



SunOS リファレンスマニュアル 7 : デバイスとネットワークインタ フェース

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

Part No: 816-3994-10
2002 年 5 月

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

Federal Acquisitions: Commercial Software-Government Users Subject to Standard License Terms and Conditions.

本製品に含まれる HG 明朝 L、HG-MincyoL-Sun、HG ゴシック B、および HG-GothicB-Sun は、株式会社リコーがリコービマジクス株式会社からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。HG 平成明朝体 W3@X12 は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、docs.sun.com、AnswerBook、AnswerBook2 は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。© Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政事業庁が公開したデータを元に制作された物です (一部データの加工を行なっています)。

本製品に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド '98』に添付のものを使用しています。© 1997 ビレッジセンター

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

DtComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *man pages section 7: Device and Network Interfaces*

Part No: 816-0222-10

Revision A



020324@2851



目次

はじめに 5

SunOS リファレンスマニュアル 7: デバイスとネットワークインタフェース 9

Intro(7) 10

afb(7d) 13

cpr(7) 14

cvc(7D) 16

cvcredir(7D) 17

ecpp(7D) 18

fas(7D) 25

ffb(7D) 34

ge(7D) 35

glm(7D) 39

hme(7D) 45

m64(7D) 50

pm(7D) 51

qfe(7d) 58

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 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。

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	<p>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 (<i>Italic</i>) 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。</p> <p>以下の文字がそれぞれの項目で使われています。</p> <p>[] 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。</p> <p>... 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...')。</p> <p> 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。</p> <p>{ } 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。</p>
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 7: デ バイスとネットワークインタフェース

Intro(7)

名前	Intro, intro – 特殊ファイルの序章
機能説明	<p>本セクションは、システム上で利用可能なさまざまなデバイスとネットワークインタフェースについて説明します。説明されているインタフェースの種類には、文字型デバイス、ブロック型デバイス、STREAMS モジュール、ネットワークプロトコル、ファイルシステム、ドライバサブシステムとドライバクラスの ioctl 要求、などがあります。</p> <p>本セクションは次のように分けられています。</p> <p>(7D) システムは、さまざまなハードウェアデバイス (磁気テープ、シリアル通信回線、マウス、フレームバッファ、疑似ターミナルや疑似ウィンドウなどの仮想デバイス) 用のドライバを提供します。</p> <p>特定のハードウェア周辺装置やデバイスドライバを参照する特殊ファイル、STREAMS デバイスドライバについて説明しています。ハードウェアデバイスおよびそれに対応するデバイスドライバの特徴についても、適宜説明しています。</p> <p>アプリケーションは、デバイスの特殊ファイルを介してデバイスにアクセスします。デバイスドライバを使用するために必要なアプリケーションプログラムインタフェース (API) 情報と共に、デバイスの特殊ファイルは、デバイスにアクセスするために使用されるものとして指定されています。</p> <p>デバイス特殊ファイルはすべて /devices ディレクトリに置かれています。/devices ディレクトリ階層は、システム上に構成されているシステムバス、コントローラ、デバイスの階層構造を反映しようとしません。/devices ディレクトリにある特殊ファイルの論理デバイス名は、/dev に置かれています。/devices に置かれているすべての特殊ファイルが、/dev ディレクトリに論理デバイス名を持っているわけではありませんが、アプリケーションは、可能な場合は論理デバイス名を使用してデバイスを参照します。論理デバイス名は、そのデバイスのマニュアルページの「ファイル」の項目に記載されています。</p> <p>セクション 7D には、該当する場合にはドライバ構成についても説明しています。多くのデバイスドライバには、<i>driver_name.conf</i> という形式のドライバ構成ファイルがあります (<i>driver.conf(4)</i> を参照)。ドライバ構成ファイルに記述されている構成情報は、ドライバやデバイスを構成するために使用されます。ドライバ構成ファイルは、ディレクトリ /kernel/drv および /usr/kernel/drv にあります。プラットフォームに依存するドライバの構成ファイルは、/platform/'uname -i '/kernel/drv ディレクトリにあります。'uname -i は、-i オプション付きの <i>uname(1)</i> コマンドの出力結果です。</p> <p>いくつかのドライバ構成ファイルには、ユーザーが構成できる属性が含まれています。ドライバ構成ファイルに加えた変更内容は、システムをリブートするかまたはドライバをいったん削除して再度追加すると、有効になります (<i>rem_drv(1M)</i> および <i>add_drv(1M)</i> を参照)。</p>

- (7FS) SunOS でサポートされている複数のファイルシステムのプログラムインタフェースについて説明します。
- (7I) クラスまたはサブドライバに適用される `ioctl` 要求について説明します。たとえば、大部分のテープデバイスに適用される `ioctl` 要求については、`mtio(7I)` で説明しています。特定のデバイスにのみ適用される `ioctl` 要求については、そのデバイスのマニュアルページで説明しています。そのようなデバイスのマニュアルページは、セクション 7I に記載されている `ioctl` 要求に対する例外についても説明しています。
- (7M) STREAMS モジュールについて説明します。STREAMS ドライバについてはセクション 7D で説明します。`streamio(7I)` には、STREAMS モジュールとインタフェースを STREAMS フレームワークで操作するために使用される `ioctl` 要求のリストが含まれています。STREAMS モジュールに固有の `ioctl` 要求については、そのモジュールのマニュアルページで説明しています。
- (7P) SunOS で利用できるさまざまなネットワークプロトコルについて説明しています。

SunOS は、ソケットに基づくネットワーク通信と、STREAMS に基づくネットワーク通信をサポートしています。`inet(7P)` に記述されているインターネットファミリは、システムがその他多数のプロトコルをサポートしている場合でも、SunOS でサポートされる主プロトコルです。`raw` インタフェースによって、パケット分割や再アセンブリ、ルーティング、アドレス指定、ソケットに基づく実装における基本転送など、下位レベルのサービスが提供されます。通常、インターネットファミリプロトコルを使用する通信機能にアクセスするには、ソケットを結合する時に `AF_INET` アドレスファミリを指定します。詳細は、`socket(3SOCKET)` を参照してください。

インターネットにおける主要プロトコルは、次のとおりです。

- インターネットプロトコル (IP; Internet Protocol)。共通のデータグラム形式をサポートします。詳細は `ip(7P)` で説明されています。これは、`AF_INET` ドメイン中の `SOCK_RAW` 型ソケットのデフォルトのプロトコルです。
- 伝送制御プロトコル (TCP; Transmission Control Protocol)。 `tcp(7P)` を参照してください。これは、`SOCK_STREAM` 型ソケットのデフォルトのプロトコルです。
- ユーザーデータグラムプロトコル (UDP; User Datagram Protocol)。 `udp(7P)` を参照してください。これは、`SOCK_DGRAM` 型ソケットのデフォルトのプロトコルです。
- アドレス解決プロトコル (ARP; Address Resolution Protocol)。 `arp(7P)` を参照してください。
- インターネット制御メッセージプロトコル (ICMP; Internet Control Message Protocol)。 `icmp(7P)` を参照してください。

Intro(7)

関連項目 `add_drv(1M)`, `rem_drv(1M)`, `intro(3)`, `ioctl(2)`, `socket(3SOCKET)`,
`driver.conf(4)`, `arp(7P)`, `icmp(7P)`, `inet(7P)`, `ip(7P)`, `mtio(7I)`, `st(7D)`,
`streamio(7I)`, `tcp(7P)`, `udp(7P)`

『Solaris のシステム管理 (IP サービス)』

『STREAMS Programming Guide』

『Writing Device Drivers』

コマンド一覧

名前	説明
Intro(7)	特殊ファイルの序章
afb(7d)	Elite3D グラフィックスアクセラレータドライバ
cpr(7)	保存停止および復元再開モジュール
cvc(7D)	仮想コンソールドライバ
cvcdir(7D)	仮想コンソール入出力先変更ドライバ
ecpp(7D)	IEEE 1284 互換の平行ポートドライバ
fas(7D)	FAS SCSI ホストバスアダプタドライバ
ffb(7D)	24 ビット UPA カラーフレームバッファとグラフィックスアクセラレータ
ge(7D)	GEM GigabitEthernet デバイスドライバ
glm(7D)	GLM SCSI ホストバスアダプタドライバ
hme(7D)	SUNW,hme Fast-Ethernet デバイスドライバ
m64(7D)	PGX、PGX24、PGX64 フレームバッファデバイスドライバ
pm(7D)	電源管理システムのドライバ
qfe(7d)	SUNW,qfe Quad Fast-Ethernet デバイスドライバ

名前	afb – Elite3D グラフィックスアクセラレータドライバ
機能説明	afb ドライバは、Sun Elite3D グラフィックスアクセラレータのデバイスドライバです。afbdaemon プロセスは、システムの起動時と、一時停止・再開サイクルの再開処理中に、afb マイクロコードを読み込みます。
ファイル	/dev/fbs/afb <i>n</i> デバイス特殊ファイル /usr/lib/afb.unicode afb マイクロコード /usr/sbin/afbdaemon afb マイクロコード読み込みプログラム
関連項目	afbconfig(1M)

cpr(7)

名前	cpr - 保存停止および復元再開モジュール
形式	/kernel/misc/cpr
機能説明	<p>cpr は、システム全体を保存停止したり復元再開したりするための読み込み可能なモジュールです。電力の節約や移動の目的で一時的に電源を切断するために、システムの保存停止が必要になる場合があります。ハードウェアの再構成や交換を行う場合は、cpr モジュールを使用せずに通常の停止を行ってください。復元再開操作が成功するには、ハードウェア構成が同じままであることが重要です。システムが保存停止されるときに、システム全体の状態は、復元再開が実行されるまで揮発性の記憶装置に保存されます。</p> <p>タイプ 5 キーボードの電源制御キーと Shift-電源制御キーの操作は、cpr モジュールを使用します。システムにインストールされていて、このモジュールを使用する可能性があるユーティリティは、uadmin(1M) と uadmin(2) の 2 つです。</p> <p>cpr モジュールは、システムを保存停止するときに次の処理を実行します。まず、SIGFREEZE 信号がすべてのユーザースレッドに送られ、その後、それらのスレッドは停止されます。マルチプロセッサシステムの場合は、システムを単一プロセッサモードにします。次に、ユーザーのダーティーページは、バックアップ用の記憶装置にスワップアウトされ、すべてのファイルシステムは同期 (sync) が取られます。すべての装置は活動停止状態になり、システム割り込みは無効になります。システムの保存停止を完了するために、カーネルのメモリーページと残りのユーザーページは、圧縮形式でルートファイルシステムに書き込まれます。</p> <p>再びシステムの電源が入れると、基本的に、保存停止の処理手順と逆の処理が実行されます。カーネルイメージは、起動プログラムの /cprboot によってルートファイルシステムから復元されます。割り込みや装置は、以前の状態に復元されます。最後に、ユーザースレッドは再スケジュールされ、システムの復元再開を待つプロセスに SIGTHAW を伝送して通知します。他のプロセッサがある場合は、復元され、オンラインになります。これで、システムは完全に保存停止前の状態に戻ります。</p> <p>いくつかの場合において、cpr モジュールが保存停止操作を実行することができない場合があります。出荷時の標準構成以外の追加装置がある場合には、追加装置が cpr に対応していない可能性があります。この場合は、保存停止が失敗し、エラーメッセージが出力されます。保存停止が成功するようにするには、装置を取り除くか、またはそのデバイスドライバを読み込み解除する必要があります。装置の製造元に連絡し、cpr に対応しているバージョンのデバイスドライバを取得してください。保存停止は、装置またはプロセスが、リアルタイム処理のような、余裕のない、または時間に関係する処理を行っている場合も失敗します。この場合は、システムは現在の実行状態のままとなります。失敗を通知するメッセージがコンソールに表示され、状態は呼び出し側に返されます。いったんシステムが正常に保存停止されると、ハードウェアの再構成など外部からの影響がないかぎり、復元再開操作は必ず成功します。</p> <p>一部のネットワークベースのアプリケーションは、保存停止・復元再開処理の際に失敗する場合があります。この問題は、主に背後のネットワークプロトコルと関係するアプリケーションに依存します。一般的に、再試行して自動的に接続を確立するアプリケーションは、復元再開の際に何事もなかったかのように動作を継続します。再試行して自動的に接続を確立しないアプリケーションは、多くの場合失敗します。</p>

cvc(7D)

名前	cvc - 仮想コンソールドライバ						
機能説明	<p>cvc 仮想コンソールドライバは、ネットワークコンソールを利用可能にする STREAMS ベースの擬似ドライバです。cvc へのドライバインタフェースは console(7D) です。</p> <p>cvc ドライバは、論理的に console ドライバの下位に位置するドライバです。cvc は、ネットワークコンソール接続が動作中の場合に、コンソール出力を cvcredir(7D) ドライバに切り替えます。ネットワークコンソール接続が動作していない場合は、cvc はコンソールの出力先を内部ハードウェアのインタフェースに変更します。</p> <p>cvc ドライバは、cvcredir と内部ハードウェアからのコンソール入力を受け取り、/dev/console に結び付けられているプロセスに渡します。</p>						
注意事項	<p>cvc の機能は、SunOS の wscons(7D) の機能に取って代わるものです。この両方を、同時に使用しないでください。wscons ドライバは、直接に接続されたコンソール (フレームバッファやキーボード) を持つシステムで有用ですが、ローカルのキーボードやフレームバッファを持たない、cvc を使用するプラットフォームでは有用ではありません。</p>						
属性	<p>以下の属性については、attributes(5) を参照してください。</p> <table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>アーキテクチャ</td> <td>Sun Enterprise 10000 サーバー、Sun Fire 15000 サーバー</td> </tr> <tr> <td>使用条件</td> <td>SUNWcvc.u</td> </tr> </tbody> </table>	属性タイプ	属性値	アーキテクチャ	Sun Enterprise 10000 サーバー、Sun Fire 15000 サーバー	使用条件	SUNWcvc.u
属性タイプ	属性値						
アーキテクチャ	Sun Enterprise 10000 サーバー、Sun Fire 15000 サーバー						
使用条件	SUNWcvc.u						
関連項目	<p>cvcd(1M), attributes(5), console(7D), cvcredir(7D), wscons(7D)</p> <p>『Sun Enterprise 10000 SSP リファレンスマニュアル』</p> <p>『System Management Services (SMS) リファレンスマニュアル』</p>						

名前 cvcredir – 仮想コンソール入出力先変更ドライバ

機能説明 cvcredir 仮想コンソール入出力先変更ドライバは、一部のプラットフォームで提供されているネットワークコンソールをサポートする、STREAMS ベースの疑似ドライバです。cvcredir ドライバは、仮想コンソールドライバ cvc(7D) および仮想コンソールデーモン cvcd(1M) と連携して機能します。

cvcredir ドライバは、cvc からのコンソール出力を受け取って cvcd に渡します。また、cvcd からのコンソール入力を受け取って cvc に渡します。

属性 以下の属性については、attributes(5) を参照してください。

属性タイプ	属性値
アーキテクチャ	Sun Enterprise 10000 サーバー、Sun Fire 15000 サーバー
使用条件	SUNWcvc.u

関連項目 cvcd(1M), attributes(5), console(7D), cvc(7D)

『Sun Enterprise 10000 SSP リファレンスマニュアル』

『System Management Services (SMS) リファレンスマニュアル』

ecpp(7D)

名前	ecpp – IEEE 1284 互換の平行ポートドライバ
形式	<pre>#include <sys/types.h> #include <sys/ecppio.h> ecpp@unit-address</pre>
機能説明	<p>ecpp ドライバは、IEEE 1284 に準拠したデバイスに対して双方向のインタフェース、セントロニクスデバイスに対して単方向の順方向インタフェースを提供します。ecpp ドライバは、IEEE 1284 互換プロトコル、ニブルプロトコル、ECP プロトコル、およびセントロニクスプロトコルをサポートしています。ECPP_COMPAT_MODE および ECPP_CENTRONICS の動作モードは、論理的に同じであるハンドシェイク方式のプロトコルです。ただし、ECPP_COMPAT_MODE をサポートしているデバイスは、IEEE 1284 に準拠したデバイスです。IEEE 1284 に準拠したデバイスは、少なくとも ECPP_COMPAT_MODE および ECPP_NIBBLE_MODE をサポートしています。セントロニクスデバイスは、ECPP_CENTRONICS モードをサポートします。</p> <p>デフォルトでは、ECPP_COMPAT_MODE デバイスのハンドシェイク方式のストロブパルス幅は、500 ナノセカンドです。このモードの場合、データの順方向転送は DMA によって行われます。デフォルトでは、ECPP_CENTRONICS デバイスのストロブパルス幅は、2 マイクロセカンドです。これらのデバイスでは、順方向転送は PIO を介して行われます。ECPP_COMPAT_MODE および ECPP_CENTRONICS デバイスのデフォルトの動作は、ecpp.conf で定義された調整可能な変数によって変更可能です。</p> <p>ecpp ドライバは、「排他的に使用する」デバイスです。すなわち、デバイスがすでに開いている場合は、その後に関しても EBUSY によって失敗します。</p>
デフォルトの動作	<p>ecpp デバイスが開かれるたびに、そのデバイスは EBUSY としてマークされ、設定変数はデフォルト値に設定されます。write_timeout は、90 秒に設定されます。</p> <p>ecpp ドライバは、次に説明するアルゴリズムに従って、モード変数を設定します。まず、ecpp ドライバは、open(2) が実行されている間に ECPP_ECP_MODE モードへの移行を試みます。これに失敗すると、ドライバは、モードを ECPP_NIBBLE_MODE モードにするようにデバイスとネゴシエーションを行います。このネゴシエーションに失敗すると、ドライバは ECPP_CENTRONICS モードで動作します。デバイスを開くのに成功すると、IEEE 1284 に準拠するデバイスは、ECPP_ECP_MODE または ECPP_NIBBLE_MODE のいずれかで逆方向アイドルフェーズのアイドル状態になります。その後 write(2) 呼び出しにより、ドライバは ECPP_ECP_MODE または ECPP_COMPAT_MODE のいずれかの順方向リンクフェーズに移動させます。この転送が終了するとリンクはアイドル状態に戻ります。</p> <p>アプリケーションは、ECPPIOC_SETPARMS ioctl(2) の呼び出しによって、デバイスを特定のモードにするようにネゴシエーションを行ったり write_timeout 値を設定したりすることができます。ネゴシエーションが成功するには、要求されたモードにホストワークステーションと周辺デバイスの両方が対応している必要があります。</p>

調整	<p>ecpp ドライバの特徴は、<code>/kernel/drv/ecpp.conf</code> に記述された変数で調整可能です。これらの変数は、システム起動中にドライバによって読み込まれます。変数を調整するには、<code>ecpp.conf</code> ファイルを編集し、<code>update_drv(1M)</code> を起動してカーネルにファイルを再度読み込ませます。</p> <p>セントロニクス周辺機器および特定の <i>IEEE 1284</i> に準拠した周辺機器は、高速ハンドシェイキングモードで動作するパラレルポートと共には動作しません。印刷に関する問題が発生した場合は、<code>"fast-centronics"</code> および <code>"fast-1284-compatible"</code> を <code>"false"</code> に設定します。詳細は、<code>/kernel/drv/ecpp.conf</code> を参照してください。</p>
読み取り・書き込み操作	<p>ecpp ドライバは、全二重の STREAMS デバイスドライバです。アプリケーションが <i>IEEE 1284</i> に準拠するデバイスに書き込んでいる間、別のスレッドはそのデバイスから読み取ることができます。</p>
書き込み操作	<p><code>write(2)</code> 操作は、ストリームヘッドに正常に書き込まれたバイト数を返します。セントロニクスデバイスがデータを転送している間に異常が発生した場合は、状態ビットの内容はエラーが発生した時のものが保存されます。アプリケーションプログラムは、この状態ビットを <code>BPPIOC_GETERR ioctl(2)</code> 呼び出しを使用して取り出すことができます。取り出された状態情報は、転送の試行または <code>BPPIOC_TESTIO ioctl(2)</code> が発生するたびに書き換えられます。</p>
読み取り操作	<p><code>read(2)</code> の実行中に失敗、またはエラー条件が発生した場合には、正常に読み取られたバイト数が戻されます (不十分な読み取り)。利用可能なデータがないポートを読み取ろうとすると、<code>read(2)</code> は、<code>O_NDELAY</code> が設定されている場合には、<code>0</code> を返します。<code>O_NONBLOCK</code> が設定されている場合には、<code>read(2)</code> は、<code>-1</code> を返し、<code>errno</code> に <code>EAGAIN</code> を設定します。<code>O_NDELAY</code> と <code>O_NONBLOCK</code> が設定されていない場合、<code>read(2)</code> は、データが用意されるまで待ちます。</p>
IOCTLS	<p>以下の <code>ioctl(2)</code> 呼び出しを使用することができます。ecpp がデータを転送する場合、ドライバは、<code>ioctl(2)</code> 呼び出しが処理される前にデータがデバイスに送られるまで待機します。</p> <p>ecpp ドライバでは、<code>prnio(7I)</code> インタフェースが使用できます。</p> <p>注 - <code>PRNIOC_RESET</code> コマンドは、<code>nInit</code> シグナルを 2 ミリ秒間切り替えた後、デフォルトのネゴシエーションを行います。</p> <p>次の <code>ioctl(2)</code> 呼び出しは、下位互換性を維持するためにのみ残されています。そのため、新たに作成するアプリケーションでは使用しないことをお勧めします。</p> <p><code>ECPPIOC_GETPARMS</code> 現在の転送パラメータを獲得します。引数は、<code>struct ecpp_transfer_parms</code> を指すポインタです。この構造体の要素については、以下を参照してください。デバイスが開かれてから設定されたパラメータがない場合には、この構造体にはデフォルトの値が設定されません。上記の「デフォルトの動作」を参照してください。</p> <p><code>ECPPIOC_SETPARMS</code> 転送パラメータを設定します。引数は、<code>struct ecpp_transfer_parms</code> を指すポインタです。設定された</p>

パラメタが範囲外の場合には、EINVAL が返されま
す。周辺デバイスまたはホストデバイスが要求された
モードに対応していない場合は、EPROTONOSUPPORT
が返されます。ecpp_transfer_parms とその有効な
パラメタについては、以下を参照してください。

転送パラメタ構造体は、<sys/ecppio.h> で定義され
ています。

```
struct ecpp_transfer_parms {
    int write_timeout;
    int mode;
};
```

write_timeout フィールドは、ecpp.conf に指定され
ている ecpp-transfer-timeout の値 に設定され
ます。write_timeout フィールドは、転送要求に対
して周辺デバイスが応答するまでドライバが待つ時間
を指定します。この値は、0 より大きく、ECPP_MAX
_TIMEOUT より小さく指定します。これ以外の値は、
範囲外の値となります。

mode フィールドは、現在のパラレルポートが設定され
ている IEEE 1284 モードを反映します。このモードの
値は、以下の ECPP_CENTRONICS、ECPP_COMPAT
_MODE、ECPP_NIBBLE_MODE、ECPP_ECP_MODE のい
ずれかに設定することができされます。他のすべての
値は無効です。要求されたモードに対応していない場
合には、ECPPIOC_SETPARMS は EPROTONOSUPPORT
を返します。またこのモードは、ECPP_CENTRONICS
モードに設定されたあと、ECPPIOC_SETPARMS は、
アプリケーションを元のモードに戻します。

ECPPIOC_GETDEVID

この ioctl コマンドは、指定されたモードで周辺デバイ
スから IEEE 1284 デバイスの ID を獲得します。現在、
デバイスの ID は、ニブルモードでしか獲得できませ
ん。<sys/ecppsys.h> で定義された構造体へのポイ
ントは、引数として渡される必要があります。

1284 デバイスの ID の構造体は以下のようになりま
す。

```
struct ecpp_device_id {
    int mode; /* デバイスの ID の読み取りに使用されるモード */
    int len; /* バッファの長さ */
    int rlen; /* デバイスの ID の文字列の実際の長さ */
    char *addr; /* バッファのアドレス */
};
```

ecpp(7D)

モードは *IEEE 1284* のモードのことで、このモードでデバイスの ID 情報を取り出すためのネゴシエーションがポートに対して行われます。周辺デバイスまたはホストがこのモードをサポートしていない場合、`EPROTONOSUPPORT` が返されます。アプリケーションは、モードを `ECPP_NIBBLE_MODE` に設定します。len は、addr が指すバッファの長さです。rlen は、周辺デバイスから返されるデバイスの ID 文字列の実際の長さを表します。返された rlen が len よりも長い場合、アプリケーションは、再度 `ECPPIOC_GETDEVID` を呼び出すことができます。その際、バッファの長さは rlen と等しいかそれより長くなります。*IEEE 1284* デバイスの ID の 2 バイトの長さは、考慮にはいれていないため、ユーザーのバッファでは返されないことに注意してください。

`ECPPIOC_GETDEVID` の実行に成功したら、ドライバは `ECPP_COMPAT_MODE` にリンクを返します。アプリケーションは、リンクが機能していた以前のモードを判定し、そのモードにリンクを戻します。

BPPIOC_TESTIO

セントロニクスまたは互換モードでの周辺デバイスの操作について、順方向転送の準備ができていないかをテストします。

TESTIO は、オープンフラグとセントロニクスのステータス信号を検査することにより周辺デバイスがデータを受け取る準備ができていないかを判定します。デバイスの現在のモードが、`ECPP_NIBBLE_MODE` の場合、ドライバは `ECPP_COMPAT_MODE` モードへの移行を試みて、ステータス信号を検査してから、`ECPP_NIBBLE_MODE` モードに戻します。現在のモードが、`ECPP_CENTRONICS` または `ECPP_COMPAT_MODE` の場合、TESTIO は現在のモードでセントロニクスステータス信号を検査します。データを受け取るには、デバイスは、`nErr` と `Select` 信号は立っている必要がありますが、`PE` と `Busy` 信号は立っていることは許されません。ecpp がデータを転送する場合、TESTIO は、TESTIO を実行する前にドライバに送られた以前のデータが配信されるまで待機しています。しかし、TESTIO が待機している間にエラーが起こった場合、TESTIO は即座にエラーを返します。TESTIO が、状況が良いと判定した場合は、0 が返されます。しかし、-1 が返された場合は、`errno` が、`EIO` に設定され、状態ピンの状況が捕らえられます。この状況は、`BPPIOC_GETERR ioctl(2)` 呼び出しにより取り出すことができます。 `timeout_occurred` と `bus_error`

BPPIOC_GETERR	<p>フィールドは、このioctl(2)により設定されることはありません。BPPIOC_TESTIOとBPPIOC_GETERRは、bpp(7D)に指定されているioctlと互換性があります。</p> <p>最後のエラー状態を取り出します。引数は、<sys/bpp_io.h>ヘッダーファイルに定義されているstruct bpp_error_statusを指すポインタです。エラーステータスの構造体は以下のようになります。</p> <pre> struct bpp_error_status { char timeout_occurred; /* 1= タイムアウト */ char bus_error; /* 使用されない */ uchar_t pin_status; /* エラーを起こす可能性の /* あるピンの状態 */ }; </pre> <p>pin_statusフィールドには、起こりうるエラーの条件を示します。pin_statusの有効なビットは、BPP_ERR_ERR、BPP_SLCT_ERR、BPP_PE_ERR、BPP_BUSY_ERRです。設定されたビットは、関連するピンが何であるかを示します。</p> <p>この構造体は、write(2)呼び出し時の最新のエラー条件のすべての適切な状態ビット、または最新のBPPIOC_TESTIO ioctl(2)呼び出し時のビットの状態を示します。</p> <p>pin_statusは、ECPP_CENTRONICSまたはECPP_COMPAT_MODEで発生する可能性があるエラー条件を示します。これらのモードでは、状態ピンは、デバイスの状態を示します。たとえば、多くのセントロニクスプリンタでは、紙詰まりの発生時にnErrの信号レベルを下げます。状態ピンの動作は、デバイスによって異なります。追加状態情報は、逆方向チャンネルを通して返されます。</p> <p>timeout_occurredの値は、write(2)の呼び出し中に時間切れが発生した時に設定されます。bus_errorは、このインタフェースでは使用されません。</p>
IOCTLS	<p>以下のioctlは、パラレルポートのステータスを直接読み込んで書き込み、信号を制御するために使用されます。デバイスの現在のモードが、ECPP_ECP_MODEまたはECPP_NIBBLE_MODEの場合、ドライバはECPP_COMPAT_MODEにリンクするようにネゴシエーションを行い、レジスタの値を取得または設定し、リンクをECPP_NIBBLE_MODEに戻します。現在のモードが、ECPP_CENTRONICSまたはECPP_COMPAT_MODEの場合、これらのioctlは現在のモードでレジスタの値を獲得または設定します。</p>

ECPPIOC_GETREGS

レジスタ値を読み取ります。引数は、struct ecpp_regs 構造体へのポインタです。この構造体については、以下の記述を参照してください。

ECPPIOC_SETREGS

ecpp レジスタ値を設定します。引数は、struct ecpp_regs へのポインタです。この構造体については、以下の記述を参照してください。パラメータが範囲外の場合は、EINVAL が返されます。

ポートレジスタ構造体は、<sys/ecppio.h> に定義されています。

```
struct ecpp_regs {
    uchar    dsr; /* status reg */
    u_char   dcr; /* control reg */
};
```

ステータスレジスタは、読み取り専用です。ECPPIOC_SETREGS ioctl は、このレジスタには影響を与えません。dsr の有効なビット値は、ECPP_nER、ECPP_SLCT、ECPP_PE、ECPP_nACK、ECPP_nBUSY です。ほかのすべてのビットは、予約されていて、常に 1 を返します。

制御レジスタは、読み取りと書き取りができます。dcr の有効なビット値は、ECPP_STB、ECPP_AFX、ECPP_nINIT、ECPP_SLCTIN です。ほかのすべてのビットは予約されています。予約されているビットを読み取ると、常に 1 を返します。これらのビットに 0 を書き込もうとすると、EINVAL が返されます。

デバイスの特殊
ファイル

/dev/ecppN

/dev/printersN

1284 に準拠するパラレルポートデバイスの特殊ファイルは、両方のネームスペースに存在します。

ファイル

kernel/drv/ecpp 32 ビット ELF カーネルモジュール

kernel/drv/sparcv9/ecpp64 ビット ELF カーネルモジュール

kernel/drv/ecpp.conf ドライバ構成ファイル

エラー

EBADF 書き込み専用で開いたデバイスに対して、読み取りが行われようとして、または、読み取り専用で開いたデバイスに対して、書き込みが行われようとして、

EBUSY すでに開かれているデバイスが開かれようとして、デバイスの 1 つが開いているときに、ドライバを読み込み解除しようとして、

EINVAL ecpp_transfer_params 構造体の中で ECPPIOC_SETPARMS ioctl() が範囲外の値で呼び出されました。ecpp_regs 構造体の中で ECPPIOC_SETREGS ioctl() が無効な値で呼び出されました。modload(1M) または modunload(1M) の実行中に、無効なコマンド引数を受け取りました。

EIO アクセス時にドライバがバスエラーを検出しました。読み取り処理または書き込み処理が、周辺デバイスのエラーまたは転送の時間切れのために、正常に終了しませんでした。

ecpp(7D)

ENXIO ドライバが、接続ができないデバイスを開く要求を受け付けました。ドライバが、障害のある周辺デバイスに対する書き込み処理を受け取りました。

属性 以下の属性については、attributes(5) を参照してください。

属性タイプ	属性値
アーキテクチャ	PCI ベースのシステムに限定
使用条件	SUNWpd
インタフェースの安定性	Evolving

関連項目 modload(1M), modunload(1M), update_drv(1M), ioctl(2), open(2), read(2), write(2), attributes(5), bpp(7D), usbprn(7D), prnio(7I), streamio(7I)

IEEE Std 1284-1994

診断 パラレルポートのコントローラが使用できない
 ドライバは、指定されたホストでパラレルポートのコントローラをサポートしていません。接続が失敗しました。

名前	fas – FAS SCSI ホストバスアダプタドライバ
形式	fas@sbus-slot, 0x8800000
機能説明	<p>fas ホストバスアダプタドライバは、Qlogic FAS366 SCSI チップに対応した SCSI 準拠のネクサスドライバです。</p> <p>fas ドライバは、SCSA インタフェースが提供する標準機能に対応しています。ドライバは、タグ付きおよびタグなしの待ち行列化、Fast/Wide SCSI、ほとんど制限のない転送サイズ (移動 DVMA ウィンドウ法による)、自動要求検知に対応しています。ただし、リンクされたコマンドには対応していません。</p>
ドライバの設定	<p>fas ドライバは、広域的な SCSI 設定に優先する属性を fas.conf に定義することによって設定することができます。使用することができる属性は、scsi-options、target<n>-scsi-options、scsi-reset-delay、scsi-watchdog-tick、scsi-tag-age-limit、scsi-initiator-id です。</p> <p>target<n>-scsi-options は target<n> の scsi-options 属性に優先します。<n> には、10 進数の 0 ~ 15 の値を指定することができます。使用することができる scsi オプションは、SCSI_OPTIONS_DR、SCSI_OPTIONS_SYNC、SCSI_OPTIONS_TAG、SCSI_OPTIONS_FAST、SCSI_OPTIONS_WIDE です。</p> <p>周期的な時間間隔である scsi-watchdog-tick が経過する度に、fas ドライバは、現在実行中および切り離されたすべてのコマンドを調べ、時間切れかどうかを確認します。</p> <p>scsi-tag-age-limit は、fas ドライバがすべてのタグ ID を一巡して調べた後に、現在使用されている特定のタグ ID の割り当てを試みる回数です。同一のタグ ID が scsi-tag-age-limit の回数だけ使用されると、すべての未処理のコマンドが完了するか、時間切れになるまで、このターゲットにコマンドは発行されません。</p> <p>詳細については、scsi_hba_attach(9F) を参照してください。</p>
使用例	<p>例 1 fas 設定ファイルの例</p> <p>ファイル /kernel/drv/fas.conf を作成して、以下の行を追加してください。</p> <pre>scsi-options=0x78;</pre> <p>これは、すべての fas インスタンスに対して、タグ付き待ち行列化、Fast SCSI、Wide モードを無効にします。以下の例は、1 つの特定の fas のオプションを無効にします。詳細については、(driver.conf(4) を参照してください。</p> <pre>name="fas" parent="/iommu@f,e0000000/sbus@f,e0001000" reg=3,0x8800000,0x10,3,0x8810000,0x40 target1-scsi-options=0x58 scsi-options=0x178 scsi-initiator-id=6;</pre> <p>OBP のデフォルトのイニシエータ ID は 7 ですが、接続時に 6 に変更されます。OBP のイニシエータ ID を変更してください。</p> <p>上記の例では、ターゲット 1 の scsi-options を 0x58 に設定し、この SCSI バス上にある他のすべてのターゲットは 0x178 に設定しています。</p>

fas(7D)

親の物理パス名は、/devices ツリーを使用するか、論理デバイス名のリンクをたどることによって判定することができます。

```
# ls -l /dev/rdisk/c1t3d0s0
lrwxrwxrwx 1 root other 78 Aug 28 16:05 /dev/rdisk/c1t3d0s0 ->
. . / . . /devices/iommu@f,e0000000\
      sbus@f,e0001000/SUNW,fas@3,8800000/sd@3,0:a,raw
```

レジスタの属性値は、prtconf(1M) の出力 (-v オプションを使用)によって判定されます。

```
SUNW,fas, instance #0
```

```
Register Specifications:
```

```
Bus Type=0x3, Address=0x8800000, Size=10
```

Bus Type=0x3, Address=0x8810000, Size=40 また、デバイス照会文字列を使用することによって、scsi-options をデバイスごとに指定することができます。同一の照会文字列を持つすべてのデバイスには、同一の scsi-options が設定されます。これを利用して、同種のすべてのデバイスに対して、いくつかの scsi-options を無効にすることができます。

```
device-type-scsi-options-list=
```

```
"TOSHIBA XM5701TASUN12XCD", "cd-scsi-options";
```

cd-scsi-options = 0x0;/kernel/drv/fas.conf の中の上記のエントリは、照会文字列が TOSHIBA XM5701TASUN12XCD のすべてのデバイスの scsi-options を cd-scsi-options に設定します。照会文字列を調べるには、システムを起動する前に ok プロンプトに対して probe-scsi または probe-scsi-all コマンドを実行します。

scsi-options をターゲットごとに設定するには、以下のように入力します。

```
target1-scsi-options=0x78;
device-type-scsi-options-list =
    "SEAGATE ST32550W", "seagate-scsi-options" ;
seagate-scsi-options = 0x58;
scsi-options=0x3f8;
```

この例は、ターゲット1の scsi-options を 0x78 に設定し、ある特定の種類のディスクの scsi-options を 0x58 に設定し、この SCSI バス上にある他のすべてのターゲットの scsi-options を 0x3f8 に設定します。

ターゲット ID ごとに指定された scsi-options は、最も優先順位が高くなります。その次に優先順位が高いのは、デバイスの種類ごとに指定された scsi-options です。バスごとに指定される広域的な fas scsi-options (すべてのインスタンスに影響します) は、最も優先順位が低くなります。

指定された scsi-options が有効になるには、システムを再起動する必要があります。

ドライバの機能

ターゲットドライバがドライバの機能を有効にするには、fasドライバの機能を設定する必要があります。ターゲットドライバは、これらの機能の照会および変更ができます。照会および変更ができる機能には、synchronous、tagged-qing、wide-xfer、auto-rqsense、qfull-retries、qfull-retry-intervalがあります。他のすべての機能は、照会のみ行うことができます。

デフォルトでは、tagged-qing、auto-rqsense、wide-xfer 機能は無効で、disconnect、synchronous、untagged-qing 機能は有効です。これらの機能には、2進数の値(0または1)のみを設定することができます。qfull-retriesのデフォルト値は10で、qfull-retry-intervalのデフォルト値は100です。qfull-retries 機能は uchar_t 型(0～255)で、qfull-retry-interval は u_short 型(0～65535)です。

ターゲットドライバは、tagged-qing および wide-xfer を明示的に有効にする必要があります。tagged-qing が無効になっていても、fas がコマンドを待ち行列に入れることができるため、untagged-qing 機能は常に有効で、その値を変更することはできません。

scsi-options の値と機能が矛盾している場合は、scsi-options に設定されている値が常に優先されます。scsi_ifsetcap(9F) 呼び出しでは、whom != 0 のみを使用することができます。

詳細については、scsi_ifsetcap(9F) および scsi_ifgetcap(9F)を参照してください。

ファイル

/kernel/drv/fas ELF カーネルモジュール
/kernel/drv/fas.conf 省略可能な設定ファイル

属性

以下の属性については、attributes(5)を参照してください。

属性の種類	属性の値
Architecture	FAS366 ベースの SCSI ポートおよび SunSWIFT SBus SCSI ホストアダプタ/Fast Ethernet を持つ SPARC SBus ベースのシステムに限定

関連項目

prtconf(1M), driver.conf(4), attributes(5), scsi_abort(9F), scsi_hba_attach(9F), scsi_ifgetcap(9F), scsi_ifsetcap(9F), scsi_reset(9F), scsi_sync_pkt(9F), scsi_transport(9F), scsi_device(9S), scsi_extended_sense(9S), scsi_inquiry(9S), scsi_pkt(9S)

『Writing Device Drivers』

『ANSI Small Computer System Interface-2 (SCSI-2)』

『FAS366 Technical Manuals』 (QLogic Corporation 刊)

診断

以下で説明するメッセージは、コンソールに表示されると同時にログファイルに記録されます。

fas(7D)

最初の 5 つは、fas ドライバの接続処理中に表示されるメッセージです。これらの 5 つのメッセージはすべて、fas ドライバが接続処理に失敗したことを意味します。これらのメッセージの前に、「fas%d」が表示されます。「%d」は、fas コントローラのインスタンス番号です。

Device in slave-only slot, unused

SBus デバイスがスレーブ専用のスロットに配置されたため、アクセスすることができません。SBus デバイスをスレーブ以外のスロットに移動してください。

Device is using a hilevel intr

fas ドライバが使用することができない割り込みレベルでデバイスが設定されています。SBus デバイスを調べてください。

Cannot allocate soft state

Cannot alloc dma handle

Cannot alloc cmd area

Cannot create kmem_cache

ドライバが内部データ構造に対して記憶域を割り当てることができません。

Unable to map FAS366 registers

ドライバがデバイスレジスタを割り当てることができません。問題のあるハードウェアを調べてください。ドライバがデバイスに接続することができないために、SCSI デバイスにアクセスすることができません。

Cannot add intr

ドライバが割り込みサービスルーチンをカーネルに追加することができません。

Cannot map dma

ドライバは、DMA コントローラを検出することができません。自動構成エラーです。

Cannot bind cmdarea

ドライバがアドレスに DMA ハンドルを割り当てることができません。

Cannot create devctl minor node

ドライバがコントローラのマイナーノードを作成することができません。

Cannot attach

ドライバが接続処理に失敗しました。通常、接続に失敗した原因を示す警告メッセージが後に続きます。

Disabled TQ since disconnects are disabled

scsi-options で切り離しが無効にされているため、タグ付きの待ち行列化が無効となりました。

Bad clock frequency

問題のあるハードウェアを調べてください。

Sync of pkt (<address>) failed

scsi パケットの同期処理が失敗しました。scsi_sync_pkt(9F) を参照してください。

All tags in use!

ドライバが別のタブ番号を割り当てられませんでした。ターゲットデバイスが、タグ付き待ち行列化に正しく対応していません。

Gross error in FAS366 status

ドライバが、重大な SCSI バス障害を検知しました。ケーブルとターミネータを調べてください。

Spurious interrupt

ドライバが、ハードウェアの割り込みが発生していないのに割り込みを受信しました。

Lost state in phasemanage

ドライバが、SCSI バスの状態を正しく判断することができませんでした。

Unrecoverable DMA error during selection

DMA がホスト SBus の障害を検知しました。問題のあるハードウェアを調べてください。

Bad sequence step (<step number>) in selection

FAS366 ハードウェアが問題のある処理手順を報告しました。問題のあるハードウェアを調べてください。

Undetermined selection failure

ターゲットの選択が予想されたとおりになりません。問題のあるハードウェアを調べてください。

Target <n>:failed reselection (bad reselect bytes)

再接続処理が失敗しました。ターゲットが間違った長さのメッセージバイトを送信しました。問題のあるハードウェアを調べてください。

Target <n>:failed reselection (bad identify message)

再接続処理が失敗しました。ターゲットが識別メッセージを送信しなかったか、メッセージが壊れています。

Target <n>:failed reselection (not in msgin phase)

再接続処理後の不正な SCSI バスフェーズ。問題のあるハードウェアを調べてください。

Target <n>:failed reselection (unexpected bus free)

再接続処理後の不正な SCSI バスフェーズ。問題のあるハードウェアを調べてください。

Target <n>: failed reselection (timeout on receiving tag msg)

再接続処理が失敗しました。ターゲットは、タグバイトの送信に失敗しました。問題のあるハードウェアを調べてください。

Target <n>: failed reselection (botched tag)

再接続処理が失敗しました。ターゲットは、タグバイトの送信に失敗しました。問題のあるハードウェアを調べてください。

Target <n>:failed reselection (invalid tag)

再接続処理が失敗しました。ターゲットは、間違ったタグバイトを送信しました。問題のあるハードウェアを調べてください。

- Target <n>:failed reselection (Party error in reconnect msg's)**
再接続処理が失敗しました。パリティエラーを検出しました。問題のあるハードウェアを調べてください。
- Target <n>: failed reselection (no command)**
再接続処理が失敗しました。ターゲットは、中止またはリセットを受け付けましたが、再接続を試みます。問題のあるハードウェアを調べてください。
- Unexpected bus free**
通知なしに、ターゲットがバスから切り離されました。問題のあるハードウェアを調べてください。
- Target <n> didn't disconnect after sending <message>**
ターゲットが、<message> を送信した後、予想されたとおりに切り離されませんでした。
- Bad sequence step (0x?) in selection**
処理手順レジスタが不適切な値を示しています。ターゲットの動作が正しくない可能性があります。
- Illegal dma boundary?**
ドライバが、処理できない境界にアクセスしようとしてしました。
- Unwanted data xfer direction for Target <n>**
ターゲットが、予想されていないフェーズに入りました。
- Unrecoverable DMA error on dma <send/receive>**
データを送信・受信している際に DMA エラーがあります。ホスト DMA コントローラが問題をいくつか検知しました。
- SCSI bus DATA IN phase parity error**
ドライバが、SCSI バスでパリティエラーを検出しました。
- SCSI bus MESSAGE IN phase parity error**
ドライバが、SCSI バスでパリティエラーを検出しました。
- SCSI bus STATUS phase parity error**
ドライバが、SCSI バスでパリティエラーを検出しました。
- Premature end of extended message**
拡張 SCSI バスメッセージが終了しませんでした。ターゲットのファームウェアを調べてください。
- Premature end of input message**
複数バイトの入力メッセージが切り捨てられました。ターゲットのファームウェアを調べてください。
- Input message botch**
ドライバが、ターゲットからのメッセージを判断することができませんでした。
- Extended message <n> is too long**
ターゲットが送信した拡張メッセージが予想されていたものより長いものでした。

<name> message <n> from Target <m> garbled
 ターゲット <m> が、ドライバが理解することができない値 <n> のメッセージ <name> を送信しました。

Target <n> rejects our message <name>
 ターゲット <n> が、ドライバが送信したメッセージを拒否しました。

Rejecting message <name> from Target <n>
 ドライバが、ターゲットから受信したメッセージを拒否しました。

Cmd transmission error
 ドライバは、コマンドバイトを送信することができませんでした。

Target <n> refused message resend
 ターゲットは、再送されたメッセージを受け付けませんでした。

MESSAGE OUT phase parity error
 ドライバが SCSI バス上でパリティエラーを検出しました。

Two bytes message <name> <value> rejected
 ドライバは、この 2 バイトメッセージを受け付けませんでした。

Gross error in fas status <stat>
 fas チップが FIFO オーバーフローなどの重大なエラーを示しました。

Polled cmd failed (target busy)
 ポーリングされたコマンドが失敗しました。これは、ターゲットが未処理のコマンドを妥当な時間内に終了することができなかったためです。

Polled cmd faild
 ポーリングされたコマンドが時間切れまたはバスエラーのために失敗しました。

Auto request sense failed
 ドライバがターゲットからの要求検知を取得することができません。

Disconnect command timeout for Target <id>.<lun>
 ターゲット *id/lun* が切り離されている間に時間切れが発生しました。これは、通常、ターゲットのファームウェアの問題によるものです。タグ付きの待ち行列化のターゲットでは、<n> コマンドは時間切れが検出された時点でまだ処理が終了していませんでした。

Disconnected tagged cmds (<n>) timeout for Target <id>.<lun>
 ターゲットまたは、*lun* が切り離されている間に時間切れが発生しました。これは、通常、ターゲットのファームウェアの問題によるものです。タグ付きの待ち行列化のターゲットでは、<n> コマンドは時間切れが検出された時点でまだ処理が終了していませんでした。

Connected command timeout for Target <id>.<lun>
 これは、通常、SCSI バスの問題です。ケーブルと終端を調べてください。

Target <id>.<lun> reverting to async. mode
 データ転送の停滞が検出されました。ドライバは、データ転送速度を落としてこの問題を回避します。

fas(7D)

Target <id>.<lun> reducing sync. transfer rate

データ転送の停滞が検出されました。ドライバは、データ転送速度を落としてこの問題を回避します。

Reverting to slow SCSI cable mode

データ転送の停滞が検出されました。ドライバは、データ転送速度を落としてこの問題を回避します。

Target <id> reducing sync. transfer rate

Target <id> reverting to async. mode

Target <id> disabled wide SCSI mode

Due to problems on the SCSI バス上での問題がありました。これ以上の問題を避けるために、ドライバはより保守的な動作モードに入ります。

Reset SCSI bus failed

SCSIバスの再設定が失敗しました。

External SCSI bus reset

別のイニシエータが SCSI バスを再設定しました。

警告

fasfas ハードウェア (FAS366) は、Fast/Wide SCSI モードに対応していますが、Fast20 には対応していません。最大 SCSI 帯域幅は、20 MB/秒です。イニシエータモードブロック処理 (IBS) には対応していません。

注意事項

fas ドライバは、ネゴシエートされた転送速度 (target<n>-sync-speed)、Wide バスに対応しているかどうか (target<n>-wide)、特定のターゲットの scsi-options (target<n>-scsi-options)、タグ付き待ち行列化が有効であるかどうか (target<n>-TQ) をターゲットごとに示す属性をエクスポートします。sync-speed の属性値は、KB/秒単位のデータ転送速度です。target<n>-TQ および target<n>-wide 属性は、ターゲットに対して対応する機能が有効であれば 1 を、無効であれば 0 の値を取ります。fas の属性を表示する場合は、prtconf(1M) (表示オプション) を参照してください。

SUNW,fas, instance #1

Driver software properties:

```
name <target3-TQ> length <4>
  value <0x00000001>.
name <target3-wide> length <4>
  value <0x00000000>.
name <target3-sync-speed> length <4>
  value <0x00002710>.
name <target3-scsi-options> length <4>
  value <0x000003f8>.
name <target0-TQ> length <4>
  value <0x00000001>.
name <pm_norm_pwr> length <4>
  value <0x00000001>.
name <pm_timestamp> length <4>
  value <0x30040346>.
name <scsi-options> length <4>
  value <0x000003f8>.
name <scsi-watchdog-tick> length <4>
  value <0x0000000a>.
```



```
name <scsi-tag-age-limit> length <4>
  value <0x00000002>.
name <scsi-reset-delay> length <4>
  value <0x00000bb8>.
Register Specifications:
  Bus Type=0x3, Address=0x8800000, Size=10
  Bus Type=0x3, Address=0x8810000, Size=40
Interrupt Specifications:
  Interrupt Priority=0x35 (ipl 5)
```

ffb(7D)

名前	ffb-24 ビット UPA カラーフレームバッファとグラフィックスアクセラレータ
機能説明	<p>ffb は 24 ビット UPA ベースのカラーフレームバッファとグラフィックスアクセラレータです。ffb は、シングルバッファのフレームとダブルバッファのフレームの 2 つの構成で提供されます。</p> <p>シングルバッファのフレーム バッファ 32 の 1280 × 1024 ピクセルの表示メモリー平面によって構成されます。これには、24 ビットシングルバッファリングと 8 ビット X 平面が含まれます。</p> <p>ダブルバッファのフレームバッファ 32 の 1280 × 1024 ピクセルの表示メモリー平面によって構成されます。これには、24 ビットダブルバッファリング、8 ビット X 平面、28 ビット Z バッファ平面、4 ビット Y 平面が含まれます。</p> <p>ffb ドライバは、fbio(7I) で定義されている以下のフレームバッファ ioctl に対応しています。</p> <p>FBIOPUTCMAP、FBIOGETCMAP、FBIOSVIDEO、FBIOGVIDEO、FBIOVERTICAL、 FBIOSCURLOR、FBIOGCURLOR、FBIOSCURPOS、FBIOGCURPOS、FBIOGCURMAX、 FBIO_WID_PUT、FBIO_WID_GET</p>
ファイル	/dev/fbs/ffb0 デバイス特殊ファイル
関連項目	ffbconfig(1M), mmap(2), fbio(7I)

名前	ge – GEM GigabitEthernet デバイスドライバ
形式	/dev/ge
機能説明	<p>GigabitEthernet ドライバ (ge ドライバ) は、マルチスレッド化され、読み込みや複製が可能な STREAMS ハードウェアドライバです。GEM、SBus、および PCI GigabitEthernet アドイン型アダプタに対してコネクションレス型のデータリンクプロバイダインタフェース dlpi(7P) をサポートします。ge ドライバは、システムに組み込まれた、複数の GEM アダプタをサポートするものです。、ge ドライバは、おもに GEM 対応の Ethernet ハードウェアをサポートし、SUNW, sbus-gem (SBus 対応 GEM) や pci108e, 2bad (PCI 対応 GEM) などのデバイスを制御します。その機能は、チップの初期化、フレームの転送と受信、マルチキャストと プロミスキュアスのサポート、エラーの復旧処理および報告です。</p> <p>GEM デバイスは、外部 SERDES および光ファイバー通信トランシーバを制御する GEM ASIC (専用チップ) を使用して、100BASE-SX のネットワークインタフェースを提供します。GEM ASIC は、適切なバスインタフェースに対し、MAC 機能や PCS (Physical Code Sub-layer : 物理コードサブレイヤー) 機能を提供します。外部 SERDES は、光ファイバートランシーバに接続され、物理的な接続を提供します。</p> <p>1000BASE-SX 規格には、自動的に動作モードを選択する自動ネゴシエーション機能が規定されています。さらに全二重モードのために、GEM ASIC は、フロー制御に準拠した自動ネゴシエーション機能を実現しています。GEM PCS 機能は、遠隔リンク先 (リンクパートナー) に対して、自動ネゴシエーションを実行したり、遠隔リンク先からのメッセージを受信したりします。また、優先度に応じてもっとも優先順位の高いモードを選択します。ドライバが対応しているシステムでは、強制モードもサポートします。</p>
アプリケーション プログラミング インタフェース	<p>文字複製された特殊デバイスである /dev/ge は、システムに組み込まれている、すべての ge コントローラへのアクセスに使用されます。</p> <p>ge ドライバは、スタイル 2 型のデータリンクサービスを提供するものです。M_PROTO 型、および M_PCPROTO 型のメッセージは、DLPI プリミティブとして解釈されます。DLPI プリミティブについては、<sys/dlpi.h> ファイルで定義されています。詳細については、dlpi(7P) のマニュアルページを参照してください。</p> <p>。ユーザーからの DL_ATTACH_REQ メッセージは、特殊デバイス/ ppa に対して開かれたストリームに関連付けられなければなりません。ppa ID は、unsigned long型のデータとして解釈され、対応するデバイスインスタンス (ユニット) 番号を指し示します。ppa フィールドの値が、そのシステムの有効なデバイスインスタンス番号と一致しない場合、ドライバは、エラー値とし DL_ERROR_ACK を返します。デバイスは、最初の接続時に初期化され、最後の接続時に終了処理 (停止) されます。</p> <p>DL_INFO_REQ に対してドライバから換えされる DL_INFO_ACK の値は、以下のようになります。</p> <ul style="list-style-type: none"> ■ SDU の最大値は 1500 です (ETHERMTU -<sys/ethernet.h> で定義されている)。 ■ SDU の最小値は 0 です。

ge(7D)

- dlsap のアドレス長は 8 です。
- MAC 型は DL_ETHER です。
- sap 長の値は -2 です。これは、物理アドレスコンポーネントの直後に、DLSAP アドレス内部にある、2 バイトの sap コンポーネントが続くことを意味しています。
- サービスモードは DL_CLDLS です。
- オプションなしのサービス品質 (QOS) サポートしない場合、QOS フィールドの値は 0 になります。
- サービス提供の形式は DL_STYLE2 です。
- バージョンは DL_VERSION_2 です。
- ブロードキャストアドレスの値は、Ethernet/IEEE ブロードキャストアドレス (0xFFFFFFFF) です。

DL_ATTACHED 状態において特殊な Service Access Pointer (SAP) をストリームに関連付けるために、ユーザーは DL_BIND_REQ を送信する必要があります。フィールドの値が [0-0xFFFF] の範囲内にある限り、ドライバは、DL_BIND_REQ 中の sap フィールドを Ethernet の *type* (種類) として解釈します。いかなる場合も、ただ 1 つの種類 Ethernet をストリームに結合することができます。

ユーザーが sap の値として 0 を指定した場合、受け取り側は、802.3 モードになります。type フィールドの値が [0-1500] の範囲内にあるメディアを受信したすべてのフレームは、802.3 準拠とみなされます。そして、fsap の値が 0 に結合されている、開いているすべてのストリームに経路付けられます。1 つ以上のストリームがであるなら、DL_UNITDATA_IND メッセージとして、複数のストリームに経路付けられます。

データ伝送中、ge ドライバは、DL_BIND_REQ 中の sap 値を検査して、sap が 0 かどうか、および、destination type フィールドの値が [0-1500] の範囲内になるかどうかを判断します。もし、いずれの条件も真の場合、ドライバは、すべての DL_UNITDATA_REQ メッセージの直後に続く最初の M_PROTO mblk (メッセージブロック) を含めずに、メッセージの長さを算出します。そして、この値を MAC フレームヘッダ長フィールドに持つ 802.3 フレームを送信します。

ge ドライバの DLSAP アドレスの書式は、8 バイトの DLSAP アドレスを生成する、2 バイトの sap (型) コンポーネントと、それに続く (Ethernet の) 6 バイトの物理アドレスコンポーネントから成り立っています。アプリケーションは、この特殊な実装を指定する DLSAP アドレスの書式をハードコードすべきではありません。アドレスの構成と分解を行う DL_INFO_ACK プリミティブに返される情報を使用すべきです。sap のデータ長、DLSAP の全データ長、sap または物理的な指示は、DL_INFO_ACK の中に含まれています。物理アドレス長は、全 DLSAP アドレス長から sap の値を弾くか、ストリームに関連付けられた現在の物理アドレスを得る、DL_PHYS_ADDR_REQ を発行することにより、算出できます。

DL_BOUND ステートでは、sap ドライバに DL_UNITDATA_REQ メッセージを送ることによって、ユーザーは Ethernet 上にフレームを送信することがあります。ge ドライバは、DL_UNITDATA_IND メッセージに Ethernet タイプが一致している sap を持つ、開かれた、もしくは結合されたストリームに、受信した Ethernet フレームを経路

付けすることになります。必要に応じて、受信された Ethernet フレームは複製され、複数の開かれたストリームに経路付けされます。DL_UNITDATA_REQ および DL_UNITDATA_IND メッセージの中に含まれる DLSAP アドレスは、sap (型) と物理 (Ethernet) コンポーネントから構成されています。

強制コネクションレス DLPI メッセージセットに加えて、ge ドライバは、以下のプリミティブをサポートしています。

ge プリミティブ

DL_ENABMULTI_REQ と DL_DISABMULTI_REQ プリミティブは、個々のマルチキャストグループアドレスの受け取りを許可・不許可します。この 2 つのプリミティブを使うことによって、マルチキャストアドレスの組み合わせが各ストリーム上で何度も生成されたり、変更されたりします。ドライバは、DL_ATTACHED に続くいかなるステートにおいても、DL_ENABMULTI_REQ と DL_DISABMULTI_REQ のプリミティブを受け入れます。

d1_level フィールドの DL_PROMISC_PHYS フラグセットによって、DL_PROMISCON_REQ と DL_PROMISCOFF_REQ プリミティブは、ローカルホストの生成したフレームを含むメディアにある、すべてのプロミスキュアモードフレームの受け入れを許可・不許可します。DL_PROMISC_SAP フラグセットとともに使用される時、この機能は、すべての sap (Ethernet の種類を表す) の値の受け入れを許可・不許可します。DL_PROMISC_MULTI フラグセットとともに使用される時、この機能は、すべてのマルチキャストグループアドレスの受け入れを許可・不許可します。それぞれの効果は、常にそれぞれのストリームの土台となり、該当するストリーム、または他のストリームの、他の sap および物理レベルの設定に依存します。

DL_PHYS_ADDR_REQ プリミティブは、DL_PHYS_ADDR_ACK プリミティブ中のストリームにその時点で関連 (結合) 付けられた 6 オクテットの Ethernet アドレスを返します。このプリミティブは、成功した DL_ATTACH_REQ の後に続く状態であるときのみ有効です。

DL_SET_PHYS_ADDR_REQ プリミティブは、このストリームにその時点で関連 (結合) 付けられた 6 オクテットの Ethernet アドレスを変更します。最初にこのストリームを開いたプロセスの資格 (credentials) は、スーパーユーザでなければなりません。そうでない場合、DL_ERROR_ACK 内に EPERM が換えされます。DL_SET_PHYS_ADDR_REQ プリミティブは、このデバイスに結合された、現在および未来の他のすべてのストリームに影響を及ぼす点で、破壊的なものです。DL_SET_PHYS_ADDR_REQ DL_SET_PHYS_ADDR_REQ プリミティブがこのストリーム上で成功したとき、このデバイスに結合された他のすべてのストリームに M_ERROR が送られます。いったん変更があると、その後に関われ、このデバイスに結合されたすべてのストリームは、この新しい物理アドレスを取得するはずですが、変更があったとき、この物理アドレスは、DL_SET_PHYS_ADDR_REQ によって物理アドレスが再度変更されるか、システムが再起動されるまで保持されます。

ge ドライバ

デフォルトでは、ge ドライバは自動ネゴシエーションを実行して、リンクのモードとフロー制御モードを選択します。リンクは、以下に示す 4 つのモードのいずれかになります。

- 1000 Mbps、全二重接続
- 1000 Mbps、半二重接続

ge(7D)

- 対称的中断 (Symmetric Pause)
- 非対称的中断 (Asymmetric Pause)

通信速度とモードは、1000BASE-TX 規格に規定されています。

自動ネゴシエーションプロトコルは、以下の項目を自動的に選択します。

- 動作モード (半二重または全二重)
- フロー制御機能 (対称または非対称、あるいはその両方)

自動ネゴシエーションプロトコルは、以下の処理を行います。

- リンク先パートナーのサポートする全動作モードの取得
- リンク先パートナーに対する、各機能の通知
- 優先度に基づいた、最も優先度の高いモードの選択

GEM ハードウェアは、上記の全動作モードを実現しています。デフォルト時は、リンク先との接続のために自動ネゴシエーションを使用し、次に、リンク先パートナーと同じ動作モードを選択します。PCS は、強制モードもサポートしています。これは、`ndd` ユーティリティを使用して、ドライバが動作モードとフロー制御機能を選択できるモードです。

GEM デバイスは、プログラム可能な Inter-Packet Gap (IPG) パラメータの `ipg1` および `ipg2` をサポートしています。デフォルトでは、ドライバは、`ipg1` を 8 バイトの倍数に、`ipg2` を 4 バイトの倍数に設定します (これらは標準的な数値です)。ユーザーは、これらの数値を標準的な 1000 Mbps IPG セットから 0.096 マイクロ秒へ変更してもかまいません。

ge のパラメータリスト

`ge` ドライバを使用して、デバイスの各種パラメータを設定したり、取得することができます。パラメータのリストには、現在のトランシーバの状態、現在の接続の状態、パケット間隔、PCS の各機能、およびリンク先パートナーの各機能が含まれています。

PCS には、機能別に 2 つのセットがあります。1 つは、ハードウェアの各機能を反映したものであり、読み取り専用です。2 つめのセットは読み取り/書き込み可能で、高速選択 (speed selection) として使用され、ユーザーが選択した数値を反映したものです。ブート時には、これらの 2 つの機能セットは同じになります。リンク先パートナーの各機能は読み取り専用で、変更できません。

ファイル	<code>/dev/ge</code>	<code>ge</code> 文字型特殊デバイス
	<code>/kernel/drv/ge.conf</code>	システム全般のデフォルトデバイスドライバ属性
関連項目	<code>ndd(1M)</code> , <code>netstat(1M)</code> , <code>driver.conf(4)</code> , <code>dlpi(7P)</code> , <code>le(7D)</code> , <code>hme(7D)</code> , <code>qfe(7D)</code>	

名前	glm – GLM SCSI ホストバスアダプタドライバ
形式	scsi@unit-address
機能説明	<p>glm ホストバスアダプタドライバは、LSI 53c810、LSI 53c875、LSI 53c876、LSI 53C896、および LSI 53C1010 SCSI チップに対応した SCSI 準拠のネクサスドライバです。</p> <p>glm ドライバは、SCSA インタフェースが提供する標準機能に対応しています。ドライバは、タグ付きおよびタグなしの待ち行列化、Narrow/Wide/Fast/Ultra SCSI/Ultra SCSI 2/Ultra SCSI 3、自動要求検知に対応しています。ただし、リンクされたコマンドには対応していません。</p>
ドライバの設定	<p>glm ドライバは、広域的な SCSI 設定に優先する属性を glm.conf に定義することによって設定することができます。glm で使用・変更することができる属性は、scsi-options、target<n>-scsi-options、scsi-reset-delay、scsi-tag-age-limit、scsi-watchdog-tick、および scsi-initiator-id です。</p> <p>target<n>-scsi-options は target<n> の scsi-options 属性に優先します。<n> には、10 進数の 0 ~ 15 の値を指定することができます。使用することができる scsi オプションは、SCSI_OPTIONS_DR、SCSI_OPTIONS_SYNC、SCSI_OPTIONS_TAG、SCSI_OPTIONS_FAST、SCSI_OPTIONS_WIDE、SCSI_OPTIONS_FAST20、SCSI_OPTIONS_FAST40、および SCSI_OPTIONS_FAST80 です。</p> <p>周期的な時間間隔である scsi-watchdog-tick が経過する度に、glm ドライバは、現在実行中および切り離されたすべてのコマンドを調べ、時間切れかどうかを確認します。</p> <p>scsi-tag-age-limit は、glm ドライバがすべてのタグ ID を一巡して調べた後に、現在使用されている特定のタグ ID の割り当てを試みる回数です。同一のタグ ID が scsi-tag-age-limit の回数だけ使用されると、すべての未処理のコマンドが完了するか、時間切れになるまで、このターゲットにコマンドは発行されません。</p> <p>詳細については、scsi_hba_attach(9F) を参照してください。</p>
使用例	<p>例 1 glm 設定ファイルの使用</p> <p>ファイル /kernel/drv/glm.conf を作成して、以下の行を追加してください。</p> <pre>scsi-options=0x78;</pre> <p>これは、すべての glm インスタンスに対して、タグ付き待ち行列化、Fast SCSI、Fast/Ultra SCSI、Wide モードを無効にします。</p> <p>以下の例は、1 つの特定の glm のオプションを無効にします。詳細については、driver.conf(4) と pci(4) を参照してください。</p> <pre>name="glm" parent="/pci@1f,4000" unit-address="3" target1-scsi-options=0x58 scsi-options=0x178 scsi-initiator-id=6;</pre>

例1 glm 設定ファイルの使用 (続き)

OBP のデフォルトのイニシエータ ID は7ですが、接続時に6に変更されます。OBP のイニシエータ ID を変更してください。

上記の例では、ターゲット1の `scsi-options` を `0x58` に設定し、この SCSI バス上にある他のすべてのターゲットは `0x178` に設定しています。

親の物理パス名は、`/devices` ツリーを使用するか、論理デバイス名のリンクをたどることによって判定することができます。

```
# ls -l /dev/rdisk/c0t0d0s0
lrwxrwxrwx 1 root root 45 May 16 10:08 /dev/rdisk/c0t0d0s0 ->
  . . / . . devices/pci@1f,4000/scsi@3/sd@0,0:a,raw
```

この場合、上記の例と同じように、親は `/pci@1f,4000` で、`unit-address` は、`scsi@3` ノードに結合されている番号です。

`scsi-options` をターゲットごとに設定するには、以下のように入力します。

```
target1-scsi-options=0x78;
device-type-scsi-options-list =
  "SEAGATE ST32550W", "seagate-scsi-options" ;
seagate-scsi-options = 0x58;
scsi-options=0x3f8;
```

この例は、ターゲット1の `scsi-options` を `0x78` に設定し、ある特定の種類のディスクの `scsi-options` を `0x58` に設定し、この SCSI バス上にある他のすべてのターゲットの `scsi-options` を `0x3f8` に設定します。

ターゲット ID ごとに指定された `scsi-options` は、最も優先順位が高くなります。その次に優先順位が高いのは、デバイスの種類ごとに指定された `scsi-options` です。バスごとに指定される広域的な `scsi-options` (すべてのインスタンスに影響します) は、最も優先順位が低くなります。

指定された `scsi-options` が有効になるには、システムを再起動する必要があります。

ドライバの機能

ターゲットドライバがドライバの機能を有効にするには、glm ドライバの機能を設定する必要があります。ターゲットドライバは、これらの機能の照会および変更ができます。照会および変更ができる機能には、`synchronous`、`tagged-qing`、`wide-xfer`、`auto-rqsense`、`qfull-retries`、`qfull-retry-interval` があります。他のすべての機能は、照会のみ行うことができます。

デフォルトでは、`tagged-qing`、`auto-rqsense`、`wide-xfer` 機能は無効で、`disconnect`、`synchronous`、`untagged-qing` 機能は有効です。これらの機能には、2進数の値(0または1)のみを設定することができます。`qfull-retries` のデフォルト値は10で、`qfull-retry-interval` のデフォルト値は100です。`qfull-retries` 機能は、`u_char`型(0～255)で、`qfull-retry-interval` は、`u_short`型(0～65535)です。

ターゲットドライバは、tagged-qing および wide-xfer を明示的に有効にする必要があります。untagged-qing 機能は常に有効で、その値を変更することはできません。

scsi-options の値と機能が矛盾している場合は、scsi-options に設定されている値が常に優先されます。scsi_ifsetcap(9F) 呼び出しでは、whom != 0 のみを使用することができます。

詳細については、scsi_ifsetcap(9F) および scsi_ifgetcap(9F)を参照してください。

ファイル /kernel/drv/glm ELF ELF カーネルモジュール

/kernel/drv/glm.conf 省略可能な設定ファイル

属性 以下の属性については、attributes(5) を参照してください。

属性の種類	属性の値
アーキテクチャ	LSI 53c810、LSI 53c875、LSI 53c876、LSI 53C896、および LSI 53C1010 SCSI、入出力プロセッサを持つPCI ベースのシステムに限定

関連項目 prtconf(1M), driver.conf(4), pci(4), attributes(5), scsi_abort(9F), scsi_hba_attach(9F), scsi_ifgetcap(9F), scsi_ifsetcap(9F), scsi_reset(9F), scsi_sync_pkt(9F), scsi_transport(9F), scsi_device(9S), scsi_extended_sense(9S), scsi_inquiry(9S), scsi_pkt(9S)

Writing Device Drivers

『ANSI Small Computer System Interface-2 (SCSI-2)』

LSI Logi 社 (旧 Symbios Logic 社.):

- SYM53c810 PCI-SCSI 入出力プロセッサ - Narrow モードをサポートする
- SYM53c875 PCI-SCSI 入出力プロセッサ - Fast-20 をサポートする
- SYM53c876 PCI-SCSI 入出力プロセッサ - デュアルチャネル Fast-20 をサポートする
- SYM53c896 PCI-SCSI 入出力プロセッサ - デュアルチャネル Fast-40 をサポートする
- SYM53c1010 PCI-SCSI 入出力プロセッサ - デュアルチャネル Fast-80 をサポートする

診断 以下で説明するメッセージは、コンソールに表示されると同時にログファイルに記録されます。

Device is using a hilevel intr

glm ドライバが使用することができない割り込みレベルでデバイスが設定されています。PCI デバイスを調べてください。

map setup failed

ドライバがデバイスレジスタを割り当てることができません。問題のあるハードウェアを調べてください。ドライバがデバイスに接続することができないために、SCSI デバイスにアクセスすることができません。

glm_script_alloc failed

ドライバが SCSI プロセッサの SCRIPTS を読み込むことができません。問題のあるハードウェアを調べてください。ドライバがデバイスに接続することができないために、SCSI デバイスにアクセスすることができません。

cannot map configuration space.

ドライバが設定レジスタを割り当てることができません。問題のあるハードウェアを調べてください。ドライバがデバイスに接続することができないために、SCSI デバイスにアクセスすることができません。

attach failed

ドライバが接続することができません。通常、接続が失敗した理由を示す警告が出力されます。これらは、ハードウェア障害とみなすことができます。

SCSI bus DATA IN phase parity error

ドライバが、SCSI バスでパリティエラーを検出しました。

SCSI bus MESSAGE IN phase parity error

ドライバが、SCSI バスでパリティエラーを検出しました。

SCSI bus STATUS phase parity error

ドライバが、SCSI バスでパリティエラーを検出しました。

Unexpected bus free

通知なしに、ターゲットがバスから切り離されました。問題のあるハードウェアを調べてください。

Disconnected command timeout for Target <id>.<lun>

ターゲット *id/lun* が切り離されている間に時間切れが起きました。これは、通常、ターミナルのファームウェアの問題です。タグ付き待ち行列化を行うターゲットについては、時間切れが検出された際に *<n>* コマンドが未処理になります。

Disconnected tagged cmds (<n>) timeout for Target <id>.<lun>

ターゲットまたは、*lun* が切り離されている間に時間切れが発生しました。これは、通常、ターゲットのファームウェアの問題によるものです。タグ付きの待ち行列化のターゲットでは、*<n>* コマンドは時間切れが検出された時点でまだ処理が終了していませんでした。

Connected command timeout for Target <id>.<lun>

これは、通常、SCSI バスの問題です。ケーブルと終端を調べてください。

Target <id> reducing sync. transfer rate

データ転送の停滞または DATA-IN フェーズパリティエラーが検出されました。ドライバは、データの転送レートを下げることによって、この問題の解決を試みます。

Target <id> reverting to async. mode

このターゲットの 2 つ目のデータ転送の停滞が検出されました。ドライバは、データの転送レートを下げることによって、この問題の解決を試みます。

Target <id> disabled wide SCSI mode

このターゲットの 2 つ目のデータフェーズの停滞が検出されました。ドライバは、Wide SCSI モードを無効にすることによって、この問題の解決を試みます。

auto request sense failed

自動要求パケットの開始が失敗しました。別の自動要求パケットがすでに転送されている可能性があります。

invalid reselection (<id>.<lun>)

再選択が失敗しました。ターゲットは、abort または reset を受け付けましたが、再接続を試みます。問題のあるハードウェアを調べてください。

invalid intcode

SCRIPTS プロセッサが無効の SCRIPTS 割り込みを生成しました。問題のあるハードウェアを調べてください。

注意事項

glm ドライバは次の LSI チップをサポートします。

- LSI 53C810 - Narrow、Fast SCSI モードをサポートします。最大 SCSI 帯域幅は 10M バイト/秒です。
- LSI 53C875 - Wide、Fast、および Ultra SCSI モードをサポートします。最大 SCSI 帯域幅は 40M バイト/秒です。
- LSI 53C896 - Wide、Fast、および Ultra SCSI 2 モードをサポートします。最大 LVD SCSI 帯域幅は 80M バイト/秒です。
- LSI 53C1010 - Wide、Fast、および Ultra SCSI 3 モードをサポートします。最大 LVD SCSI 帯域幅は 160M バイト/秒です。

glm ドライバは、ネゴシエートされた転送速度 (target<n>-sync-speed)、Wide バスに対応しているかどうか (target<n>-wide)、特定のターゲットの scsi-options (target<n>-scsi-options)、タグ付き待ち行列化が有効であるかどうか (target<n>-TQ) をターゲットごとに示す属性をエクスポートします。sync-speed の属性値は、KB/秒単位のデータ転送速度です。target<n>-TQ および target<n>-wide 属性は、ターゲットに対して対応する機能が有効であれば 1 を、無効であれば 0 の値を取ります。glm の属性を表示する場合は、prtconf(1M) (表示オプション) を参照してください。

```
scsi, instance #0
```

```
  Driver properties:
```

```
    name <target6-TQ> length <4>
      value <0x00000000>.
    name <target6-wide> length <4>
      value <0x00000000>.
    name <target6-sync-speed> length <4>
      value <0x00002710>.
    name <target1-TQ> length <4>
      value <0x00000001>.
    name <target1-wide> length <4>
      value <0x00000000>.
    name <target1-sync-speed> length <4>
      value <0x00002710>.
    name <target0-TQ> length <4>
      value <0x00000001>.
```

glm(7D)

```
name <target0-wide> length <4>
  value <0x00000001>.
name <target0-sync-speed> length <4>
  value <0x00009c40>.
name <scsi-options> length <4>
  value <0x000007f8>.
name <scsi-watchdog-tick> length <4>
  value <0x0000000a>.
name <scsi-tag-age-limit> length <4>
  value <0x00000002>.
name <scsi-reset-delay> length <4>
  value <0x00000bb8>.
name <latency-timer> length <4>
  value <0x00000088>.
name <cache-line-size> length <4>
  value <0x00000010>.
```

名前	hme – SUNW,hme Fast-Ethernet デバイスドライバ
形式	/dev/hme
機能説明	<p>SUNW,hme Fast-Ethernet ドライバは、マルチスレッド化された、読み込みや複製が可能な STREAMS ハードウェアドライバです。このドライバでは、SUNW,hme Fast-Ethernet コントローラにおいてコネクションレス型のデータリンクプロバイダインタフェース(dlpi(7P))を使用することができます。マザーボードと数種類の追加型 SBus SUNW,hme コントローラを使用することができます。このドライバでは、システムにインストールされた複数の SUNW,hme コントローラを使用することができます。</p> <p>hme ドライバは、SUNW,hme ハードウェアの基本的な機能を提供します。このドライバは、SUNW,hme デバイスを扱うために使用します。機能には、チップの初期化、フレームの転送と受信、マルチキャストとプロミスキュアスへの対応、エラーからの回復および報告が含まれます。SUNW,hme デバイスは、サンの FEPS ASIC と内部トランシーバを使用して 100Base-TX ネットワークインタフェースを提供します。FEPS ASIC は、SBus インタフェースと MAC 機能を提供します。を使用します。物理層機能は、RJ45 コネクタに接続する内部トランシーバによって提供されます。PCI 対応の SUNW,hme デバイスは、サンの PFEX ASIC を使用して PCI インタフェースと MAC 機能を提供します。両方のデバイスとも 100Base-TX オンボードトランシーバに接続します。100Base-TX オンボードトランシーバは、RJ-45 コネクタに接続し、物理層の機能と外部との接続を提供します。RJ-45 コネクタに加えて、MII (Media Independent Interface) コネクタも SunSwitch SBus アダプタボードを除くすべての SUNW,hme デバイスで提供されます。MII インタフェースは、100Base-TX 規格で指定された物理媒体 (銅またはファイバ) を使用する外部トランシーバに接続するために使用されます。外部トランシーバが MII に接続されている場合、hme ドライバは外部トランシーバを選択し、内部トランシーバを無効にします。</p> <p>100Base-TX 規格によって、操作のモードと速度を自動的に選択するための「自動ネゴシエーション」プロトコルが指定されます。内部トランシーバは、接続の遠隔地側 (接続相手) と「自動ネゴシエーション」を行うことができ、遠隔地側の機能に関する情報を受け取ります。内部トランシーバは、属性に基づいて、「最も優先順位の高い共通のモード」を選択します。また、ドライバが操作のモードを選択することができます。場合によっては、強制モードを使用することができます。</p>
アプリケーション プログラミング インタフェース	複製された文字型特殊デバイスの /dev/hme は、システムにインストールされたすべての SUNW,hme コントローラに接続するために使用します。
hme と DLPI	<p>hme ドライバは、「スタイル 2」のデータリンクサービスプロバイダです。すべての M_PROTO 型および M_PCPROTO 型のメッセージは、DLPI 基本式として解釈されません。有効な DLPI 基本式は <sys/dlpi.h> の中で定義されています。詳細は、dlpi(7P) を参照してください。開いているストリームに特殊デバイス (ppa) を関連付けるには、ユーザーからの明示的な DL_ATTACH_REQ メッセージが必要となります。この ppa ID は、符号なしの長整数として伝達され、対応するデバイスインスタンス (ユニット) 番号を示します。ppa フィールドの値がシステムにおける有効なデバイスインスタンス番号に対応していない場合は、エラー (DL_ERROR_ACK) が hme ドライバによって返されます。デバイスは最初の接続時に初期化され、最後の切り離し時に終了 (停止) します。</p>

hme(7D)

ユーザーからの DL_INFO_REQ に応答して hme ドライバが返す DL_INFO_ACK 基本式の値を以下に示します。

- SDU の最大値は 1500 になります (<sys/ethernet.h> で定義されている ETHERMTU)。
- SDU の最小値は 0 です。
- dlsapdlsap アドレス長は 8 です。
- MAC 形式は DL_ETHER です。
- sap 長の値は -2 で、これは、DLSAP アドレスの中で物理アドレス構成要素のすぐ後に 2 バイト長の sap 構成要素が位置することを意味します。
- サービスモードは DL_CLDLS です。
- サービスの品質 (QOS: quality of service) オプションは現在のところ含まれていないため、QOS フィールドは 0 です。
- プロバイダスタイルは DL_STYLE2 です。
- バージョンは DL_VERSION_2 です。
- ブロードキャストアドレスの値は Ethernet/IEEE のブロードキャストアドレス (0xFFFFFFFF) です。

DL_ATTACHED 状態になると、特定の sap (Service Access Pointer) をストリームと関連付けるために、ユーザーは DL_BIND_REQ を送る必要があります。hme ドライバは、DL_BIND_REQ 内の SAP フィールドを Ethernet の「タイプ」として解釈します。したがって、sap フィールドの有効な値は [0-0xFFFF] の範囲になります。一度にストリームに接続できるのは 1 つのタイプの Ethernet だけです。

ユーザーが sap の値として 0 を選択すると、受信側は 802.3 モードになります。タイプフィールドが [0-1500] の範囲内にある媒体から受信したすべてのフレームは 802.3 フレームとみなされ、sap 値が 0 のすべての開いているストリームに配信されます。複数のストリームが 802.3 モードにある場合は、フレームは複製され、DL_UNITDATA_IND メッセージとして複数のストリームに配信されます。

hme ドライバは、送信の際に DL_BIND_REQ の sap フィールドの値が 0 であるかどうかと、宛先タイプフィールドが [0-1500] の範囲内であるかどうかを確認します。どちらかの条件が満たされている場合、ドライバは最初の M_PROTO mblk (メッセージブロック) を除いたすべての後続の DL_UNITDATA_REQ メッセージの長さを計算し、MAC フレームヘッダー長フィールドが計算結果である 802.3 フレームを送信します。

hme ドライバの DLSAP アドレス形式は、6 バイトの物理 (Ether-net) アドレス部分と、それに続く 2 バイトの sap (タイプ) 部分とで構成されます。アプリケーションでは、DLSAP アドレスの作成や削除を行う際に、実装に依存した DLSAP アドレス形式を明示的に使用せずに、DL_INFO_ACK 基本式で返される情報を使用してください。sap 長、完全な DLSAP 長、sap/物理の順序は DL_INFO_ACK に含まれます。物理アドレス長は、完全な DLSAP アドレス長から sap 長を引くか、DL_PHYS_ADDR_REQ を発行して、ストリームに対応する現在の物理アドレスを取得することによって計算することができます。

DL_BOUND 状態になると、hme ドライバに DL_UNITDATA_REQ メッセージを送信することによって、ユーザーは Ethernet 上でフレームを送信することができます。hme ドライバは、DL_UNITDATA_IND メッセージの Ethernet のタイプと一致する sap 値を持ち、開いていて結合されているストリームすべてに、受信された Ethernet フレームを配信します。受信された Ethernet フレームは複製され、必要に応じて複数の開いているストリームに配信されます。DL_UNITDATA_REQ および DL_UNITDATA_IND メッセージに含まれる DLSAP アドレスは、sap (タイプ) と物理 (Ethernet) 部分で構成されます。

hme ドライバは、強制的なコネクションレス DLPI メッセージセットに加えて、以下の基本式に対応しています。

hme の基本式

DL_ENABMULTI_REQ および DL_DISABMULTI_REQ 基本式は、個々のマルチキャストグループアドレスの受信を有効または無効にします。これらの基本式を使用して、一群のマルチキャストアドレスをストリームごとに繰り返し作成および変更することができます。これらの基本式は、DL_ATTACHED 後の状態にある場合に、hme ドライバによって受け付けられます。

dl_level フィールド内に DL_PROMISC_PHYS フラグが設定された状態での DL_PROMISCOFF_REQ および DL_PROMISCON_REQ の基本式は、ローカルホストによって生成されたフレームを含む、媒体上のすべての(「プロミスキュアモード」の)フレームの受信を有効または無効にします。DL_PROMISC_SAP フラグが設定されている状態で使用すると、sap 値 (Ethernet のタイプ) の受信を有効または無効にします。DL_PROMISC_MULTI フラグが設定されている状態で使用すると、すべてのマルチキャストグループアドレスの受信を有効または無効にします。どちらの効果も、常にストリームごとに適用され、当該ストリームやその他のストリームの sap および物理レベルの設定の影響は受けません。

DL_PHYS_ADDR_REQ 基本式は、DL_PHYS_ADDR_ACK 基本式の中のストリームに対応する (接続されている) Ethernet アドレス (6 桁の数) を返します。この基本式は、DL_ATTACH_REQ の成功後の状態においてのみ有効です。

DL_SET_PHYS_ADDR_REQ 基本式は、このストリームに対応する (接続されている) Ethernet アドレスを変更します。このストリームを最初に開く処理は、スーパーユーザーのみが行うことができます。スーパーユーザーでない場合は、DL_ERROR_ACK で EPERM が返されます。この基本式は、このデバイスに接続された現在および未来のすべてのストリームに影響する意味で破壊的です。この基本式がこのストリームに対して成功すると、このデバイスに接続された他のすべてのストリームに M_ERROR が送信されます。物理アドレスが一度変更されると、その後開かれ、このデバイスに接続されたストリームは、この新しい物理アドレスを取得します。物理アドレスが一度変更されると、物理アドレスを再び変更するためにこの基本式が使用されるか、システムが再起動されるまでこの物理アドレスは存続します。

hme ドライバ

デフォルトでは、内部トランシーバが使用されている場合に、hme ドライバは接続のモードおよび速度を選択するために「自動ネゴシエーション」を実行します。

hme(7D)

外部トランシーバが MII インタフェースに接続されている場合、hme ドライバは、ネットワーク操作用に外部トランシーバを選択します。外部トランシーバが「自動ネゴシエーション」に対応している場合、hme ドライバは、自動ネゴシエーションの手続きを使用して接続のモードと速度を選択します。外部トランシーバが自動ネゴシエーションに対応していない場合、hme ドライバはそのトランシーバに対応している最も優先順位の高いモードを選択します。

接続は、以下の 4 つのモードのいずれかになります。

- 100 Mbps、全二重
- 100 Mbps、半二重
- 10 Mbps、全二重
- 10 Mbps、半二重

これらの速度とモードについては、100Base-TX 規格に記述されています。

自動ネゴシエーションプロトコルは以下を自動的に選択します。

- 動作モード (全二重または半二重)
- 速度 (100 Mbps または 10 Mbps)

自動ネゴシエーションプロトコルは以下を行います。

- 接続相手に対応しているすべての動作モードを取得します。
- 接続相手に対して自身の機能を通知します。
- 優先順位に基づいて最も優先順位の高い共通のモードを選択します。

内部トランシーバは、上記のすべての動作モードおよび速度に対応しています。デフォルトでは、内部トランシーバが使用されている場合に、接続のモードおよび速度と接続相手との共通の動作モードを選択するために自動ネゴシエーションが使用されます。

外部トランシーバが MII インタフェースに接続されている場合、hme ドライバは、ネットワーク操作用に外部トランシーバを選択します。外部トランシーバが自動ネゴシエーションに対応している場合は、以下のようになります。

- hme ドライバは、接続の速度とモードを選択するために自動ネゴシエーションを使用します。

外部トランシーバが自動ネゴシエーションに対応していない場合は、以下のようになります。

- hme ドライバは、トランシーバに対応している最も優先順位の高いモードを選択します。

ユーザーが速度とモードを指定することもできます。SUNW,hme デバイスは、プログラム可能なパケット間隔 ("IPG": Inter-Packet Gap) パラメタの ipg1 および ipg2 に対応しています。デフォルトでは、ドライバは ipg1 を 8 バイト時間に設定し、ipg2 を 4 バイト時間に設定します。これらは標準的な値です。ユーザーは、ドライバが 10 Mbps と 100 Mbps のどちらに対応しているかによって IPG パラメタの値を変更する場合があります。IPG は 9.6 または 0.96 マイクロ秒に設定されます。

m64(7D)

名前	m64 - PGX、PGX24、PGX64 フレームバッファードバイスドライバ
機能説明	m64 ドライバは、Sun PGX グラフィックアクセラレータデバイスドライバです。
ファイル	/dev/fbs/m64 <i>n</i> デバイス特殊ファイル
関連項目	m64config(1M)

名前	pm – 電源管理システムのドライバ
形式	/dev/pm
機能説明	電源管理システム (pm) のドライバは、アプリケーションが電源管理用にシステム内でデバイスを設定するインタフェースを提供します。インタフェースは、 <code>ioctl(2)</code> コマンドを経由して提供されます。pm ドライバは、 <code>/dev/pm</code> を使用してアクセスすることができます。
電源管理システムのフレームワーク	<p>電源管理システムのフレームワークのモデルは、システムをデバイスの集合体として見ます。各デバイスは部品の集合体であり、部品は電源管理が可能な最小単位です。デバイス内の電源管理が可能な部品は、デバイスドライバの制御によって異なります。</p> <p>電源管理可能な部品は、現在の電源レベルでは <i>busy</i> または <i>idle</i> のどちらかの状態となります。通常、電源管理フレームワークは、<i>idle</i> 状態にある部品を次に低い電源レベルにします。電源管理フレームワークは、部品がしきい値の時間以上アイドル状態にあること、部品が属しているデバイスが依存条件を満たしていることの2つの要素によってこの遷移を判定します。依存関係は、あるデバイスが電源管理される前に別のデバイスが電源管理されている必要がある場合に発生します。依存関係はデバイス単位で発生します。依存関係が存在する場合、デバイスが依存するすべてのデバイスが最初に管理されないかぎり、そのデバイスの部品のいずれも管理することができません。</p> <p>以下のコマンドを使用することによって、アプリケーションは電源管理フレームワークドライバからデバイスの電源管理の制御を受け取り、デバイスの電源レベルの遷移を直接管理することができます。</p>
旧式の IOCTLS	<p>このセクションで記述されているすべての <code>ioctl</code> コマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。新しいコマンドについては、このマニュアルページの「新しい IOCTLS」セクションを参照してください。</p> <p>コマンドのこのセットについて、<code>arg(ioctl(2))</code> を参照) は <code><sys/pm.h></code> で定義された型 <code>pm_request</code> の構造体を指します。</p> <pre>typedef struct { char *who; /* 設定するデバイス */ int select; /* 部品またはデバイスの依存デバイスを選択 */ int level; /* 電源レベルまたはしきい値 */ char *dependent; /* 依存デバイスの名前を格納 */ int size; /* 依存デバイスのバッファの大きさ */ } pm_request;</pre> <p>フィールドには、以下のデータが含まれている必要があります。</p> <p><i>who</i> 設定されるデバイスの名前を指すポインタです。デバイスの専用のファイル名またはデバイスへの物理パスの部分文字列となることもあります。</p> <p><i>select</i> 設定される部品または依存デバイスを指定する正の整数です。番号付けは0で始まります。</p>

pm(7D)

<i>level</i>	秒単位のしきい値または目的の電源レベルを指定する正の整数です。
<i>dependent</i>	このデバイスが依存するデバイスの名前を格納しているバッファを指すポインタです。 <i>who</i> のフィールドと同じ形式を使用します。
<i>size</i>	依存デバイスのバッファの大きさです。

すべてのフィールドが、各コマンドで使用されるわけではありません。

PM_DISABLE_AUTOPM

who で指定されたデバイスに対しては、フレームワークによる電源管理が無効になっています。呼び出し元は、以下のコマンドを使用してこのデバイスの電源を管理します。このコマンドの実行が成功しなかった場合、それに続く PM_SET_CUR_PWR は失敗します。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに PM_DIRECT_PM コマンドを使用してください。

以下にエラーコードを示します。

EBUSY	フレームワークにより、デバイスの電源管理はすでに無効になっています。
EPERM	呼び出し元がスーパーユーザーまたはデバイスの所有者ではありません。

PM_GET_NORM_PWR

who で指定されたデバイスの部品 *select* の通常の電源レベルが返されました。この部品の通常の電力は、この部品が再び使用中になるときにこの部品に対して設定される電源レベルです。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに PM_GET_FULL_POWER コマンドを使用してください。

以下にエラーコードを示します。

EINVAL	デバイスの部品が範囲外です。
EIO	デバイスには、電源管理が可能な部品がありません。

PM_GET_CUR_PWR

who で指定されたデバイスの部品 *select* の現在の電源レベルが返されました。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに PM_GET_CURRENT_POWER コマンドを使用してください。

以下にエラーコードを示します。

EINVAL	デバイスの部品が範囲外です。
EAGAIN	デバイスの部品の電源レベルは、現在わかっていません。

PM_SET_CUR_PWR

who で指定されたデバイスの部品 *select* に対して電源レベルが *level* になりました。*select* が 0 でなく、デバイスの部品 0 が電源レベル 0 にある場合、部品 0 に対

して電源レベルが通常のレベルになります。このデバイスに依存する各デバイスの各部品に対して、電源レベルは通常の電源レベルになります。影響を受ける各デバイスの各祖先の各部品に対して、電源レベルは通常の電源レベルになります。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに `PM_SET_CURRENT_POWER` コマンドを使用してください。

以下にエラーコードを示します。

<code>EINVAL</code>	デバイスの部品が範囲外か、0 より低い電源レベルです。
<code>EIO</code>	デバイスまたはその祖先、またはその依存先またはそれらの祖先に対して電源投入が失敗しました。
<code>EPERM</code>	呼び出し元がスーパーユーザーまたはデバイスの所有者ではありません。

`PM_REENABLE_AUTOPM`

who で指定されたデバイスに対して、フレームワークによる電源管理が再び有効になりました。デフォルトでは、すべての構成されたデバイスはフレームワークによって電源管理されます。このコマンドは、現在のリリースでは旧式であり、将来のリリースで削除される予定です。代わりに `PM_RELEASE_DIRECT_PM` コマンドを使用してください。

以下にエラーコードを示します。

<code>EINVAL</code>	デバイスはすでにフレームワークによって電源管理されています。
<code>EPERM</code>	呼び出し元がスーパーユーザーまたはデバイスの所有者ではありません。

新しい `IOCTLS`

このセクションの `ioctl` コマンドは、前述されているは旧式のコマンドに代わるものです。これらの新しいコマンドは、異なる構造へのポインタを取り、より完全な機能をサポートします。

`ioctl` コマンドのセットについて、`arg(ioctl(2))` を参照) は、`<sys/pm.h>` で定義された型 `pm_req` の構造体を指します。

```
typedef struct pm_req {
    char *physpath; /* 設定するデバイスの物理パス */
                  /* libdevinfo(3) を参照 */
    int component; /* デバイスの部品 */
    int value;     /* 電源レベル、しきい値、またはカウント */
    void *data;    /* コマンドに依存する可変サイズデータ */
    size_t datasize; /* データバッファの大きさ */
} pm_req_t;
```

このフィールドには以下のデータが含まれている必要があります。

<i>physpath</i>	デバイスの物理パスを指すポインタです。libdevinfo(3)devinfo) を参照してください。たとえば、デバイス <code>/devices/pseudo/pm@0:pm</code> では、 <i>physpath</i> の値は <code>/pseudo/pm@0</code> です。
-----------------	---

pm(7D)

<i>component</i>	設定される部品または依存デバイスを指定する正の整数です。番号付けは0から始まります。
<i>value</i>	秒単位のしきい値または目的の電源レベルを指定する正の整数、あるいは指定されているレベルの数値です。
<i>data</i>	このデバイスが依存するデバイスの名前などの可変サイズデータを格納しているか受け取るバッファを指すポインタです。
<i>size</i>	データバッファのサイズ。

すべてのフィールドが、各コマンドで使用されるわけではありません。

PM_DIRECT_PM

physpath で指定されたデバイスに対しては、フレームワークによって行われていた電源管理が無効になっています。呼び出し元は、PM_DIRECT_NOTIFY、PM_GET_TIME_IDLE、PM_GET_CURRENT_POWER、PM_GET_FULL_POWER および PM_SET_CURRENT_POWER コマンドを使用してこのデバイスの電源を管理します。「このドライバが、pm_raise_power(9F)、pm_lower_power(9F)、pm_power_has_changed(9F) のいずれかを呼び出す」、または「このデバイスが電源レベルが変化している別のデバイスの親であるか、このデバイスが依存しているデバイスの電源レベルが変化している」かのどちらかの理由により、このデバイスで電源レベルの変更が必要になる場合には、電源レベルの変更は行われず、呼び出し元は、以下に PM_DIRECT_NOTIFY コマンドについて記述されているように電源レベルの変化について通知されます。

以下にエラーコードを示します。

EBUSY	フレームワークにより、デバイスの電源管理はすでに無効になっています。
EPERM	呼び出し元がスーパーユーザーでないか、または実効グループ ID が 0 ではありません。

PM_RELEASE_DIRECT_PM

physpath (PM_DIRECT_PM コマンドのターゲットである必要がある) で指定されたデバイスに対して、フレームワークによる電源管理が再び有効になりました。

以下にエラーコードを示します。

EINVAL	デバイスの部品が範囲外です。
--------	----------------

PM_DIRECT_NOTIFY PM_DIRECT_NOTIFY_WAIT

これらのコマンドは、デバイスを直接電源管理しているプロセスに、このデバイスの電源レベルが変更されるかも知れないイベントについて通知します。このようなイベントが起こった場合は、このコマンドは該当イベントについての情報を返します。

arg (*ioctl*(2) を参照) は、以下のように <sys/pm.h> で定義された型 pm_state_change の構造体を指します。

```
typedef struct pm_state_change {
    char    *physpath;    /* 状態が変化したデバイス */
    int     component;    /* どの部品の状態が変化したか */
}
```

```

#if defined(_BIG_ENDIAN)
    ushort_t flags;      /* PSC_EVENT_LOST, PSC_ALL_LOWEST */
    ushort_t event;     /* イベントの型 */
#else
    ushort_t event;     /* イベントの型 */
    ushort_t flags;     /* PSC_EVENT_LOST, PSC_ALL_LOWEST */
#endif
time_t timestamp;     /* 状態の変化した時刻 */+
int old_level;        /* 変更前の電源レベル */
int new_level;        /* 変更後の電源レベル */
size_t size;         /* physpath が指すバッファの大きさ */
} pm_state_change_t;

```

イベントが起こった場合は、*arg* が指す構造体に挿入されます。このイベントの型が PSC_PENDING_CHANGE の場合は、構造体の残りの部分の情報には、このデバイスが呼び出し元により直接電源管理されない場合にフレームワークがとるアクションが記述されます。呼び出し元は、以下に示す PM_SET_CURRENT_POWER を使用して指定された電源レベルの変更を完全に行います。

PSC_HAS_CHANGED のイベントの型は、デバイスを直接電源管理するドライバがそのデバイスの電源の変化のために、`pm_power_has_changed(9F)` を呼んだことを示します。これにより呼び出し元は、デバイスの電源の状態を追跡することができます。

システムは、イベントを循環バッファに入れます。バッファがオーバーフローした場合、最も古いイベントが削除され、削除されたイベントの次に古いイベントが取り出された時に、PSC_EVENT_LOST フラグが設定されます。

保留中のイベントがない場合は、PM_DIRECT_NOTIFY は、EWOULDBLOCK を返します。また PM_DIRECT_NOTIFY_WAIT は、イベントが使用できるようになるまで実行されません。

pm は、poll(2) インタフェースもサポートします。イベントが保留中の場合、`/dev/pm` のファイル記述子を含み、かつイベントマスクに設定された POLLIN または POLLRDNORM を持つ poll(2) 呼び出しが返されます。

PM_SET_CURRENT_POWER

physpath (PM_DIRECT_PM コマンドを発行したプロセスに対するデバイスの物理パス) で指定されたデバイスの部品である *component* は、電源レベルが *value* に設定されています。*physpath* で指定されたデバイスのすべての部品の電源レベルが 0 で、*value* の値が 0 ではなく、しかも、このデバイスに依存するデバイスがある場合は、このコマンドが返される前に、デバイスのすべての部品がフルパワーになります。同様に、ターゲットデバイスの親の電源が切断されている場合、そのデバイスはこのコマンドが返される前に必要に応じて起動します。PM_SET_CURRENT_POWER がデバイスに対して発行されると、結果として発生する電源の変更が、PM_DIRECT_NOTIFY のイベントリストに追加されます。

以下にエラーコードを示します。

EINVAL	デバイスの部品が範囲外か、0 より低い電源レベルです。
EIO	デバイスまたはその祖先、またはその依存先またはそれらの祖先に対して電源投入が失敗しました。部品が <i>busy</i> の状態になっ

pm(7D)

たためにデバイスドライバがコマンドを不適切なものとして拒否した可能性があるため、このことは障害を示しているとは限らないことに注意してください。

EPERM 呼び出し元がこのデバイスに対して以前に行った `PM_DIRECT_PM` コマンドの実行が成功していません。

PM_GET_FULL_POWER
`physpath` で指定されたデバイスの部品 `component` の使用可能なもっとも高い電源レベルが返されました。

PM_GET_CURRENT_POWER
`physpath` で指定されたデバイスの部品 `component` の現在の電源レベルが返されました。

以下にエラーコードを示します。

EAGAIN デバイス部品の電源レベルは、現在わかっていません。

PM_GET_TIME_IDLE
`PM_GET_TIME_IDLE` は、`physpath` で指定されたデバイスの部品 `component` が `idel` 状態になっている秒数を返します。デバイスが `idel` 状態でない場合は、0 が返されます。

プロセスが `PM_GET_TIME_IDLE` コマンドと `PM_SET_CURRENT_POWER` コマンドを実行する間に `idel` 状態にある部品の電源レベルを小さくするために、デバイスの状態が変化することがあります。そのためにプロセスは、`PM_SET_CURRENT_POWER` コマンドの実行が成功しなかったと返された場合に対処できる必要があります。これはデバイスの部品が `busy` の状態になったために、デバイスがコマンドを不適切であるとして拒否することがあるためです。`PM_GET_TIME_IDLE` コマンドを再び実行して、その部品が `busy` の状態であるかどうかを判断することで、ほかのタイプの障害と区別することができます。

エラー エラーが起きると、コマンドは -1 を返し、`errno` を設定します。上記のコマンドごとのエラーコードに加えて、すべてのコマンドに共通するエラーコードを以下に示します。

EFAULT 引数に指定したアドレスが間違っています。

ENODEV デバイスは電源管理することができないか、構成されていません。

ENXIO デバイスを開く試行回数が多すぎます。

属性 以下の属性については、`attributes(5)` を参照してください。

属性タイプ	属性値
インタフェースの安定性	不安定 (廃止された <code>IOCTLS</code> のインタフェースは使用できません。)

pm(7D)

関連項目 | pmconfig(1M), intro(2), ioctl(2), power.conf(4), attributes(5), attach(9E), detach(9E), power(9E), ddi_dev_is_needed(9F), pm_busy_component(9F), pm_create_components(9F), pm_destroy_components(9F), , pm_idle_component(9F), pm_lower_power(9F), pm_power_has_changed(9F), pm_raise_power(9F)

『*Writing Device Drivers*』

qfe(7d)

名前	qfe – SUNW,qfe Quad Fast-Ethernet デバイスドライバ
形式	/dev/qfe
機能説明	<p>SUNW,qfe Quad Fast-Ethernet ドライバは、マルチスレッド化された、読み込みや複製が可能な STREAMS ハードウェアドライバです。このドライバでは、SUNW,qfe Quad Fast-Ethernet コントローラにおいてコネクションレス型のデータリンクプロバイダインタフェース (dlpi(7P)) を使用することができます。このドライバでは、システムにインストールされた複数の SUNW,qfe コントローラを使用することができます。qfe ドライバは、SUNW,qfe ハードウェアの基本的な機能を提供します。このドライバは、SUNW,qfe デバイスを扱うために使用します。機能には、チップの初期化、フレームの転送と受信、マルチキャストとプロミスキュアスへの対応、エラーからの回復および報告が含まれます。</p>
SUNW,qfe	<p>SUNW,qfe デバイスは、100Base-TX ネットワークインタフェースを提供します。SUNW,qfe デバイスには、SBus に対応しているものと、PCI バスインタフェースに対応しているものの 2 種類があります。SBus 対応の SUNW,qfe デバイスは、サンの FEPS ASIC(SBus インタフェースと MAC 機能を提供します) を使用します。PCI 対応の SUNW,qfe デバイスは、サンの PFEX ASIC を使用して PCI インタフェースと MAC 機能を提供します。両方のデバイスとも 100Base-TX オンボードトランシーバに接続します。100Base-TX オンボードトランシーバは、RJ45 コネクタに接続し、物理層の機能と外部との接続を提供します。</p> <p>100Base-TX 規格によって、操作のモードと速度を自動的に選択するための「自動ネゴシエーション」プロトコルを指定されます。内部トランシーバは、接続の遠隔地側(接続相手)と「自動ネゴシエーション」を行うことができ、遠隔地側の機能に関する情報を受け取ります。内部トランシーバは、属性に基づいて、「最も優先順位の高い共通のモード」を選択します。また、ドライバが操作のモードを選択することができる場合は、強制モードを使用することができます。</p>
アプリケーション プログラミング インタフェース qfe and DLPI	<p>複製された文字型特殊デバイスの /dev/qfe は、システムにインストールされたすべての SUNW,qfe コントローラに接続するために使用します。</p> <p>qfe ドライバは、「スタイル 2」のデータリンクサービスプロバイダです。すべての M_PROTO 型および M_PCPROTO 型のメッセージは、DLPI 基本式として解釈されません。有効な DLPI 基本式は <sys/dlpi.h> の中で定義されています。詳細は、dlpi(7P) を参照してください。開いているストリームに特殊デバイス (ppa) を関連付けるには、ユーザーからの明示的な DL_ATTACH_REQ メッセージが必要となります。この ppa ID は、符号なしの長整数として伝達され、対応するデバイスインスタンス (ユニット) 番号を示します。ppa フィールドの値がシステムにおける有効なデバイスインスタンス番号に対応していない場合は、エラー (DL_ERROR_ACK) が qfe ドライバによって返されます。デバイスは最初の接続時に初期化され、最後の切り離し時に終了 (停止) します。</p> <p>ユーザーからの DL_INFO_REQ に応答して qfe ドライバが返す DL_INFO_ACK 基本式の値を以下に示します。</p> <ul style="list-style-type: none">■ SDU の最大値は 1500 になります (<sys/ethernet.h> で定義されている ETHERMTU)。■ SDU の最小値は 0 です。

- dlsap アドレス長は 8 です。
- MAC 形式は DL_ETHER です。
- sap 長の値は -2 で、これは、DLSAP アドレスの中で物理アドレス構成要素のすぐ後に 2 バイト長の sap 構成要素が位置することを意味します。
- サービスモードは DL_CLDLS です。
- サービスの品質 (QOS: quality of service) オプションは現在のところ含まれていないため、QOS フィールドは 0 です。
- プロバイダスタイルは DL_STYLE2 です。
- バージョンは DL_VERSION_2 です。
- ブロードキャストアドレスの値は Ethernet/IEEE のブロードキャストアドレス (0xFFFFFFFF) です。

DL_ATTACHED 状態になると、特定の SAP (*Service Access Pointer*) をストリームと関連付けるために、ユーザーは DL_BIND_REQ を送る必要があります。qfe ドライバは、DL_BIND_REQ 内の sap フィールドを Ethernet の「タイプ」として解釈します。したがって、sap フィールドの有効な値は [0-0xFFFF] の範囲になります。一度にストリームに接続できるのは 1 つのタイプの Ethernet だけです。

ユーザーが sap の値として 0 を選択すると、受信側は 802.3 モードになります。タイプフィールドが [0-1500] の範囲内にある媒体から受信したすべてのフレームは 802.3 フレームとみなされ、sap 値が 0 のすべての開いているストリームに配信されます。複数のストリームが 802.3 モードにある場合は、フレームは複製され、DL_UNITDATA_IND メッセージとして複数のストリームに配信されます。

qfe ドライバは、送信の際に DL_BIND_REQ の sap フィールドの値が 0 であるかどうかと、宛先タイプフィールドが [0-1500] の範囲内であるかどうかを確認します。どちらかの条件が満たされている場合、ドライバは最初の M_PROTO mblk (メッセージブロック) を除いたすべての後続の DL_UNITDATA_REQ メッセージの長さを計算し、MAC フレームヘッダー長フィールドが計算結果である 802.3 フレームを送信します。

qfe ドライバの DLSAP アドレス形式は、6 バイトの物理 (Ethernet) アドレス部分と、それに続く 2 バイトの sap (タイプ) 部分とで構成されます。アプリケーションでは、DLSAP アドレスの作成や削除を行う際に、実装に依存した DLSAP アドレス形式を明示的に使用せずに、DL_INFO_ACK 基本式で返される情報を使用してください。sap 長、完全な DLSAP 長、sap/物理の順序は DL_INFO_ACK に含まれます。物理アドレス長は、完全な DLSAP アドレス長から sap 長を引くか、DL_PHYS_ADDR_REQ を発行して、ストリームに対応する現在の物理アドレスを取得することによって計算することができます。

DL_BOUND 状態になると、qfe ドライバに DL_UNITDATA_REQ メッセージを送信することによって、ユーザーは Ethernet 上でフレームを送信することができます。qfe ドライバは、DL_UNITDATA_IND メッセージの Ethernet のタイプと一致する sap 値を持ち、開いていて結合されているストリームすべてに、受信された Ethernet フレーム

qfe(7d)

qfe の基本式

ムを配信します。受信された Ethernet フレームは複製され、必要に応じて複数の開いているストリームに配信されます。DL_UNITDATA_REQ および DL_UNITDATA_IND メッセージに含まれる DLSAP アドレスは、sap (タイプ) と物理 (Ethernet) 部分で構成されます。

qfe ドライバは、強制的なコネクションレス DLPI メッセージセットに加えて、以下の基本式に対応しています。

DL_ENABMULTI_REQ および DL_DISABMULTI_REQ 基本式は、個々のマルチキャストグループアドレスの受信を有効または無効にします。これらの基本式を使用して、一群のマルチキャストアドレスをストリームごとに繰り返し作成および変更することができます。これらの基本式は、DL_ATTACHED 後の状態にある場合に、qfe ドライバによって受け付けられます。

dl_level フィールド内に DL_PROMISC_PHYS フラグが設定された状態での DL_PROMISCON_REQ および DL_PROMISCOFF_REQ の基本式は、ローカルホストによって生成されたフレームを含む、媒体上のすべての (「プロミスキュアモード」の) フレームの受信を有効または無効にします。

DL_PROMISC_SAP フラグが設定されている状態で使用すると、sap 値 (Ethernet のタイプ) の受信を有効または無効にします。DL_PROMISC_MULTI フラグが設定されている状態で使用すると、すべてのマルチキャストグループアドレスの受信を有効または無効にします。どちら効果も、常にストリームごとに適用され、当該ストリームやその他のストリームの sap および物理レベルの設定の影響は受けません。

DL_PHYS_ADDR_REQ 基本式は、DL_PHYS_ADDR_ACK 基本式の中のストリームに対応する (接続されている) Ethernet アドレス (6 桁の数) を返します。この基本式は、DL_ATTACH_REQ の成功後の状態においてのみ有効です。

DL_SET_PHYS_ADDR_REQ 基本式は、このストリームに対応する (接続されている) Ethernet アドレスを変更します。このストリームを最初に開く処理は、root でのみ行うことができます。root でない場合は、DL_ERROR_ACK で EPERM が返されます。この基本式は、このデバイスに接続された現在および未来のすべてのストリームに影響する意味で破壊的です。この基本式がこのストリームに対して成功すると、このデバイスに接続された他のすべてのストリームに M_ERROR が送信されます。物理アドレスが一度変更されると、その後開かれ、このデバイスに接続されたストリームは、この新しい物理アドレスを取得します。物理アドレスが一度変更されると、物理アドレスを再び変更するためにこの基本式が使用されるか、システムが再起動されるまでこの物理アドレスは存続します。

qfe ドライバ

デフォルトでは、qfe ドライバは接続のモードおよび速度を選択するために「自動ネゴシエーション」を実行します。

接続は、以下の4つのモードのいずれかになります。

- 100 Mbps、全二重
- 100 Mbps、半二重
- 10 Mbps、全二重
- 10 Mbps、半二重

これらの速度とモードについては、100Base-TX 規格に記述されています。

自動ネゴシエーションプロトコルは以下を自動的に選択します。

- 動作モード (全二重または半二重)
- 速度 (100 Mbps または 10 Mbps)

自動ネゴシエーションプロトコルは以下を行います。

- 接続相手に対応しているすべての動作モードを取得します。
- 接続相手に対して自身の機能を通知します。
- 優先順位に基づいて最も優先順位の高い共通のモードを選択します。
- 最も高い優先順位は 100 Mbps の全二重で、最も低い優先順位は 10 Mbps の半二重です。

100Base-TX トランシーバは、上記のすべての動作モードおよび速度に対応しています。デフォルトでは、接続のモードおよび速度と、接続相手との共通の動作モードを選択するために自動ネゴシエーションが使用されます。

ユーザーが速度とモードを指定することもできます。SUNW,qfe デバイスは、プログラム可能なパケット間隔 (IPG: Inter-Packet Gap) パラメタの ipg1 および ipg2 に対応しています。デフォルトでは、ドライバは ipg1 を 8 バイト時間に設定し、ipg2 を 4 バイト時間に設定します。これらは標準的な値です。ユーザーは、ドライバが 10 Mbps と 100 Mbps のどちらに対応しているかによって IPG パラメタの値を変更する場合があります。IPG は 9.6 または 0.96 マイクロ秒に設定されます。

qfe のパラメタ

qfe ドライバでは、SUNW,qfe デバイスに関するさまざまなパラメタを設定および取得することができます。これらのパラメタには次の内容が含まれます。

- 現在のトランシーバの状態
- 現在の接続の状態
- パケット間隔
- ローカルトランシーバの機能
- 接続相手の機能

ローカルトランシーバには 2 組の機能があります。1 つ目の組はハードウェアの性能に関するもので、読み取り専用 (RO) のパラメタです。もう 1 つの組は、ユーザーが選択する値に関するもので、「速度の選択」で使用されます。読み取り・書き込み (RW) 機能もあります。起動時には、これら 2 組の機能は同じです。接続相手の機能も読み取り専用のパラメタです。これは、これらのパラメタの現在のデフォルト値が読み取り専用で、変更することができないためです。

ファイル	/dev/qfe	qfe 文字型特殊デバイス
	/kernel/drv/qfe.conf	システム共通のデフォルトデバイスドライバ属性
関連項目	nnd(1M), netstat(1M), driver.conf(4), dlpi(7P), le(7D)	

qfe(7d)