



SunOS リファレンスマニュアル 2 : システムコール

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

Part No: 816-3988-10
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 2 : System Calls*

Part No: 816-0212-10

Revision A



020324@2851



目次

はじめに 5

SunOS リファレンスマニュアル 2: システムコール 9

Intro(2) 10

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 2: シ ステムコール

Intro(2)

名前	Intro, intro – システムコールとエラー番号の紹介
形式	#include <errno.h>
機能説明	<p>このセクションでは、全システムコールについて説明します。大多数のシステムコールは、通常はあり得ない値を返すことによってエラー条件を示します。エラー条件を示す一般的な値は、-1 または NULL ポインタです。詳細については、個々のシステムコールの説明を参照してください。エラー発生時には、外部変数 <code>errno</code> にエラー番号が与えられます。システムコールが正常に終了しても <code>errno</code> はクリアされないため、この変数はエラーが返されたときにだけ確認してください。</p> <p>マルチスレッドアプリケーションを使用する場合は、コンパイル時に <code>-mt</code> オプションをコマンド行に指定する必要があります (<code>threads(3THR)</code> のマニュアルページを参照)。 <code>-mt</code> オプションを指定すると、<code>errno</code> はマクロになり、各スレッドが独自の <code>errno</code> を持つことができるようになります。この <code>errno</code> マクロは、変数と同様に、代入文の左辺と右辺のどちらにも使用できます。</p> <p>アプリケーションは、 <code>_lwp_*()</code> 関数ではなく結合スレッドを使用する必要があります (<code>thr_create(3THR)</code> 参照)。ライブラリはスレッドで使用した場合に限り安全なので、LWP (軽量プロセス) を直接使用することはお勧めできません。</p> <p>このセクションでは、各システムコールによって生成される可能性があるエラー番号をすべて紹介します。次に、全エラー番号と、そのエラー名 (<errno.h> 中に定義) を示します。</p> <p>1 EPERM スーパーユーザーではありません。</p> <p>通常、ファイルの所有者やスーパーユーザーにのみ許可されている更新処理を、他のユーザーが行おうとした場合に返されます。スーパーユーザーにのみ許可されている処理を一般のユーザーが実行しようとした場合にも、このエラーが返されます。</p> <p>2 ENOENT ファイルまたはディレクトリが存在しません。</p> <p>指定された名前を持つファイルが存在すべきなのに存在しない場合や、パス名に存在しないディレクトリが指定されている場合に返されます。</p> <p>3 ESRCH プロセス、LWP、またはスレッドが存在しません。</p> <p>指定された PID、LWPID_t、または <code>thread_t</code> に該当するプロセスが、システム上に存在しません。</p> <p>4 EINTR システムコールに割り込みが発生しました。</p> <p>システムサービス関数の実行中に、ユーザーがキャッチするように指定した非同期シグナル (割り込みや終了など) が発生しました。シグナルの処理後にシステム関数の実行を再開すると、この関数へのコールがエラーを返したかのように見えます。</p>

	マルチスレッドアプリケーションでは、他のスレッドや LWP が <code>fork(2)</code> を呼び出すと、 <code>EINTR</code> が返されることがあります。
5 EIO	<p>入出力エラー。</p> <p>物理的な入出力エラーが発生しました。場合によっては、実際に入出力エラーを起こしたコールの直後のコールで、このエラーが返されることがあります。</p>
6 ENXIO	<p>デバイスまたはアドレスが存在しません。</p> <p>特殊ファイルへの入出力を指定したが、該当するサブデバイスが存在しない場合や、デバイスの限度を超えている場合に返されます。テープデバイスがオンライン状態になっていない場合や、ディスクデバイスにディスクバックが読み込まれていない場合にも、このエラーが返されます。</p>
7 E2BIG	<p>引数リストが長すぎます。</p> <p><code>exec</code> ファミリの関数に、<code>ARG_MAX</code> で指定されているバイト数より長い引数リストが渡されました (<code>exec(2)</code> 参照)。引数リストの最大長は、実際の引数リストの長さ、その環境でエクスポートされたシェル変数の長さの合計値です。</p>
8 ENOEXEC	<p>実行ファイル形式のエラー。</p> <p>実行要求のあったファイルのアクセス権は適切であるが形式が正しくない場合に返されます (<code>a.out(4)</code> 参照)。</p>
9 EBADF	<p>ファイル番号が正しくありません。</p> <p>ファイル記述子で定義されたファイルが開いていません。書き込み専用で開いたファイルへの読み取り要求が発行された場合や (<code>read(2)</code>)、読み取り専用で開いたファイルへの書き込み要求が発行された場合 (<code>write(2)</code>) にも、このエラーが返されます。</p>
10 ECHILD	<p>子プロセスがありません。</p> <p><code>wait(2)</code> 関数が実行されたが、これを実行したプロセスには子プロセスがないか、用意されていませんでした。</p>
11 EAGAIN	<p>これ以上のプロセスまたは LWP を処理できません。</p>

12 ENOMEM	<p>システムのプロセステーブルがいっぱいであるか、ユーザーがこれ以上のプロセスの生成を許可されていないことが原因で、<code>fork(2)</code> 関数の実行に失敗しました。メモリー不足やスワップ領域不足が原因で、コールに失敗した場合にも、このエラーが返されます。</p> <p>領域が不足しています。</p>
13 EACCES	<p><code>brk()</code>、<code>sbrk()</code> (<code>brk(2)</code> を参照)、または <code>exec</code> ファミリの関数の実行中に、プログラムから、システムによる割り当て可能限度を超過した領域が要求されました。割り当て可能領域の最大値は一時的な値ではなく、システムパラメータとして定義されています。アーキテクチャーによっては、テキストセグメント、データセグメント、スタックセグメントなどから過剰なセグメンテーションレジスタが要求された場合や、<code>fork(2)</code> 関数の実行中にスワップ領域が不足した場合にも、このエラーが発生します。リモートファイルシェアリング (RFS) に関連したリソースでこのエラーが発生する場合は、システムコール発行時のシステムの使用状況が原因で、一時的なメモリー不足が発生している可能性があります。</p> <p>アクセス権がありません。</p>
14 EFAULT	<p>ファイルをアクセスしようとしたが、その方法は、ファイル保護システムで許可されていません。</p> <p>不正なアドレスです。</p>
15 ENOTBLK	<p>ルーチンの引数を使用するときに、ハードウェア障害を検出しました。たとえば、ポインタ引数を使用するルーチンに不正なアドレスが与えられたとき、この状態がシステムによって検出されると、<code>errno</code> の値として <code>EFAULT</code> が設定されることがあります。不正なアドレスの検出機能はシステムによって異なるので、ルーチンに不正なアドレスが渡された場合の動作は一定ではありません。</p> <p>ブロック型デバイスを指定してください。</p>
16 EBUSY	<p>ブロック型デバイスを指定しなければならない個所に、ブロック型以外のデバイスまたはファイルが指定されました (例: <code>mount(2)</code> 関数の呼び出し時)。</p> <p>デバイスは使用中です。</p> <p>すでにマウントされているデバイスをマウントしようとしたか、アクティブなファイル (開いた状態のファイル、現在のディレクトリ、マウントされているファイ</p>

	<p>ル、使用中のテキストセグメントなど)が存在しているデバイスのマウントを解除しようとした。すでに使用可能な状態になっているアカウンティングをさらに使用可能にしようとした場合にも、このエラーが発生します。このとき、デバイスやリソースは使用できません。EBUSY は、相互排他ロック、セマフォ、条件付き変数、読み取り / 書き込みロックでも使用され、ロックが保持されていることを示します。また、プロセス制御関数 P_ONLINE でも使用されます。</p>
17 EEXIST	<p>ファイルが存在します。</p> <p>不適切なコンテキストで既存のファイル名が指定されました (例: link(2) 関数の呼び出し時)。</p>
18 EXDEV	<p>クロスデバイスリンクです。</p> <p>別のデバイス上のファイルへのハードリンクを試みました。</p>
19 ENODEV	<p>デバイスが存在しません。</p> <p>デバイスに対する操作が不適切です (例: 書き込み専用デバイスへの読み取り要求)。</p>
20 ENOTDIR	<p>ディレクトリではありません。</p> <p>ディレクトリ名を指定しなければならない個所 (例: chdir(2) 関数の引数、パスの開始部分) に、ディレクトリ以外が指定されました。</p>
21 EISDIR	<p>ディレクトリです。</p> <p>ディレクトリに書き込みを実行しようとした。</p>
22 EINVAL	<p>不正な引数です。</p> <p>不正な引数が指定されました。たとえば、マウントされていないデバイスをマウント解除しようとした。あるいは、signal(3C) または kill(2) 関数の呼び出しにおいて未定義のシグナルが使用されたか、拡張属性に対してサポートされていない操作が行われました。</p>
23 ENFILE	<p>ファイルテーブルのオーバーフローが発生しました。</p> <p>システムのファイルテーブルがいっぱいです。SYS_OPEN で指定された数のファイルがすでに開いていて、これ以上のファイルを開くことができません。</p>
24 EMFILE	<p>開いたファイルの数が多すぎます。</p>

	<p>プロセスが、OPEN_MAX ファイル記述子に指定されている数以上のファイルを開くことは、許可されていません。</p>
25 ENOTTY	<p>デバイスの入出力制御 (ioctl) が正しくありません。</p>
	<p>特殊文字デバイス以外のファイルを指定して ioctl(2) 関数を呼び出しました。</p>
26 ETXTBSY	<p>テキストファイルは使用中です (廃止)。</p>
	<p>書き込み用に開いた、プロセスのみからなるプログラムを実行しようとした。プロセスのみからなるプログラムの実行中に、これを書き込み用に開こうとしたり、削除しようとしても、同じエラーが発生します (このメッセージは廃止されました)。</p>
27 EFBIG	<p>ファイルが大きすぎます。</p>
	<p>ファイルのサイズが、RLIMIT_FSIZE で指定されている値を超えました。指定されたファイルは、ファイルシステムがサポートするファイルサイズを超えているか、ファイル記述子のオフセット最大値を超えています。後述の「定義」で、「ファイル記述子」の項を参照してください。</p>
28 ENOSPC	<p>デバイスに空き領域がありません</p>
	<p>通常ファイルへの書き込みまたはディレクトリエントリの作成を試みましたが、デバイス上に空き領域がありません。システム上にレコードエントリが残っていないので、fcntl(2) 関数によるファイル上のレコードロックの設定または削除ができませんでした。</p>
29 ESPIPE	<p>シークが正しくありません。</p>
	<p>lseek(2) 関数の呼び出しがパイプに対して発行されました。</p>
30 EROFS	<p>読み取り専用のファイルシステムです。</p>
	<p>読み取り専用でマウントされたデバイス上で、ファイルまたはディレクトリを更新しようとした。</p>
31 EMLINK	<p>リンクが多すぎます。</p>
	<p>LINK_MAX で指定されている値を超える数のリンクを作成しようとした。</p>
32 EPIPE	<p>パイプが中断されました。</p>

	データを読み込むプロセスが存在しないパイプで書き込みを行おうとしました。この条件では、通常はシグナルが発行され、シグナルが無視されるとエラーが返されます。
33 EDOM	数学引数が関数の領域定義域外です。
	数学パッケージ (3M) 中の関数の引数が、その関数の定義域外です。
34 ERANGE	数学関数の結果を表示できません。
	数学パッケージ (3M) 中の関数の結果の値は、このマシンの精度では表示できません。
35 ENOMSG	要求された種類のメッセージがありません。
	指定されたメッセージ待ち行列に存在しない種類のメッセージを受信しようとした (<code>msgrcv(2)</code> を参照)。
36 EIDRM	識別子が削除されています。
	このエラーは、ファイルシステムの名前空間から識別子が削除され、その結果、実行を再開するプロセスに対して返されます (<code>msgctl(2)</code> 、 <code>semctl(2)</code> 、 <code>shmctl(2)</code> を参照)。
37 ECHRNG	チャンネル番号が範囲外です。
38 EL2NSYNC	レベル 2 が同期していません。
39 EL3HLT	レベル 3 停止。
40 EL3RST	レベル 3 リセット。
41 ELNRNG	リンク番号が範囲外です。
42 EUNATCH	プロトコルドライバが装備されていません。
43 ENOCSI	CSI 構造体がありません。
44 EL2HLT	レベル 2 停止。
45 EDEADLK	デッドロック状態。
	デッドロック状態を検出したので回避しました。このエラーは、ファイルやレコードのロック処理に付随して発生します。また、相互排他ロック、セマフォ、条件付き変数、読み取り / 書き込みロックでも発生します。
46 ENOLCK	レコードロックが使用できません。

47 ECANCELED	使用可能なロックが存在しません。システムのロックテーブルがいっぱいです (fcntl(2) を参照)。 命令が取り消されました。
48 ENOTSUP	関連する非同期命令が、未完了のまま取り消されました。 サポートされていません。
49 EDQUOT	このバージョンのシステムではこのサポートされていない機能です。将来のバージョンではサポートされる可能性があります。 ディスク容量の制限を超えました。
58-59	予約領域。
60 ENOSTR	ストリームデバイスではありません。 STREAMS デバイスではないファイル記述子に対して、putmsg(2) または getmsg(2) システムコールを試みました。
61 ENODATA	データがありません。
62 ETIME	タイマーが終了しました。 STREAMS の ioctl(2) コールに設定した時間が経過しました。このエラーの原因はデバイスによって異なり、ハードウェアまたはソフトウェアに障害が発生したか、設定した時間が要求した操作に対して短すぎたことが考えられます。ioctl() の処理状態は確定できません。_lwp_cond_timedwait(2) や cond_timedwait(3thr) でも、同じエラーが返されません。
63 ENOSR	ストリーム資源が不足しています。 STREAMS の open(2) コールで、STREAMS 待ち行列または STREAMS ヘッドのデータ構造体を使用できませんでした。このエラー状態は一時的なもので、他のプロセスがリソースを解放すれば回復できる場合があります。

64 ENONET	<p>マシンがネットワーク上にありません。</p> <p>リモートファイルシェアリング (RFS) 固有のエラーです。起動時にネットワークに接続するように設定しなかったマシン上で、リモートリソースのアドバタイズ、アンアドバタイズ、マウント、またはマウント解除を試みました。</p>
65 ENOPKG	<p>パッケージがインストールされていません。</p> <p>インストールされていないパッケージのコールをおうとしました。</p>
66 EREMOTE	<p>リモートオブジェクトです。</p> <p>リモートファイルシェアリング (RFS) 固有のエラーです。ローカルマシン上にないリソースをアドバタイズしようとしたか、リモートマシン上のデバイス (またはパス名) をマウントまたはマウント解除しようとした。</p>
67 ENOLINK	<p>リンクが切断されました。</p> <p>リモートファイルシェアリング (RFS) 固有のエラーです。リモートマシンと接続していたリンク (仮想回線) が切断されました。</p>
68 EADV	<p>アドバタイズエラー。</p> <p>リモートファイルシェアリング (RFS) 固有のエラーです。アドバタイズ済みのリソースをアドバタイズしようとした。または、リソースのアドバタイズ中に RFS を停止しようとしたか、アドバタイズ中のリソースのマウントを強制的に解除しようとした。</p>
69 ESRMNT	<p>Srmount エラー。</p> <p>リモートファイルシェアリング (RFS) 固有のエラーです。リモートマシンが、マウントしたリソースが存在しているのに RFS を停止しようとした。または、現在リソースをマウントしているリモートマシンが含まれていないクライアントリストを使って、リソースを再アドバタイズしようとした。</p>
70 ECOMM	<p>送信時に通信エラーが発生しました。</p> <p>リモートファイルシェアリング (RFS) 固有のエラーです。現プロセスがリモートマシンからのメッセージを待っているときに、仮想回線に障害が発生しました。</p>
71 EPROTO	<p>プロトコルエラー。</p>

	何らかのプロトコルエラーが発生しました。このエラーはデバイス固有のものですが、通常、ハードウェアの障害とは関係がありません。
76 EDOTDOT	エラー 76。 リモートファイルシェアリング (RFS) 固有のエラーです。プロセスをマウントポイントから転送し直したことを、サーバーがクライアントに伝える手段として使います。
77 EBADMSG	データメッセージではありません。 STREAMS デバイスへの read(2)、getmsg(2)、または ioctl(2) I_RECVFD コール中に、待ち行列の先頭に処理できない対象が置かれました。これは、次に示すようにコールによって異なります。 read() 制御情報、または渡されたファイル記述子 getmsg() 渡されたファイル記述子 ioctl() 制御情報またはデータ情報
78 ENAMETOOLONG	ファイル名が長すぎます。 パス引数の長さが PATH_MAX で指定されている最大長を超えているか、_POSIX_NO_TRUNC が有効であるのにパス名を構成する名前の 1 つが NAME_MAX で指定されている値より長すぎました。最大長については limits(4) を参照してください。
79 EOVERFLOW	定義されたデータ型の値が大きすぎます。
80 ENOTUNIQ	同じ名前がネットワーク上に複数存在します。 指定されたログ名が、ネットワーク上で一意ではありません。
81 EBADFD	ファイル記述子が正しくありません。 ファイル記述子が開いたファイルを参照していないか、書き込み専用で開いたファイルに対して読み取り要求を発行しました。
82 EREMCHG	リモートアドレスが変更されています。
83 ELIBACC	必要な共有ライブラリにアクセスできません。

84 ELIBBAD	<p>a.out の実行には共有静的ライブラリが必要ですが、ライブラリが存在しませんでした。または、ユーザーには、そのライブラリを使用するアクセス権がありません。</p> <p>共有ライブラリが壊れています。</p>
85 ELIBSCN	<p>a.out の実行には共有静的ライブラリが必要 (リンク対象) ですが、exec は、このライブラリを読み込めませんでした。ライブラリが壊れている可能性があります。</p> <p>a.out ファイルの .lib 部分が壊れています。</p>
86 ELIBMAX	<p>a.out の実行には共有静的ライブラリが必要 (リンク対象) ですが、a.out 中の .lib 部分に不正なデータがありました。 .lib 部分は、必要な共有ライブラリを exec に伝える役割を果たします。エラーの原因としては、a.out ファイルが壊れている可能性があります。</p> <p>システムの限界数以上の共有ライブラリをリンクしようとしています。</p>
87 ELIBEXEC	<p>実行しようとした a.out ファイルには、現在システムに定義されている許容最大値を超える数の共有ライブラリが必要です。詳細については、『Solaris のシステム管理 (IP サービス)』を参照してください。</p> <p>共有ライブラリは直接実行 (exec) できません。</p>
88 EILSEQ	<p>共有ライブラリを直接実行しようとしてしました。</p> <p>エラー 88。</p> <p>不正なバイトシーケンスを検出しました。複数の文字を 1 文字として扱ってください。</p>
89 ENOSYS	<p>適用不能な操作です。</p>
90 ELOOP	<p>パス名の走査中に見つかったシンボリックリンクの数が、MAXSYMLINKS で義されている値を超えました。</p>
91 ESTART	<p>再開可能なシステムコールです。</p>
92 ESTRPIPE	<p>割り込みを受けたシステムコールは再起動する必要があります。</p> <p>パイプまたは先入れ先出し (FIFO) を使用している場合、ストリームヘッドでは休止できません。</p> <p>ストリームのパイプエラーです。このエラーは外部からは見えません。</p>

Intro(2)

93 ENOTEMPTY	ディレクトリが空ではありません。
94 EUSERS	ユーザー数が多すぎます。
95 ENOTSOCK	ソケットを対象とした操作がソケット以外に適用されました。
96 EDESTADDRREQ	宛先のアドレスが必要です。 トランスポートの終端からの操作で、必要な宛先アドレスが省略されていました。
97 EMSGSIZE	メッセージが長すぎます。 トランスポートプロバイダに送信されたメッセージの長さが、内部のメッセージバッファサイズまたはネットワーク上の制限値を超えていました。
98 EPROTOTYPE	プロトコルの型がソケット用ではありません。 指定されたプロトコルは、要求されたソケットタイプを解釈できません。
99 ENOPROTOOPT	プロトコルを使用できません。 プロトコルの取得、設定時に、不正なオプションまたはレベルが指定されました。
120 EPROTONOSUPPORT	プロトコルがサポートされていません。 指定されたプロトコルがシステムに定義されていないか、そのプロトコル用の実装がありません。
121 ESOCKTNOSUPPORT	ソケットの型がサポートされていません。 指定されたソケットがシステムに定義されていないか、そのソケット用の実装がありません。
122 EOPNOTSUPP	トランスポートの終端でサポートされていない操作です。 たとえば、データグラムของトランスポートの終端で接続を受け付けようとした場合などに、このエラーが発生します。
123 EPFNOSUPPORT	プロトコルファミリがサポートされていません。 指定されたプロトコルファミリがシステムに定義されていません。またはないか、そのプロトコルファミリ用の実装がありません。この状態はインターネットプロトコルに適用されます。
124 EAFNOSUPPORT	プロトコルファミリがアドレスファミリをサポートしていません。

	使用したアドレスには、要求されたプロトコルとの互換性ありません。
125 EADDRINUSE	すでに使用されているアドレスです。
	現在使用中のアドレスを使用しようとしたが、このような指定は、プロトコルで許可されていません。
126 EADDRNOTAVAIL	要求されたアドレスを割り当てられません。
	現在のマシン上に存在しないアドレスを使ってトランスポートの終端を作成しようとした。
127 ENETDOWN	ネットワークがダウンしています。
	指定した操作の実行中に、停止状態のネットワークを検出しました。
128 ENETUNREACH	ネットワークに到達できません。
	到達できないネットワークに対する操作を試みしました。
129 ENETRESET	リセットにより、ネットワーク接続がドロップされました。
	接続先のホストに障害が発生し、レポートされました。
130 ECONNABORTED	ソフトウェアによって接続が強制切断されました。
	ホストマシン内部で、接続が異常終了しました。
131 ECONNRESET	相手側によって接続がリセットされました。
	接続が、相手側から強制的に切断されました。通常この状態は、タイムアウトまたはリポートによりリモートホストとの接続が失われたときに発生します。
132 ENOBUFS	バッファ領域を用できません。
	バッファ領域が不足しているか、待ち行列がいっぱいのため、トランスポートの終端またはパイプでの操作を実行できませんでした。
133 EISCONN	トランスポートの終端は接続されています。
	すでに接続されているトランスポート終端に対して接続要求が発行されました。または、接続されているトランスポート終端上で、接続済みの宛先が指定された <code>sendto(3SOCKET)</code> または <code>sendmsg(3SOCKET)</code> 要求が発行されました。
134 ENOTCONN	トランスポートの終端が接続されていません。

Intro(2)

	トランスポートの終端が接続されていません。また、データグラム送信の場合、アドレスが指定されていません。このため、データの送信要求または受信要求を受け付けられません。
143 ESHUTDOWN	ソケットのシャットダウン後で送信できません。 トランスポート終端がシャットダウン済みなので、データ送信要求は受け付けられません。
144 ETOOMANYREFS	参照が多すぎるため、結合できません。
145 ETIMEDOUT	接続の時間切れです。 相手側が一定時間内に正しい応答を返さなかったため、connect(3SOCKET) 要求または send(3SOCKET) 要求に失敗しました。soft オプションでマウントされた NFS ファイルシステム上にファイルが存在し、write(2) 要求または fsync(3C) 要求に失敗した場合も、同じエラーになります。
146 ECONNREFUSED	接続が拒否されました。 相手側マシンにより拒否されたので、接続を確立できませんでした。通常この状態は、リモートホスト上でアクティブになっていないサービスに接続しようとした場合に発生します。
147 EHOSTDOWN	ホストがダウンしています。
148 EHOSTUNREACH	宛先ホストがダウンしているため、トランスポートプロバイダの処理に失敗しました。 ホストへの経路がありません。 到達不可能なホストに対して、トランスポートプロバイダの処理を試みました。
149 EALREADY	操作中です。 非ブロック型オブジェクトの処理を試みましたが、そのオブジェクトはすでに何らかの処理を実行中でした。
150 EINPROGRESS	操作中です。 非ブロック型オブジェクトに対して、長時間を要する処理 (connect () など) を試みました。
151 ESTALE	NFS のファイルハンドルが無効です。

バックグラウンド プロセスグループ	制御端末との接続を確立したセッションのフォアグラウンドプロセスグループを除くすべてのプロセスグループ。
制御プロセス	制御端末との接続を確立したセッションリーダープロセス。
制御端末	セッションで使用している端末。各セッションでは制御端末を1台だけ使用でき、制御端末1台で1つのセッションを確立できます。制御端末から入力されるある種のシーケンスは、その制御端末を使用しているセッション中のプロセスグループにシグナルを送信します。詳細については <code>termio(7I)</code> を参照してください。
ディレクトリ	ファイルを階層構造システムに組織化する働きを持つ、ファイル階層構造におけるノードです。ディレクトリもファイルの一種であり、一群のファイルや、すぐ下の階層にある他のディレクトリ(サブディレクトリ)をまとめています。また、ディレクトリファイル内の項目はリンクと呼ばれ、ファイル記述子とファイル名を対応付ける役割を果たしています。通常、各ディレクトリには2つのリンク、 <code>.</code> (ドット)と <code>..</code> (ドットドット)が含まれています。ドットはそのディレクトリ自体を指すリンクで、ドットドットはすぐ上の階層のディレクトリ(親ディレクトリ)を指すリンクです。階層構造の頂点にあるルートディレクトリの親ディレクトリは、そのディレクトリ自体です。ルートディレクトリのパス名は <code>/</code> で、親ディレクトリのパス名も <code>/</code> です。
ダウンストリーム	ストリームの方向を表します。ストリームヘッドからドライバへ方向を、ダウンストリームと呼びます。
ドライバ	1つのストリームでは、ドライバが周辺ハードウェア機器とストリームとのインタフェースを提供します。ドライバは、マルチプレクサやログドライバ(<code>log(7D)</code> を参照)のような擬似ドライバの形をとることもあります。疑似ドライバは、ハードウェアデバイスとは無関係です。
実効ユーザー ID と 実効グループ ID	アクティブなプロセスは、実効ユーザー ID と実効グループ ID を1つずつ持っています。プロセスは、この2つの ID を使用して、ファイルへのアクセス権(後述)を検査します。実効ユーザー ID と実効グループ ID は、それぞれ、プロセスの実ユーザー ID および実グループ ID と同一です。ただし、プロセスまたはその祖先がセットユーザー ID またはセットグループ ID のビットが設定されたファイル(<code>exec(2)</code> を参照)から生成されている場合を除きます。
ファイルのアクセス権	次のいずれかの条件が満たされる場合、プロセスに、ファイルに対する読み取り権、書き込み権、実行権 / 検索権が付与されます。 <ul style="list-style-type: none"> ■ プロセスの実効ユーザー ID がスーパーユーザーである場合 ■ プロセスの実効ユーザー ID がファイルの所有者のユーザー ID と一致していて、ファイルに「所有者」モード(0700)のアクセスビットが設定されている場合 ■ プロセスの実効ユーザー ID がファイルの所有者のユーザー ID と一致していなくても、プロセスの実効グループ ID または補助グループ ID がファイルのグループ ID と一致していて、ファイルに「グループ」モード(0070)のアクセスビットが設定されている場合 ■ プロセスの実効ユーザー ID がファイルの所有者のユーザー ID と一致せず、プロセスの実効グループ ID、補助グループ ID のどちらもファイルのグループ ID と一致しないが、ファイルに「その他」モード(0007)のアクセスビットが設定されて

Intro(2)

	いる場合
	上記の条件のいずれもあてはまらない場合、アクセスは拒否されます。
ファイル記述子	ファイルへの入出力の実行に使用する小さな整数値。最小値は 0 で、最大値は <code>NOFILES</code> の値から 1 を引いた値です。プロセスには、 <code>NOFILES</code> 値を超える数のファイル記述子を同時に開くことは許可されていません。ファイル記述子は <code>open(2)</code> や <code>pipe(2)</code> などのコールにより返されます。また <code>read(2)</code> 、 <code>write(2)</code> 、 <code>ioctl(2)</code> 、 <code>close(2)</code> などのコールの引数として使用されます。
	各ファイル記述子には、対応するオフセット最大値があります。 <code>O_LARGEFILE</code> フラグを設定しないで開いた通常ファイルでは、オフセット最大値は 2G バイトより 1 バイト小さい値 ($2^{31}-1$ バイト) です。 <code>O_LARGEFILE</code> フラグを設定して開いた通常ファイルでは、オフセット最大値は 2^{63} バイトよりも 1 バイト小さい値です。
ファイル名	通常ファイル、特殊ファイル、ディレクトリに付与される名前。名前を構成する文字数の最小値は 1 で、最大値は <code>NAME_MAX</code> の値です。
	ファイル名には、 <code>\0</code> (NULL) および <code>/</code> (スラッシュ) の ASCII コードを除くすべての文字を使用できます。
	ただし、 <code>*</code> 、 <code>?</code> 、 <code>[</code> 、 <code>]</code> の 4 文字は、シェルに対して特別な意味を持つため (<code>sh(1)</code> 、 <code>csh(1)</code> 、 <code>ksh(1)</code> を参照)、通常は使用しません。また、許可されている文字であっても、印刷できない文字は、できるだけ使用しないでください。
	ファイル名は、パス名を構成する 1 項目とも考えられます。パス名の各項目の解釈は、その項目の接頭辞に対応した <code>NAME_MAX</code> および <code>_POSIX_NO_TRUNC</code> の値により異なります。このため、パス名に含まれている項目の長さが <code>NAME_MAX</code> 値を超えているとき、その項目のパス接頭辞に対して <code>_POSIX_NO_TRUNC</code> が有効な場合 (<code>fpathconf(2)</code> および <code>limits(4)</code> を参照)、この項目はエラーとみなされます。それ以外の場合、項目の先頭部分 (バイト数は <code>NAME_MAX</code> 値に等しい) が項目名として使用されます。
フォアグラウンドプロセスグループ	制御端末と接続を確立したセッションは、そのセッションのプロセスグループ群のうちの一つを制御端末のフォアグラウンドプロセスグループとして識別します。このプロセスグループは、制御端末にアクセスする際、他のグループ (バックグラウンドプロセスグループ) には許可されない特権を付与されています。
<code>{IOV_MAX}</code>	<code>struct iovec</code> 配列内の項目の最大数。
<code>{LIMIT}</code>	中括弧 <code>{LIMIT}</code> は、実装ごとに決められた大きさの制限を示すために使用されます。これは、 <code>{}</code> を使用せず、ヘッダーファイルによって定義される値、または、 <code>pathconf(2)</code> に引数 <code>_PC_LIMIT</code> を指定して構成の照会を行うことによって実行時に得ることができる実際の値を示しています。
マスク	プロセスのファイルモード生成マスク。 <code>mode</code> 引数で指定されたアクセス権ビットを無効にするような関数生成コール時に使用されます。 <code>umask (cmask)</code> で設定されていたビットは、生成されるファイルのモードではクリアされています。

メッセージ	ストリーム中のデータまたは情報の集まり。ブロック単位で構成され、STREAMS 制御構造体を伴います。メッセージにはいくつかの形式が定義されていて、形式によってその内容の種類が決まります。ストリーム中では、メッセージがデータ送信および通信の唯一の手段です。
メッセージ待ち行列	ストリーム中で、モジュールまたはドライバによる処理を待っているリンクされたメッセージ群。
メッセージ待ち行列識別子	msgget(2) コールによって生成される一意の正の整数で、メッセージを識別します。msgid とも表されます。各 msgid には、メッセージ待ち行列と、対応するデータ構造体が含まれています。データ構造体は msgid_ds と表され、次のメンバーからなります。

```

struct    ipc_perm msg_perm;
struct    msg *msg_first;
struct    msg *msg_last;
ulong_t   msg_cbytes;
ulong_t   msg_qnum;
ulong_t   msg_qbytes;
pid_t     msg_lspid;
pid_t     msg_lrpid;
time_t    msg_stime;
time_t    msg_rtime;
time_t    msg_ctime;

```

次に、msgid_ds 構造体のメンバーについて説明します。

msg_perm メンバーは ipc_perm 構造体の 1 つであり、メッセージ操作のアクセス権を指定します (後述)。この構造体には次のメンバーが含まれています。

```

uid_t     cuid;    /* creator user id */
gid_t     cgid;    /* creator group id */
uid_t     uid;     /* user id */
gid_t     gid;     /* group id */
mode_t    mode;    /* r/w permission */
ulong_t   seq;     /* slot usage sequence # */
key_t     key;     /* key */

```

*msg_first メンバーは、待ち行列内の先頭のメッセージへのポインタです。

*msg_last メンバーは、待ち行列内の最後のメッセージへのポインタです。

msg_cbytes メンバーは、現在の待ち行列の内容の総バイト数です。

msg_qnum メンバーは、現在待ち行列にあるメッセージの数です。

msg_qbytes メンバーは、待ち行列に置くことのできる最大バイト数です。

msg_lspid メンバーは、最後に msgsnd() を実行したプロセスのプロセス ID です。

msg_lrpid メンバーは、最後に msgrcv() を実行したプロセスのプロセス ID です。

Intro(2)

	<p>msg_stime メンバーは、最後に msgsnd() を実行した時刻です。</p> <p>msg_rtime メンバーは、最後に msgrcv() を実行した時刻です。</p> <p>msg_ctime メンバーは、上記構造体のメンバーを変更する msgctl() を最後に実行した時刻です。</p>
メッセージ操作 アクセス権	<p>msgctl(2)、msgget(2)、msgrcv(2)、msgsnd(2) の各関数の説明では、操作に必要なアクセス権を {token} で表しています。この token は、必要なアクセス権の種類であり、次のように解釈されます。</p> <pre>00400 ユーザーによる読み取り 00200 ユーザーによる書き込み 00040 グループによる読み取り 00020 グループによる書き込み 00004 その他による読み取り 00002 その他による書き込み</pre> <p>次のいずれかの条件が満たされるとき、プロセスに、msqid への読み取りおよび書き込み権が付与されます。</p> <ul style="list-style-type: none">■ プロセスの実効ユーザー ID がスーパーユーザーである場合■ プロセスの実効ユーザー ID が msqid に対応するデータ構造体の msg_perm.cuid または msg_perm.uid と一致していて、msg_perm.mode の「ユーザー」部分 (0600) に正しいビットが設定されている場合■ cr_gid および cr_groups の有効プロセスのグループ ID が msg_perm.cgid または msg_perm.gid と一致していて、msg_perm.mode の「グループ」部分 (060) に正しいビットが設定されている場合■ msg_perm.mode の「その他」部分 (006) に正しいビットが設定されている場合 <p>上記の条件がいずれもあてはまらない場合、アクセスは拒否されます。</p>
モジュール	<p>データの入出力用の処理ルーチンの集まり。常にストリームの間、つまりストリームヘッドからドライバまでの間に存在します。ストリームにおけるモジュールは、シェルのパイプラインにおけるコマンドと似ています。ただしモジュールには、双方向 (ダウンストリームとアップストリーム) のデータのフローと処理を独立して実施することを可能にするための 2 つの関数が含まれています。</p>
マルチプレクサ	<p>複数のユーザープロセスに対応したストリーム群を 1 つのドライバに接続したり、複数のドライバを 1 つのユーザープロセスに接続することを可能にするドライバ。STREAMS は一般的な多重化用ドライバを備えてはいませんが、ストリームを構築したり多重化されたストリーム群を接続したりする機能は提供します。</p>
オフセット最大値	<p>オフセット最大値は、開いたファイルの記述の属性で、ファイルオフセットとして使用できる最大値を表します。</p>
親のないプロセス グループ	<p>グループ内の全メンバーの親が、そのグループのメンバーか、グループのセッションのメンバーではないようなプロセスグループ。</p>

パス名	<p>スラッシュ (/) で区切られたディレクトリ名の集まりで、NULL 文字で終わる文字列。先頭がスラッシュの場合や、最後がファイル名の場合もあります。</p> <p>パス名の先頭がスラッシュの場合、パスの検索はルートディレクトリから始まります。先頭がスラッシュでなければ、検索は現在の作業ディレクトリから始まります。</p> <p>スラッシュ 1 つだけからなるパス名はルートディレクトリを指します。</p> <p>特に指定されていない場合、NULL のパス名は、存在しないファイルを表すものと解釈されます。</p>
プロセス ID	<p>各プロセスは、存在している間 (寿命が終わるまで)、プロセス ID と呼ばれる一意の正の整数で認識されます。あるプロセス ID を割り当てた後、その ID を持つプロセス、プロセスグループ、セッションの寿命がすべて終わるまで、システムはその ID を再使用できません。プロセス内には <code>thread_t</code>、<code>LWPID_t</code> と呼ばれているスレッド ID を持つスレッドがあります。これらのスレッドは外部のプロセスからは見えません。</p>
親プロセス ID	<p>現在アクティブなプロセスが新しいプロセスを生成した場合 (<code>fork(2)</code> を参照)、現プロセスが新プロセスの親プロセスとなり、現プロセスのプロセス ID が親プロセス ID になります。</p>
特権	<p>特権を持つということは、システムによる制限を受けずに操作する権利を持つことです。</p>
プロセスグループ	<p>システム中の各プロセスは、必ずあるプロセスグループのメンバーになっています。各グループはプロセスグループ ID によって認識されます。現在プロセスグループリーダーではないプロセスは、新しいグループを生成してそのグループのリーダーになることができます。また、現在プロセスグループリーダーではないプロセスは、同じセッション中の他の既存のプロセスグループに加わることができます。新しく生成されたプロセスは、その親プロセスが属するプロセスグループのメンバーになります。</p>
プロセスグループ リーダー	<p>プロセスグループ ID は、プロセスグループリーダーの ID と同一になります。</p>
プロセスグループ ID	<p>アクティブな各プロセスは、必ずあるプロセスグループのメンバーになっています。また、プロセスグループはプロセスグループ ID と呼ばれる正の整数によって認識されます。プロセスグループ ID は、そのグループのリーダーのプロセスのプロセス ID と同一になります。プロセスをグループ化することにより、関連したプロセス群に一括してシグナルを送信できます (<code>kill(2)</code> を参照)。</p>
プロセスの寿命	<p>プロセスの寿命とは、フォークされたときから、終了したことを親プロセスが認識するまでの間を指します。 <code>wait(2)</code> を参照してください。</p>
プロセスグループ の寿命	<p>プロセスグループの寿命は、プロセスグループリーダーによってグループが生成されたときから、そのグループの最後のプロセスの寿命が終わるとき、または最後のプロセスがグループから離れるときまでを指します。</p>

Intro(2)

プロセッサセット ID	システムのプロセッサは、プロセッサセットとして知られるサブセットに分割できます。ある1つのプロセッサセットに結合されたプロセスは、そのプロセッサセット内のプロセッサ上でのみ動作し、またセット内のプロセッサは、通常、そのセットに結合されたプロセスだけを実行します。有効な各プロセッサセットは正の整数によって識別されます (pset_create(2) を参照)。
読み込み待ち行列	ストリーム内のモジュールまたはドライバ中のメッセージ待ち行列で、アップストリーム方向のメッセージを含んでいます。
実ユーザー ID と実グループ ID	<p>システム上に存在することを許可されたユーザーは、実ユーザー ID と呼ばれる正の整数 (範囲は 0 から MAXUID の値まで) によって認識されます。</p> <p>各ユーザーはグループのメンバーでもあります。各グループは、実グループ ID と呼ばれる正の整数によって認識されます。</p> <p>アクティブなプロセスの実ユーザー ID と実グループ ID は、そのプロセスを生成したユーザーの実ユーザー ID と実グループ ID に設定されています。</p>
ルートディレクトリと現在の作業ディレクトリ	パス名を検索するために、各プロセスにはルートディレクトリと現在の作業ディレクトリが対応付けられています。プロセスのルートディレクトリは、ルートファイルシステムのルートディレクトリと一致していなくてもかまいません。
保存されたリソース制限	保存されたリソース制限はプロセスの属性で、exec 関数群や setrlimit(2) で記述されるように、現在は有効でないリソース制限の処理を適用できるようにします。
保存されたユーザー ID と保存されたグループ ID	保存されたユーザー ID と保存されたグループ ID の値は、セットユーザーまたはセットグループのファイルモードビットが設定されたファイルを実行した直後の実効ユーザー ID および実効グループ ID の値です (exec(2) を参照)。
セマフォ識別子	<p>セマフォ識別子 (semid) は、semget(2) コールによって生成される一意の正の整数です。各 semid は、一群のセマフォとそれに関連付けられたデータ構造体を持ちます。このデータ構造体は semid_ds と呼ばれ、次のメンバーから構成されます。</p> <pre>struct ipc_perm sem_perm; /* operation permission struct */ struct sem *sem_base; /* ptr to first semaphore in set */ ushort_t sem_nsems; /* number of sems in set */ time_t sem_otime; /* last operation time */ time_t sem_ctime; /* last change time */ /* Times measured in secs since */ /* 00:00:00 GMT, Jan. 1, 1970 */</pre> <p>次に、semid_ds 構造体のメンバーについて説明します。</p> <p>sem_perm メンバーは、セマフォ操作用アクセス権 (後述) を指定する ipc_perm 構造体であり、以下のメンバーを含んでいます。</p> <pre>uid_t uid; /* user id */ gid_t gid; /* group id */ uid_t cuid; /* creator user id */ gid_t cgid; /* creator group id */ mode_t mode; /* r/a permission */</pre>

```
ulong_t  seq;    /* slot usage sequence number */
key_t    key;    /* key */
```

`sem_nsems` メンバーは、そのセマフォ群に含まれているセマフォの数と同一になります。各セマフォは、`sem_num` と呼ばれる負ではない整数によって参照されます。`sem_num` の値の最小値は 0 で、最大値は `sem_nsems` の値から 1 を引いた値です。

`sem_otime` メンバーは、最後に `semop(2)` 動作を実行した時刻です。

`sem_ctime` メンバーは、上記構造体のメンバーを変更する `semctl(2)` を最後に実行した時刻です。

セマフォは、次のメンバーからなる `sem` という名前のデータ構造体です。

```
ushort_t  semval; /* semaphore value */
pid_t     sempid; /* pid of last operation */
ushort_t  semncnt; /* # awaiting semval > cval */
ushort_t  semzcnt; /* # awaiting semval = 0 */
```

次に、`sem` 構造体のメンバーについて説明します。

`semval` メンバーは、セマフォの実際の値に等しい 0 以上の整数です。

`sempid` メンバーは、このセマフォの操作を最後に実行したプロセスのプロセス ID です。

`semncnt` メンバーは、このセマフォの `semval` 値が現在の値より大きくなるまで待つように、中断状態に置かれているプロセス数です。

`semzcnt` メンバーは、このセマフォの `semval` 値が 0 になるまで待つように、中断状態に置かれているプロセス数です。

セマフォ操作アクセス権

`semop(2)` と `semctl(2)` の両関数の説明では、操作に必要なアクセス権を `{token}` で表しています。この `token` は、必要なアクセス権の種類であり、次のように解釈されません。

```
00400   ユーザーによる読み取り
00200   ユーザーによる変更
00040   グループによる読み取り
00020   グループによる変更
00004   その他による読み取り
00002   その他による変更
```

次のいずれかの条件が満たされるとき、プロセスに、`semid` の読み取りおよび変更権が付与されます。

- プロセスの実効ユーザー ID がスーパーユーザーである場合

Intro(2)

	<ul style="list-style-type: none">■ プロセスの実効ユーザー ID が <code>semid</code> に対応するデータ構造体中の <code>sem_perm.cuid</code> または <code>sem_perm.uid</code> と一致していて、<code>sem_perm.mode</code> の「ユーザー」部分 (0600) に正しいビットが設定されている場合■ プロセスの実効グループ ID が <code>sem_perm.cgid</code> または <code>sem_perm.gid</code> と一致していて、<code>sem_perm.mode</code> の「グループ」部分 (060) に正しいビットが設定されている場合■ <code>sem_perm.mode</code> の「その他」部分 (06) に正しいビットが設定されている場合 <p>上記の条件がいずれもあてはまらない場合、アクセスは拒否されます。</p>
セッション	セッション ID と呼ばれる識別子によって認識される一群のプロセスで、制御端末との接続を確立する機能を持っています。現在プロセスグループリーダーではないプロセスは、新しいセッションとプロセスグループを生成することができます。このとき、生成元のプロセスは、新セッションのセッションリーダーになり、新プロセスグループのプロセスグループリーダーになります。新しく生成されたプロセスは、生成元のセッションのメンバーになります。
セッション ID	システム中の各セッションは、その寿命が終わるまで、セッション ID という一意の正の整数値によって認識されます。セッションリーダーのプロセス ID が、セッション ID になります。
セッションリーダー	プロセス ID、プロセスグループ ID は、セッションリーダーのセッション ID と同一になります。
セッションの寿命	セッションの寿命は、セッションリーダーによってセッションが開始されたときから、そのセッションの最後のプロセスの寿命が終わるとき、または最後のプロセスがセッションから離れるときまでを指します。
共有メモリー識別子	共有メモリー識別子 (<code>shmid</code>) は、 <code>shmget(2)</code> コールによって生成される一意の正の整数です。各 <code>shmid</code> は、共有メモリーセグメントと呼ばれるメモリー領域と、対応するデータ構造体を持っています (ユーザーは、共有メモリーセグメントへの参照が終わったら、必ずそのセグメントを削除しなければなりません)。データ構造体は <code>shmid_ds</code> と呼ばれ、次のメンバーで構成されます。
	<pre>struct ipc_perm shm_perm; /* operation permission struct */ size_t shm_segsz; /* size of segment */ struct anon_map *shm_amp; /* ptr to region structure */ char pad[4]; /* for swap compatibility */ pid_t shm_lpid; /* pid of last operation */ pid_t shm_cpid; /* creator pid */ shmatt_t shm_nattch; /* number of current attaches */ ulong_t shm_cnattch; /* used only for shminfo */ time_t shm_atime; /* last attach time */ time_t shm_dtime; /* last detach time */ time_t shm_ctime; /* last change time */ /* Times measured in secs since */ /* 00:00:00 GMT, Jan. 1, 1970 */</pre>
	次に、 <code>shmid_ds</code> 構造体のメンバーについて説明します。

shm_perm メンバーは、共有メモリー操作用アクセス権 (後述) を指定する ipc_perm 構造体であり、次のメンバーを含んでいます。

```
uid_t    cuid;          /* creator user id */
gid_t    cgid;          /* creator group id */
uid_t    uid;           /* user id */
gid_t    gid;           /* group id */
mode_t    mode;         /* r/w permission */
ulong    seq;           /* slot usage sequence # */
key_t    key;           /* key */
```

shm_segsz メンバーは、共有メモリーセグメントのサイズをバイト単位で表しています。

shm_cpuid メンバーは、この共有メモリー識別子を生成したプロセスのプロセス ID です。

shm_lpid メンバーは、最後に shmat () と shmdt () を実行したプロセスのプロセス ID です (shmop(2) を参照)。

shm_nattch メンバーは、このセグメントが現在接続されているプロセス数です。

shm_atime メンバーは、最後に shmat () を実行した時刻です (shmop(2) を参照)。

shm_dtime メンバーは、最後に shmdt () 動作を実行した時刻です (shmop(2) を参照)。

shm_ctime メンバーは、上記構造体のメンバーを変更する shmctl(2) を最後に実行した時刻です。

共有メモリー操作 アクセス権

shmctl(2)、shmat ()、shmdt () (shmop(2) を参照) の各関数の説明では、操作に必要なアクセス権を {token} で表しています。token は必要なアクセス権の種類であり、次のように解釈されます。

```
00400    ユーザーによる読み取り
00200    ユーザーによる書き込み
00040    グループによる読み取り
00020    グループによる書き込み
00004    その他による読み取り
00002    その他による書き込み
```

以下のいずれかの条件が満たされるとき、プロセスに、shm_id の読み取りおよび書き込み権が付与されます。

- そプロセスの実効ユーザー ID がスーパーユーザーである場合
- プロセスの実効ユーザー ID が shm_id に対応するデータ構造体中の shm_perm.cuid または shm_perm.uid と一致していて、shm_perm.mode の「ユーザー」部分 (0600) に正しいビットが設定されている場合

Intro(2)

	<ul style="list-style-type: none">■ そのプロセスの実効グループ ID が <code>shm_perm.cgid</code> または <code>shm_perm.gid</code> と一致していて、<code>shm_perm.mode</code> の「グループ」部分 (060) に正しいビットが設定されている場合■ <code>shm_perm.mode</code> の「その他」部分 (06) に正しいビットが設定されている場合 <p>上記の条件がいずれもあてはまらない場合、アクセスは拒否されます。</p>
特殊プロセス	プロセス ID の値が 0 または 1 のプロセスです (<code>kill(2)</code> を参照)。ID が 0 のプロセスは <code>proc0</code> と呼ばれるプロセススケジューラです。ID が 1 のプロセスは <code>proc1</code> と呼ばれる初期化プロセス (<code>init</code>) で、システム中の他のすべてのプロセスの祖先であり、プロセス構造体の制御に使用されます。
STREAMS	カーネルメカニズムの集合体で、ネットワークドライバおよび通信ドライバの開発を支援します。STREAMS はカーネル内部と、カーネルおよびユーザーレベルのプロセスとの間で、文字の入出力に関するインタフェース基準を定義します。STREAMS メカニズムは、各種ユーティリティルーチン、カーネル機能、データ構造体から構成されます。
ストリーム	カーネル内におけるユーザープロセスとドライバルーチン間での全二重データパス。主な内容は、1 つのストリームヘッド、1 つのドライバ、そしてその両者の間に位置するいくつか (0 個の場合もある) のモジュールです。ストリームはシェルのパイプラインと似ていますが、データフローと処理が双方向に行われる点が異なります。
ストリームヘッド	ストリームの両端のうち、ストリームとユーザープロセスとのインタフェースを提供する終端です。ストリームヘッドの主要な機能は、STREAMS に関連したシステムコールを処理することと、ストリームとユーザープロセス間でデータや情報を送信することです。
スーパーユーザー	実効ユーザー ID の値が 0 のプロセス (スーパーユーザープロセス) に対応するユーザーです。スーパーユーザープロセスには、ファイルがアクセス権で保護されていてもアクセスできるなどの特権が与えられています。
アップストリーム	ストリームの方向を表します。ドライバからストリームヘッドへの方向を、アップストリームと呼びます。
書き込み待ち行列	ストリーム内のモジュールまたはドライバ中のメッセージ待ち行列で、ダウンストリーム方向のメッセージを含んでいます。