作成または破壊する

pm_destroy_components(9F)
pm_create_components(9F) を参照

pm_get_normal_power(9F)
デバイスコンポーネントの通常の電源レベルを取得または設定する

pm_idle_component(9F)
pm_busy_component(9F) を参照

pm_lower_power(9F)
pm_raise_power(9F) を参照

pm_power_has_changed(9F)
独立した電源レベル変更を電源管理フレームワークに通知する

pm_raise_power(9F)
コンポーネントの電力を高くする、または低くする

pm_set_normal_power(9F)
pm_get_normal_power(9F) を参照

pm_trans_check(9F)
デバイスの電力再投入アドバイザリチェック

pollwakeup(9F)
イベントが発生したことをプロセスに通知する

proc_ref(9F)
proc_signal(9F) を参照

proc_signal(9F)
プロセスにシグナルを送る

proc_unref(9F)
proc_signal(9F) を参照

ptob(9F)
ページ単位のサイズをバイト単位のサイズに変換する

pullupmsg(9F)
メッセージ内のバイトを連結する

put(9F)
STREAMS の put 手続きを呼び出す

putbq(9F)
メッセージをキューの先頭に置く

putctl(9F)
制御メッセージをキューに送信する

putctl1(9F)
制御メッセージを 1 バイトのパラメータと一緒にキューに送信する

putnext(9F)
メッセージを次のキューに送信する

putnextctl(9F)
制御メッセージをキューに送信する

putnextctl1(9F)
制御メッセージを 1 バイトのパラメータと一緒にキューに送信する

putq(9F)
メッセージをキューに配置する

qassociate(9F)
STREAMS キューをドライバインスタンスと関連づける

qbufcall(9F)
バッファーが使用可能になったら関数を呼び出す

qenable(9F)
キューを有効にする

qprocsoff(9F)
qprocson(9F) を参照

qprocson(9F)
put ルーチンおよびサービスルーチンを有効、または無効にする

qreply(9F)
ストリーム上でメッセージを逆方向に送信する

qsize(9F)
キューにあるメッセージ数を検出する

qtimeout(9F)
指定した時間が経過したら、関数を実行する

qunbufcall(9F)
保留中の qbufcall 要求を取り消す

quntimeout(9F)
以前の qtimeout 関数呼び出しを取り消す

qwait(9F)
STREAMS 待機ルーチン

qwait_sig(9F)
qwait(9F) を参照

qwriter(9F)
非同期の STREAMS パラメータをアップグレードする

rd(9F)
RD(9F) を参照

repinsb(9F)
  inb(9F) を参照

repinsd(9F)
  inb(9F) を参照

repinsw(9F)
  inb(9F) を参照

repoutsb(9F)
  outb(9F) を参照

repoutsd(9F)
  outb(9F) を参照

repoutsw(9F)
  outb(9F) を参照

rmalloc(9F)
  リソースマップから空間を割り当てる

rmalloc_wait(9F)
  リソースマップから空間を割り当て、必要に応じて待機する

rmallocmap(9F)
  リソースマップを割り当てる、および解放する

rmallocmap_wait(9F)
  rmallocmap(9F) を参照

rmfree(9F)
  空間を解放してリソースマップに戻す

rmfreemap(9F)
  rmallocmap(9F) を参照

rmvb(9F)
  メッセージからメッセージブロックを削除する

rmvq(9F)
  キューからメッセージを削除する

rw_destroy(9F)
  rwlock(9F) を参照

rw_downgrade(9F)
  rwlock(9F) を参照

rw_enter(9F)
  rwlock(9F) を参照

rw_exit(9F)
  rwlock(9F) を参照

rw_init(9F)
  rwlock(9F) を参照

rw_read_locked(9F)
  rwlock(9F) を参照

rw_tryenter(9F)
  rwlock(9F) を参照

rw_tryupgrade(9F)
  rwlock(9F) を参照

rwlock(9F)
  リーダー/ライターロック関数

samestr(9F)
  SAMESTR(9F) を参照

scsi_abort(9F)
  SCSI コマンドを強制終了する

scsi_alloc_consistent_buf(9F)
  SCSI DMA 用の入出力バッファーを割り当てる

scsi_cname(9F)
  SCSI 名を復号化する

scsi_destroy_pkt(9F)
  割り当てられた SCSI パケットおよびその DMA リソースを解放する

scsi_dmafree(9F)
  scsi_dmaget(9F) を参照

scsi_dmaget(9F)
  SCSI DMA ユーティリティルーチン

scsi_dname(9F)
  scsi_cname(9F) を参照

scsi_errmsg(9F)
  SCSI リクエストセンスメッセージを表示する

scsi_free_consistent_buf(9F)
  以前割り当てられた SCSI DMA 入出力バッファーを解放する

scsi_get_device_type_scsi_options(9F)
  デバイスの種類ごとに SCSI オプションプロパティを検索する

scsi_hba_attach(9F)
  scsi_hba_attach_setup(9F) を参照

scsi_hba_attach_setup(9F)
  SCSI HBA によるルーチンの接続および切り離し

scsi_hba_detach(9F)
  scsi_hba_attach_setup(9F) を参照

scsi_hba_fini(9F)
  scsi_hba_init(9F) を参照

scsi_hba_init(9F)
SCSI ホストバスアダプタシステムの初期化ルーチンおよび完了ルーチン

scsi_hba_lookup_capstr(9F)
インデックスマッチング機能文字列を戻す

scsi_hba_pkt_alloc(9F)
scsi_pkt 構造体を割り当てる、および解放する

scsi_hba_pkt_free(9F)
scsi_hba_pkt_alloc(9F) を参照

scsi_hba_probe(9F)
デフォルトの SCSI HBA プローブ関数

scsi_hba_tran_alloc(9F)
トランスポート構造体を割り当てる、および解放する

scsi_hba_tran_free(9F)
scsi_hba_tran_alloc(9F) を参照

scsi_ifgetcap(9F)
SCSI トランスポート機能の取得および設定

scsi_ifsetcap(9F)
scsi_ifgetcap(9F) を参照

scsi_init_pkt(9F)
完全な SCSI パケットを用意する

scsi_log(9F)
SCSI デバイス関連メッセージを表示する

scsi_mname(9F)
scsi_cname(9F) を参照

scsi_pktalloc(9F)
SCSI パケットユーティリティルーチン

scsi_pktfree(9F)
scsi_pktalloc(9F) を参照

scsi_poll(9F)
ターゲットドライバにかわってポーリングされた SCSI コマンドを実行する

scsi_probe(9F)
SCSI デバイスを検索するユーティリティ

scsi_resalloc(9F)
scsi_pktalloc(9F) を参照

scsi_reset(9F)
SCSI バスまたはターゲットをリセットする

scsi_reset_notify(9F)
バスのリセットをターゲットドライバに通知する

scsi_resfree(9F)
  scsi_pktalloc(9F) を参照

scsi_rname(9F)
  scsi_cname(9F) を参照

scsi_setup_cdb(9F)
  SCSI コマンド記述子ブロック (CDB) を設定する

scsi_slave(9F)
  ターゲットの存在を確立するための SCSI ターゲットドライバ用ユーティリティ

scsi_sname(9F)
  scsi_cname(9F) を参照

scsi_sync_pkt(9F)
  CPU とメモリーの入出力ビューを同期化する

scsi_transport(9F)
  SCSI ターゲットドライバによる、コマンド起動要求

scsi_unprobe(9F)
  最初の検索時に割り当てられたリソースを解放する

scsi_unslave(9F)
  scsi_unprobe(9F) を参照

scsi_vu_errmsg(9F)
  SCSI リクエストセンスメッセージを表示する

sema_destroy(9F)
  semaphore(9F) を参照

sema_init(9F)
  semaphore(9F) を参照

sema_p(9F)
  semaphore(9F) を参照

sema_p_sig(9F)
  semaphore(9F) を参照

sema_tryp(9F)
  semaphore(9F) を参照

sema_v(9F)
  semaphore(9F) を参照

semaphore(9F)
  セマフォ関数

snprintf(9F)
  sprintf(9F) を参照

sprintf(9F)
  メモリー中の文字を書式設定する

`stoi`(9F)
整数と 16 進文字列間の変換を行う

`strcasecmp`(9F)
`strcmp`(9F) を参照

`strchr`(9F)
文字列中の 1 文字を検索する

`strcmp`(9F)
NULL で終わっている 2 つの文字列を比較する

`strcpy`(9F)
ある位置からほかの位置に文字列をコピーする

`strlen`(9F)
文字列中の NULL 以外のバイト数を判定する

`strlog`(9F)
ログドライバにメッセージを送る

`strncasecmp`(9F)
`strcmp`(9F) を参照

`strncmp`(9F)
`strcmp`(9F) を参照

`strncpy`(9F)
`strcpy`(9F) を参照

`strqget`(9F)
キューまたはキューの帯域に関する情報を取得する

`strqset`(9F)
キューまたはキューの帯域に関する情報を変更する

`strrchr`(9F)
`strchr`(9F) を参照

`swab`(9F)
バイトを単語半分にあたる 16 ビットでスワップする

`testb`(9F)
使用可能なバッファーがあるかどうか検査する

`timeout`(9F)
指定した時間が経過したら関数を実行する

`uiomove`(9F)
uio 構造体を使用してカーネルデータをコピーする

`unbufcall`(9F)
保留中の bufcall 要求を取り消す

`unfreezestr`(9F)
`freezestr`(9F) を参照

unlinkb(9F)
メッセージの先頭からメッセージブロックを削除する

untimeout(9F)
以前のタイムアウト関数呼び出しを取り消す

ureadc(9F)
uio 構造体に文字を追加する

uwritec(9F)
uio 構造体から文字を削除する

va_arg(9F)
変数の引数リストを処理する

va_copy(9F)
va_arg(9F) を参照

va_end(9F)
va_arg(9F) を参照

va_start(9F)
va_arg(9F) を参照

vcmn_err(9F)
cmn_err(9F) を参照

vsprintf(9F)
メモリー内の文字を書式設定する

wr(9F)
WR(9F) を参照

コマンド一覧

名前	説明
Intro(9F)	DDI/DKI 関数の序章
scsi_hba_attach(9f)	scsi_hba_attach_setup(9F) を参照
scsi_hba_attach_setup(9F)	SCSI HBA 接続および切り離しルーチン
scsi_hba_detach(9f)	scsi_hba_attach_setup(9F) を参照

名前

scsi_hba_attach_setup, scsi_hba_attach, scsi_hba_detach – SCSI HBA 接続および切り離しルーチン

形式

```
#include <sys/scsi/scsi.h>

int scsi_hba_attach_setup(dev_info_t *dip, ddi_dma_attr_t
    *hba_dma_attr, scsi_hba_tran_t *hba_tran, int hba_flags);

int scsi_hba_attach(dev_info_t *dip, ddi_dma_lim_t *hba_lim,
    scsi_hba_tran_t *hba_tran, int hba_flags, void *hba_options);

int scsi_hba_detach(dev_info_t *dip);
```

インタフェースレベル

Solaris アーキテクチャ固有 (Solaris DDI)。

パラメータ

dip dev_info_t 構造に対するポインタで、HBA デバイスのインスタンスを参照します。

hba_lim ddi_dma_lim(9S) 構造に対するポインタ。

hba_tran scsi_hba_tran(9S) 構造に対するポインタ。

hba_flags フラグ修飾子。唯一の定義済みフラグ値は、SCSI_HBA_TRAN_CLONE です。

hba_options 将来の拡張に備えて HBA ドライバによって提供されるオプションの機能で、NULL でなければなりません。

hba_dma_attr ddi_dma_attr(9S) 構造に対するポインタ。

機能説明

scsi_hba_attach_setup() は、scsi_hba_attach() に代わる推奨インタフェースです。

scsi_hba_attach_setup() および scsi_hba_attach() に関して：

scsi_hba_attach() は、DMA 限界値 *hba_lim* と、*dip* によって定義される HBA デバイスの各インスタンスの転送ベクトル *hba_tran* を登録します。scsi_hba_attach_setup() は、DMA 属性 *hba_dma_attr* と、*dip* によって定義される HBA デバイスの各インスタンスの転送ベクトル *hba_tran* を登録します。HBA ドライバは、異なる DMA 限界値または DMA 属性、および必要に応じてデバイスの各インタフェースに対する転送ベクトルを渡すことができ、HBA 自身によって課せられたいかなる制約にも対応します。

scsi_hba_attach() および scsi_hba_attach_setup() は、dev_ops(9S) 構造内の dev_bus_ops フィールドを使用します。 HBA ドライバは、scsi_hba_attach() または scsi_hba_attach_setup() を呼び出す前に、このフィールドを NULL に初期化する必要があります。

SCSI_HBA_TRAN_CLONE が *hba_flags* で要求される場合、*hba_tran* 構造は、各ターゲットが HBA に接続されるたびに 1 回複製されます。構造の複製は、ターゲットを初期化するために tran_tgt_init(9E) エントリポイントが呼び出される前に行われます。後続のすべての HBA エントリポイント (tran_tgt_init(9E) も含めて) で、引数として渡された、または scsi_address 構造で検出された scsi_hba_tran_t

構造は、「複製された」 scsi_hba_tran_t 構造になるので、HBA は、scsi_hba_tran_t の tran_tgt_private フィールドを使用してターゲットごとのデータを指定できます。HBA は、切り離しの際に割り当てられたものと同じ scsi_hba_tran_t 構造だけを注意して解放する必要があります。システムによって割り当てられたすべての「複製された」 scsi_hba_tran_t 構造は、システムによって解放されます。

scsi_hba_attach() および scsi_hba_attach_setup() は、同じ名前の属性がすでにそのノードに接続されていない限り、多数の整数値属性を *dip* に接続します。HBA ドライバは、これらの構成パラメータを ddi_prop_get_int(9F) を使用して検出し、HBA を提供した機能に対するあらゆる設定に従う必要があります。

scsi-options
  省略可能な SCSI 構成ビット

  SCSI_OPTIONS_DR
    設定しない場合、HBA はターゲットデバイスに切り離し (Disconnect) 特権を認可しません。

  SCSI_OPTIONS_LINK
    設定しない場合、HBA はリンクされたコマンド (Linked Commands) を有効にしません。

  SCSI_OPTIONS_TAG
    設定しない場合、HBA は、コマンドタグ付き待ち行列 (Command Tagged Queing) モードでは動作しません。

  SCSI_OPTIONS_PARITY
    設定しない場合、HBA はパリティーモードでは作動しません。

  SCSI_OPTIONS_QAS
    設定しない場合、HBA はクイックアービトレーション選択 (Quick Arbitration Select) 機能を利用しません。ご使用のマシンが QAS をサポートするかどうかは、サンのハードウェアマニュアルで確認してください。

  SCSI_OPTIONS_FAST
    設定しない場合、HBA は、バスを FAST SCSI モードで機能させません。

  SCSI_OPTIONS_FAST20
    設定しない場合、HBA は、バスを FAST20 SCSI モードで機能させません。

  SCSI_OPTIONS_FAST40
    設定しない場合、HBA は、バスを FAST40 SCSI モードで機能させません。

  SCSI_OPTIONS_FAST80
    設定しない場合、HBA は、バスを FAST80 SCSI モードで機能させません。

  SCSI_OPTIONS_FAST160
    設定しない場合、HBA は、バスを FAST160 SCSI モードで機能させません。

  SCSI_OPTIONS_FAST320
    設定しない場合、HBA は、バスを FAST320 SCSI モードで機能させません。

SCSI_OPTIONS_WIDE
設定しない場合、HBA は、バスをWIDE SCSI モードで機能させません。

SCSI_OPTIONS_SYNC
設定しない場合、HBA は、バスを同期転送モードで機能させません。

scsi-reset-delay
SCSI バスまたはデバイスのリセット回復時間 (ミリ秒)。

scsi-selection-timeout
デフォルトの SCSI 選択フェーズタイムアウト値 (ミリ秒)。HBA 固有の情報については、HBA の各マニュアルページを参照してください。

scsi_hba_detach() に関して：

scsi_hba_detach() は、DMA 限界値または属性の構造および HBA ドライバの指定されたインスタンスに対する転送ベクトルへの参照を除去します。

戻り値 scsi_hba_attach()、scsi_hba_attach_setup()、および scsi_hba_detach() は、関数の呼び出しに成功すると DDI_SUCCESS を返し、失敗すると DDI_FAILURE を返します。

コンテキスト scsi_hba_attach() および scsi_hba_attach_setup() は、attach(9E) から呼び出されます。scsi_hba_detach() は、detach(9E) から呼び出されます。

関連項目 attach(9E), detach(9E), tran_tgt_init(9E), ddi_prop_get_int(9F), ddi_dma_attr(9S), ddi_dma_lim(9S), dev_ops(9S), scsi_address(9S)、および scsi_hba_tran(9S)

Writing Device Drivers

注意事項 scsi_hba_detach() が呼び出された後に、SCSI ターゲットデバイスドライバの代わりに転送要求が行われないようにするのは HBA ドライバの役割です。

scsi_hba_attach() 関数は、古く、将来のリリースでは廃止されます。この関数は、scsi_hba_attach_setup() で置き換えられます。

名前

scsi_hba_attach_setup, scsi_hba_attach, scsi_hba_detach – SCSI HBA 接続および切り離しルーチン

形式

```
#include <sys/scsi/scsi.h>

int scsi_hba_attach_setup(dev_info_t *dip, ddi_dma_attr_t
    *hba_dma_attr, scsi_hba_tran_t *hba_tran, int hba_flags);

int scsi_hba_attach(dev_info_t *dip, ddi_dma_lim_t *hba_lim,
    scsi_hba_tran_t *hba_tran, int hba_flags, void *hba_options);

int scsi_hba_detach(dev_info_t *dip);
```

インタフェースレベル

Solaris アーキテクチャ固有 (Solaris DDI)。

パラメータ

dip	dev_info_t 構造に対するポインタで、HBA デバイスのインスタンスを参照します。
hba_lim	ddi_dma_lim(9S) 構造に対するポインタ。
hba_tran	scsi_hba_tran(9S) 構造に対するポインタ。
hba_flags	フラグ修飾子。唯一の定義済みフラグ値は、SCSI_HBA_TRAN_CLONE です。
hba_options	将来の拡張に備えて HBA ドライバによって提供されるオプションの機能で、NULL でなければなりません。
hba_dma_attr	ddi_dma_attr(9S) 構造に対するポインタ。

機能説明

scsi_hba_attach_setup() は、scsi_hba_attach() に代わる推奨インタフェースです。

scsi_hba_attach_setup() および scsi_hba_attach() に関して：

scsi_hba_attach() は、DMA 限界値 *hba_lim* と、*dip* によって定義される HBA デバイスの各インスタンスの転送ベクトル *hba_tran* を登録します。scsi_hba_attach_setup() は、DMA 属性 *hba_dma_attr* と、*dip* によって定義される HBA デバイスの各インスタンスの転送ベクトル *hba_tran* を登録します。HBA ドライバは、異なる DMA 限界値または DMA 属性、および必要に応じてデバイスの各インタフェースに対する転送ベクトルを渡すことができ、HBA 自身によって課せられたいかなる制約にも対応します。

scsi_hba_attach() および scsi_hba_attach_setup() は、dev_ops(9S) 構造内の dev_bus_ops フィールドを使用します。 HBA ドライバは、scsi_hba_attach() または scsi_hba_attach_setup() を呼び出す前に、このフィールドを NULL に初期化する必要があります。

SCSI_HBA_TRAN_CLONE が *hba_flags* で要求される場合、*hba_tran* 構造は、各ターゲットが HBA に接続されるたびに 1 回複製されます。構造の複製は、ターゲットを初期化するために tran_tgt_init(9E) エントリポイントが呼び出される前に行われます。後続のすべての HBA エントリポイント (tran_tgt_init(9E) も含めて) で、引数として渡された、または scsi_address 構造で検出された scsi_hba_tran_t

構造は、「複製された」 scsi_hba_tran_t 構造になるので、HBA は、scsi_hba_tran_t の tran_tgt_private フィールドを使用してターゲットごとのデータを指定できます。HBA は、切り離しの際に割り当てられたものと同じ scsi_hba_tran_t 構造だけを注意して解放する必要があります。システムによって割り当てられたすべての「複製された」 scsi_hba_tran_t 構造は、システムによって解放されます。

scsi_hba_attach() および scsi_hba_attach_setup() は、同じ名前の属性がすでにそのノードに接続されていない限り、多数の整数値属性を *dip* に接続します。HBA ドライバは、これらの構成パラメータを ddi_prop_get_int(9F) を使用して検出し、HBA を提供した機能に対するあらゆる設定に従う必要があります。

scsi-options
省略可能な SCSI 構成ビット

SCSI_OPTIONS_DR
設定しない場合、HBA はターゲットデバイスに切り離し (Disconnect) 特権を認可しません。

SCSI_OPTIONS_LINK
設定しない場合、HBA はリンクされたコマンド (Linked Commands) を有効にしません。

SCSI_OPTIONS_TAG
設定しない場合、HBA は、コマンドタグ付き待ち行列 (Command Tagged Queing) モードでは動作しません。

SCSI_OPTIONS_PARITY
設定しない場合、HBA はパリティーモードでは作動しません。

SCSI_OPTIONS_QAS
設定しない場合、HBA はクイックアービトレーション選択 (Quick Arbitration Select) 機能を利用しません。ご使用のマシンが QAS をサポートするかどうかは、サンのハードウェアマニュアルで確認してください。

SCSI_OPTIONS_FAST
設定しない場合、HBA は、バスを FAST SCSI モードで機能させません。

SCSI_OPTIONS_FAST20
設定しない場合、HBA は、バスを FAST20 SCSI モードで機能させません。

SCSI_OPTIONS_FAST40
設定しない場合、HBA は、バスを FAST40 SCSI モードで機能させません。

SCSI_OPTIONS_FAST80
設定しない場合、HBA は、バスを FAST80 SCSI モードで機能させません。

SCSI_OPTIONS_FAST160
設定しない場合、HBA は、バスを FAST160 SCSI モードで機能させません。

SCSI_OPTIONS_FAST320
設定しない場合、HBA は、バスを FAST320 SCSI モードで機能させません。

SCSI_OPTIONS_WIDE
設定しない場合、HBA は、バスをWIDE SCSI モードで機能させません。

SCSI_OPTIONS_SYNC
設定しない場合、HBA は、バスを同期転送モードで機能させません。

scsi-reset-delay
SCSI バスまたはデバイスのリセット回復時間 (ミリ秒)。

scsi-selection-timeout
デフォルトの SCSI 選択フェーズタイムアウト値 (ミリ秒)。HBA 固有の情報については、HBA の各マニュアルページを参照してください。

scsi_hba_detach() に関して：

scsi_hba_detach() は、DMA 限界値または属性の構造および HBA ドライバの指定されたインスタンスに対する転送ベクトルへの参照を除去します。

戻り値

scsi_hba_attach()、scsi_hba_attach_setup()、および scsi_hba_detach() は、関数の呼び出しに成功すると DDI_SUCCESS を返し、失敗すると DDI_FAILURE を返します。

コンテキスト

scsi_hba_attach() および scsi_hba_attach_setup() は、attach(9E) から呼び出されます。scsi_hba_detach() は、detach(9E) から呼び出されます。

関連項目

attach(9E), detach(9E), tran_tgt_init(9E), ddi_prop_get_int(9F), ddi_dma_attr(9S), ddi_dma_lim(9S), dev_ops(9S), scsi_address(9S)、および scsi_hba_tran(9S)

Writing Device Drivers

注意事項

scsi_hba_detach() が呼び出された後に、SCSI ターゲットデバイスドライバの代わりに転送要求が行われないようにするのは HBA ドライバの役割です。

scsi_hba_attach() 関数は、古く、将来のリリースでは廃止されます。この関数は、scsi_hba_attach_setup() で置き換えられます。

名前

scsi_hba_attach_setup, scsi_hba_attach, scsi_hba_detach – SCSI HBA 接続および切り離しルーチン

形式

```
#include <sys/scsi/scsi.h>

int scsi_hba_attach_setup(dev_info_t *dip, ddi_dma_attr_t
    *hba_dma_attr, scsi_hba_tran_t *hba_tran, int hba_flags);

int scsi_hba_attach(dev_info_t *dip, ddi_dma_lim_t *hba_lim,
    scsi_hba_tran_t *hba_tran, int hba_flags, void *hba_options);

int scsi_hba_detach(dev_info_t *dip);
```

インタフェースレベル

Solaris アーキテクチャ固有 (Solaris DDI)。

パラメータ

dip dev_info_t 構造に対するポインタで、HBA デバイスのインスタンスを参照します。

hba_lim ddi_dma_lim(9S) 構造に対するポインタ。

hba_tran scsi_hba_tran(9S) 構造に対するポインタ。

hba_flags フラグ修飾子。唯一の定義済みフラグ値は、SCSI_HBA_TRAN_CLONE です。

hba_options 将来の拡張に備えて HBA ドライバによって提供されるオプションの機能で、NULL でなければなりません。

hba_dma_attr ddi_dma_attr(9S) 構造に対するポインタ。

機能説明

scsi_hba_attach_setup() は、scsi_hba_attach() に代わる推奨インタフェースです。

scsi_hba_attach_setup() および scsi_hba_attach() に関して：

scsi_hba_attach() は、DMA 限界値 *hba_lim* と、*dip* によって定義される HBA デバイスの各インスタンスの転送ベクトル *hba_tran* を登録します。scsi_hba_attach_setup() は、DMA 属性 *hba_dma_attr* と、*dip* によって定義される HBA デバイスの各インスタンスの転送ベクトル *hba_tran* を登録します。HBA ドライバは、異なる DMA 限界値または DMA 属性、および必要に応じてデバイスの各インタフェースに対する転送ベクトルを渡すことができ、HBA 自身によって課せられたいかなる制約にも対応します。

scsi_hba_attach() および scsi_hba_attach_setup() は、dev_ops(9S) 構造内の dev_bus_ops フィールドを使用します。 HBA ドライバは、scsi_hba_attach() または scsi_hba_attach_setup() を呼び出す前に、このフィールドを NULL に初期化する必要があります。

SCSI_HBA_TRAN_CLONE が *hba_flags* で要求される場合、*hba_tran* 構造は、各ターゲットが HBA に接続されるたびに 1 回複製されます。構造の複製は、ターゲットを初期化するために tran_tgt_init(9E) エントリポイントが呼び出される前に行われます。後続のすべての HBA エントリポイント (tran_tgt_init(9E) も含めて) で、引数として渡された、または scsi_address 構造で検出された scsi_hba_tran_t

構造は、「複製された」 scsi_hba_tran_t 構造になるので、HBA は、scsi_hba_tran_t の tran_tgt_private フィールドを使用してターゲットごとのデータを指定できます。HBA は、切り離しの際に割り当てられたものと同じ scsi_hba_tran_t 構造だけを注意して解放する必要があります。システムによって割り当てられたすべての「複製された」 scsi_hba_tran_t 構造は、システムによって解放されます。

scsi_hba_attach() および scsi_hba_attach_setup() は、同じ名前の属性がすでにそのノードに接続されていない限り、多数の整数値属性を *dip* に接続します。HBA ドライバは、これらの構成パラメータを ddi_prop_get_int(9F) を使用して検出し、HBA を提供した機能に対するあらゆる設定に従う必要があります。

scsi-options
省略可能な SCSI 構成ビット

SCSI_OPTIONS_DR
設定しない場合、HBA はターゲットデバイスに切り離し (Disconnect) 特権を認可しません。

SCSI_OPTIONS_LINK
設定しない場合、HBA はリンクされたコマンド (Linked Commands) を有効にしません。

SCSI_OPTIONS_TAG
設定しない場合、HBA は、コマンドタグ付き待ち行列 (Command Tagged Queing) モードでは動作しません。

SCSI_OPTIONS_PARITY
設定しない場合、HBA はパリティーモードでは作動しません。

SCSI_OPTIONS_QAS
設定しない場合、HBA はクイックアービトレーション選択 (Quick Arbitration Select) 機能を利用しません。ご使用のマシンが QAS をサポートするかどうかは、サンのハードウェアマニュアルで確認してください。

SCSI_OPTIONS_FAST
設定しない場合、HBA は、バスを FAST SCSI モードで機能させません。

SCSI_OPTIONS_FAST20
設定しない場合、HBA は、バスを FAST20 SCSI モードで機能させません。

SCSI_OPTIONS_FAST40
設定しない場合、HBA は、バスを FAST40 SCSI モードで機能させません。

SCSI_OPTIONS_FAST80
設定しない場合、HBA は、バスを FAST80 SCSI モードで機能させません。

SCSI_OPTIONS_FAST160
設定しない場合、HBA は、バスを FAST160 SCSI モードで機能させません。

SCSI_OPTIONS_FAST320
設定しない場合、HBA は、バスを FAST320 SCSI モードで機能させません。

SCSI_OPTIONS_WIDE
設定しない場合、HBA は、バスをWIDE SCSI モードで機能させません。

SCSI_OPTIONS_SYNC
設定しない場合、HBA は、バスを同期転送モードで機能させません。

scsi-reset-delay
SCSI バスまたはデバイスのリセット回復時間 (ミリ秒)。

scsi-selection-timeout
デフォルトの SCSI 選択フェーズタイムアウト値 (ミリ秒)。HBA 固有の情報については、HBA の各マニュアルページを参照してください。

scsi_hba_detach() に関して：

scsi_hba_detach() は、DMA 限界値または属性の構造および HBA ドライバの指定されたインスタンスに対する転送ベクトルへの参照を除去します。

戻り値 scsi_hba_attach()、scsi_hba_attach_setup()、および scsi_hba_detach() は、関数の呼び出しに成功すると DDI_SUCCESS を返し、失敗すると DDI_FAILURE を返します。

コンテキスト scsi_hba_attach() および scsi_hba_attach_setup() は、attach(9E) から呼び出されます。scsi_hba_detach() は、detach(9E) から呼び出されます。

関連項目 attach(9E), detach(9E), tran_tgt_init(9E), ddi_prop_get_int(9F), ddi_dma_attr(9S), ddi_dma_lim(9S), dev_ops(9S), scsi_address(9S)、および scsi_hba_tran(9S)

Writing Device Drivers

注意事項 scsi_hba_detach() が呼び出された後に、SCSI ターゲットデバイスドライバの代わりに転送要求が行われないようにするのは HBA ドライバの役割です。

scsi_hba_attach() 関数は、古く、将来のリリースでは廃止されます。この関数は、scsi_hba_attach_setup() で置き換えられます。