

Oracle® Solaris 11.3 でのシステム情報、プロセス、およびパフォーマンスの管理

ORACLE®

Part No: E62606
2016 年 11 月

Part No: E62606

Copyright © 1998, 2016, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、Oracle Corporationおよびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはオラクル およびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

目次

このドキュメントの使用方法	9
1 システム情報の管理	11
システム情報の表示	11
システム情報の表示に使用されるコマンド	11
チップマルチスレッド化機能に関する情報の識別	19
システム情報の変更	21
システム情報の変更のタスクマップ	22
▼ システムの日付と時間を手動で設定する方法	22
▼ その日のメッセージを設定する方法	23
▼ システムのアイデンティティの変更方法	23
2 システムプロセスの管理	25
管理が不要なシステムプロセス	25
システムプロセスの管理	26
システムプロセスの管理のタスクマップ	26
システムプロセスを管理するコマンド	26
プロセスクラス情報の表示と管理	35
プロセスクラス情報の表示	35
プロセスクラス情報の管理のタスクマップ	37
プロセスのスケジュール優先順位の変更 (priocntl)	38
▼ プロセスの優先順位を指定する方法 (priocntl)	38
▼ タイムシェアリングプロセスのスケジューリングパラメータを変更 する方法 (priocntl)	39
▼ プロセスのクラスを変更する方法 (priocntl)	40
タイムシェアリングプロセスの優先順位の変更 (nice)	41
プロセスの優先順位の変更 (nice)	41
システムのプロセスに関するトラブルシューティング方法	42

3 システムパフォーマンスのモニタリング	45
システムパフォーマンスのモニタリングに関する情報の検索場所	45
Oracle Enterprise Manager Ops Center を使用したパフォーマンスの管理	46
システムパフォーマンスに影響を与えるシステムリソースについて	46
プロセスとシステムのパフォーマンスについて	47
システムパフォーマンスのモニタリング	48
モニタリングツール	49
システムパフォーマンス情報の表示	49
仮想メモリーの統計情報を表示する	49
仮想メモリーの統計情報の表示 (vmstat)	50
システムイベント情報の表示 (vmstat -s)	51
スワッピング統計の表示 (vmstat -S)	52
デバイス当たりの割り込み数の表示 (vmstat -i)	52
ディスクの使用状況を表示する	52
ディスク容量統計情報の表示 (df)	54
システム動作のモニタリング	56
システム動作のモニタリング (sar)	56
システム動作データの自動収集 (sar)	72
4 システムタスクのスケジュール設定	77
スケジュールされているシステムタスクの概要	77
SMF を使用した定期的なタスクまたはスケジュールされているタスクのスケジュールリング	78
crontab を使用したルーチンシステムタスクのスケジュールリング	79
at を使用した 1 つのシステムタスクのスケジュールリング	80
繰り返されるシステムタスクの例	80
SMF を使用した繰り返されるシステムタスクのスケジュールリング	81
SMF のスケジュールリング管理方法	81
SMF のメソッドおよび時間値のスケジュールリング	82
SMF マニフェストの例	82
crontab を使用した繰り返されるシステムタスクのスケジュールリング	83
crontab ファイルの内容	83
cron デーモンのスケジュールリング管理	84
crontab ファイルエントリの構文	85
crontab ファイルの作成と編集	86
crontab ファイルの表示と確認	87
crontab ファイルの削除	89

crontab コマンドの使用制御	90
at コマンドを使用した 1 つのシステムタスクのスケジューリング	93
at ジョブファイルの送信	93
at ジョブを作成する	94
at 待ち行列を表示する	95
at ジョブを確認する	95
at ジョブを表示する	95
▼ at ジョブを削除する方法	96
at コマンドの使用制御	97
5 システムコンソール、端末デバイス、および電源サービスの管理	99
システムコンソールとローカル接続された端末デバイスの管理	99
システムコンソールとローカル接続された端末デバイスを管理する SMF サービス	99
システム電源サービスの管理	102
▼ 保守モードの電源サービスから回復する方法	105
索引	107

このドキュメントの使用方法

- **概要** – システム情報、プロセス、およびモニタリングパフォーマンスの管理のタスクについて説明します。
- **対象読者** – Oracle Solaris 11 リリースを使用しているシステム管理者
- **前提知識** – UNIX システムの管理経験

製品ドキュメントライブラリ

この製品および関連製品のドキュメントとリソースは <http://www.oracle.com/pls/topic/lookup?ctx=E62101-01> で入手可能です。

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

◆◆◆ 第 1 章

システム情報の管理

この章では、基本的なシステム情報を表示および変更するために必要なタスクについて説明します。

システムリソースを柔軟な方法で割り当て、モニター、および制御できるリソース管理については、『Oracle Solaris 11.3 でのリソースの管理』の第 1 章、「リソース管理の紹介」を参照してください。

この章の内容は次のとおりです。

- 11 ページの「システム情報の表示」
- 21 ページの「システム情報の変更」

システム情報の表示

このセクションでは、一般的なシステム情報を表示できるコマンドについて説明します。

システム情報の表示に使用されるコマンド

表 1 システム情報を表示するためのコマンド

コマンド	表示されるシステム情報	マニュアルページ
date	日付と時間	date(1)
hostid	ホスト ID	hostid(1)
isainfo	動作しているシステムのネイティブアプリケーションによってサポートされるビット数。ビット数は、トークンとしてスクリプトに渡すことができます。	isainfo(1)
isalist	プロセッサタイプ	isalist(1)
prtconf	システム構成情報、インストールされたメモリー、デバイスのプロパティ、および製品名	prtconf(1M)

コマンド	表示されるシステム情報	マニュアルページ
<code>prtdiag</code>	障害の発生した現場交換可能ユニット (FRU) を含む、システムの構成および診断情報	prtdiag(1M)
<code>psrinfo</code>	プロセッサ情報	psrinfo(1M)
<code>uname</code>	オペレーティングシステム名、リリース、バージョン、ノード名、ハードウェア名、およびプロセッサタイプ	uname(1)

システムのリリース情報の表示

リリースバージョンを特定するには、`/etc/release` ファイルの内容を表示します。

```
$ cat /etc/release
```

日付と時間を表示する

システムクロックに従った現在の日付と時間を表示するには、`date` コマンドを使用します。

次の例は、`date` コマンドのサンプル出力を示します。

```
$ date
Fri Jun  1 16:07:44 MDT 2012
$
```

システムのホスト ID を表示する

ホスト ID を数値 (16 進) 形式で表示するには、`hostid` コマンドを使用します。

次の例は、`hostid` コマンドの出力を示します。

```
$ hostid
80a5d34c
```

システムのアーキテクチャータイプを表示する

`isainfo` コマンドを使用して、現在のオペレーティングシステムでサポートされているアプリケーションのネイティブ命令セットのアーキテクチャータイプと名前を表示します。

次のサンプル出力は、x86 ベースのシステムからのものです。

```
$ isainfo
amd64 i386
```

次のサンプル出力は、SPARC ベースのシステムからのものです。

```
$ isainfo
sparcv9 sparc
```

`isainfo -v` コマンドは、32 ビットおよび 64 ビットのアプリケーションサポートを表示します。たとえば、次のサンプル出力は、SPARC ベースのシステムからのものです。

```
$ isainfo -v
64-bit sparcv9 applications
    asi_blk_init
32-bit sparc applications
    asi_blk_init v8plus div32 mul32
#
```

次の例は、x86 ベースのシステムからの `isainfo -v` コマンドの出力を示します。

```
$ isainfo -v
64-bit amd64 applications
    sse4.1 ssse3 ahf cx16 sse3 sse2 sse fxsr mmx cmov amd_sysc cx8 tsc fpu
32-bit i386 applications
    sse4.1 ssse3 ahf cx16 sse3 sse2 sse fxsr mmx cmov sep cx8 tsc fpu
```

[isainfo\(1\)](#) のマニュアルページを参照してください。

詳細は、`isainfo(1)` のマニュアルページを参照してください。

システムのプロセッサタイプを表示する

`isalist` コマンドを使用して、システムのプロセッサタイプに関する情報を表示します。

次のサンプル出力は、x86 ベースのシステムからのものです。

```
$ isalist
pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

次の例は、SPARC ベースのシステムからの出力です。

```
$ isalist
sparcv9 sparcv8plus sparcv8 sparcv8-fsmuld sparcv7 sparc sparcv9+vis sparcv9+vis2 \
sparcv8plus+vis sparcv8plus+vis2
```

[isalist\(1\)](#) のマニュアルページを参照してください。

システムの製品名を表示する

システムの製品名を表示するには、`-b` オプションを付けて `prtconf` コマンドを使用します。

```
$ prtconf -b
```

詳細は、[prtconf\(1M\)](#) のマニュアルページを参照してください。

次の例は、SPARC ベースのシステムでの `prtconf -b` コマンドのサンプル出力を示しています。

```
$ prtconf -b
name: ORCL,SPARC-T4-2
banner-name: SPARC T4-2
compatible: 'sun4v'
$
```

次の例は、SPARC ベースのシステムでの `prtconf -vb` コマンドのサンプル出力を示しています。追加された `-v` オプションは、冗長出力を指定します。

```
$ prtconf -vb
name: ORCL,SPARC-T3-4
banner-name: SPARC T3-4
compatible: 'sun4v'
idprom: 01840014.4fa02d28.00000000.a02d28de.00000000.00000000.00000000.00000000
openprom model: SUNW,4.33.0.b
openprom version: 'OBP 4.33.0.b 2011/05/16 16:26'
```

システムにインストールされているメモリーを表示する

システムにインストールされているメモリー容量を表示するには、`prtconf` コマンドと `grep Memory` コマンドを使用します。次の例は、`grep Memory` コマンドが `prtconf` コマンドからの出力を選択してメモリー情報のみを表示するサンプル出力を示しています。

```
$ prtconf | grep Memory
Memory size: 523776 Megabytes
```

デバイスのデフォルトのプロパティ値とカスタマイズされたプロパティ値の表示

デバイスのデフォルトのプロパティ値とカスタマイズされたプロパティ値の両方を表示し、`prtconf` コマンドと `-u` オプションを使用できます。

```
$ prtconf -u
```

`prtconf -u` コマンドの出力には、システム上に存在するすべてのドライバのデフォルトのプロパティとカスタマイズされたプロパティが表示されます。

このオプションの詳細は、[prtconf\(1M\)](#) のマニュアルページを参照してください。

例 1 SPARC: デフォルトおよびカスタムのデバイスプロパティを表示する

この例は、`bge.conf` ファイルのデフォルトプロパティとカスタムプロパティを示しています。ベンダーから提供された構成ファイルは `/kernel` ディレクトリと `/platform` ディレクトリにあります。対応する修正されたドライバ構成ファイルは `/etc/driver/drv` にあります。

```
$ prtconf -u
System Configuration: Oracle Corporation sun4v
Memory size: 523776 Megabytes
System Peripherals (Software Nodes):

ORCL, SPARC-T3-4
  scsi_vhci, instance #0
    disk, instance #4
    disk, instance #5
    disk, instance #6
    disk, instance #8
    disk, instance #9
    disk, instance #10
    disk, instance #11
    disk, instance #12
  packages (driver not attached)
    SUNW,builtin-drivers (driver not attached)
    deblocker (driver not attached)
    disk-label (driver not attached)
    terminal-emulator (driver not attached)
    dropins (driver not attached)
    SUNW,asr (driver not attached)
    kbd-translator (driver not attached)
    obp-tftp (driver not attached)
    zfs-file-system (driver not attached)
    hsfs-file-system (driver not attached)
  chosen (driver not attached)
  openprom (driver not attached)
    client-services (driver not attached)
  options, instance #0
  aliases (driver not attached)
  memory (driver not attached)
  virtual-memory (driver not attached)
  iscsi-hba (driver not attached)
    disk, instance #0 (driver not attached)
  virtual-devices, instance #0
    flashprom (driver not attached)
    tpm, instance #0 (driver not attached)
    n2cp, instance #0
    ncp, instance #0
    random-number-generator, instance #0
    console, instance #0
    channel-devices, instance #0
      virtual-channel, instance #0
      virtual-channel, instance #1
      virtual-channel-client, instance #2
      virtual-channel-client, instance #3
      virtual-domain-service, instance #0
  cpu (driver not attached)
  cpu (driver not attached)
```

例 2 x86: デフォルトおよびカスタムのデバイスプロパティーを表示する

この例は、bge.conf ファイルのデフォルトプロパティーとカスタムプロパティーを示しています。ベンダーから提供された構成ファイルは /kernel ディレクトリと /platform ディレクトリにあります。対応する修正されたドライバ構成ファイルは /etc/driver/drv にあります。

```
$ prtconf -u
System Configuration: Oracle Corporation i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

i86pc
  scsi_vhci, instance #0
  pci, instance #0
    pci10de,5e (driver not attached)
  isa, instance #0
    asy, instance #0
    motherboard (driver not attached)
    pit_beeper, instance #0
  pci10de,cb84 (driver not attached)
  pci108e,cb84, instance #0
    device, instance #0
      keyboard, instance #0
      mouse, instance #1
  pci108e,cb84, instance #0
  pci-ide, instance #0
    ide, instance #0
      sd, instance #0
      ide (driver not attached)
  pci10de,5c, instance #0
    display, instance #0
  pci10de,cb84, instance #0
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci1022,1100, instance #0
  pci1022,1101, instance #1
  pci1022,1102, instance #2
  pci1022,1103 (driver not attached)
  pci1022,1100, instance #3
  pci1022,1101, instance #4
  pci1022,1102, instance #5
  pci1022,1103 (driver not attached)
  pci, instance #1
    pci10de,5e (driver not attached)
    pci10de,cb84 (driver not attached)
    pci10de,cb84, instance #1
    pci10de,5d (driver not attached)
    pci1022,7458, instance #1
    pci1022,7459 (driver not attached)
    pci1022,7458, instance #2
      pci8086,1011, instance #0
      pci8086,1011, instance #1
      pci1000,3060, instance #0
        sd, instance #1
        sd, instance #2
    pci1022,7459 (driver not attached)
```

```

ioapics (driver not attached)
  ioapic, instance #0 (driver not attached)
  ioapic, instance #1 (driver not attached)
fw, instance #0
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)
  sb, instance #1
used-resources (driver not attached)
iscsi, instance #0
fcoe, instance #0
pseudo, instance #0
options, instance #0
xsvc, instance #0
vga_arbiter, instance #0

```

例 3 x86: システム構成情報の表示

次の例は、x86 ベースのシステムで `prtconf` コマンドを `-v` オプションとともに使用して、システムに接続されているディスク、テープ、および DVD デバイスを識別する方法を示しています。このコマンドの出力では、デバイスが存在しないデバイスインスタンスの横に "driver not attached" メッセージが表示されます。

```

$ prtconf -v | more
System Configuration: Oracle Corporation i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

i86pc
  System properties:
    name='#size-cells' type=int items=1
      value=00000002
    name='#address-cells' type=int items=1
      value=00000003
    name='relative-addressing' type=int items=1
      value=00000001
    name='MMU_PAGEOFFSET' type=int items=1
      value=00000fff
    name='MMU_PAGESIZE' type=int items=1
      value=00001000
    name='PAGESIZE' type=int items=1
      value=00001000
    name='acpi-status' type=int items=1
      value=00000013
    name='biosdev-0x81' type=byte items=588
      value=01.38.74.0e.08.1e.db.e4.fe.00.d0.ed.fe.f8.6b.04.08.d3.db.e4.fe
  .
  .
  .

```

詳細は、[driver\(4\)](#)、[driver.conf\(4\)](#)、および [prtconf\(1M\)](#) のマニュアルページを参照してください。

管理者が提供する構成ファイルの作成方法の手順については、『[Oracle Solaris 11.3 でのデバイスの管理](#)』の第 1 章、「[Oracle Solaris でのデバイスの管理](#)」を参照してください。

システムの診断情報を表示する

prtdiag コマンドを使用して、システムの構成および診断情報を表示します。

```
$ prtdiag [-v] [-l]
```

- v 冗長モード。
- l ログ出力。システムに障害やエラーがある場合は、この情報を syslogd(1M) のみに出力します。

例 4 SPARC: システムの診断情報を表示する

次の例は、SPARC ベースのシステムでの prtdiag - v コマンドの出力を示しています。簡潔にするために、この例は省略されています。

```
$ prtdiag -v | more
System Configuration: Oracle Corporation sun4v Sun Fire T200
Memory size: 16256 Megabytes

===== Virtual CPUs =====

CPU ID Frequency Implementation      Status
-----
0      1200 MHz  SUNW,UltraSPARC-T1  on-line
1      1200 MHz  SUNW,UltraSPARC-T1  on-line
2      1200 MHz  SUNW,UltraSPARC-T1  on-line
3      1200 MHz  SUNW,UltraSPARC-T1  on-line
4      1200 MHz  SUNW,UltraSPARC-T1  on-line
5      1200 MHz  SUNW,UltraSPARC-T1  on-line
6      1200 MHz  SUNW,UltraSPARC-T1  on-line
.
.
.

===== Physical Memory Configuration =====
Segment Table:
-----
Base      Segment  Interleave  Bank  Contains
Address   Size     Factor      Size  Modules
-----
0x0       16 GB    4           2 GB  MB/CMP0/CH0/R0/D0
          2 GB    4           2 GB  MB/CMP0/CH0/R0/D1
          2 GB    4           2 GB  MB/CMP0/CH0/R1/D0
          2 GB    4           2 GB  MB/CMP0/CH0/R1/D1
          2 GB    4           2 GB  MB/CMP0/CH1/R0/D0
          2 GB    4           2 GB  MB/CMP0/CH1/R0/D1
          2 GB    4           2 GB  MB/CMP0/CH1/R1/D0
.
.
.
System PROM revisions:
-----
OBP 4.30.4.d 2011/07/06 14:29

IO ASIC revisions:
-----
Location          Path          Device
                  Revision
-----
```

```

-----
IOBD/IO-BRIDGE                               /pci@780      SUNW, sun4v-pci    0
.
.
.

```

例 5 x86: システムの診断情報を表示する

次の例は、x86 ベースのシステムでの `prtdiag - 1` コマンドの出力を示しています。

```

$ prtdiag -1
System Configuration: ... Sun Fire X4100 M2
BIOS Configuration: American Megatrends Inc. 0ABJX104 04/09/2009
BMC Configuration: IPMI 1.5 (KCS: Keyboard Controller Style)

==== Processor Sockets =====
Version                               Location Tag
-----
Dual-Core AMD Opteron(tm) Processor 2220 CPU 1
Dual-Core AMD Opteron(tm) Processor 2220 CPU 2

==== Memory Device Sockets =====
Type      Status Set Device Locator      Bank Locator
-----
unknown   empty  0   DIMM0                          NODE0
unknown   empty  0   DIMM1                          NODE0
DDR2      in use 0   DIMM2                          NODE0
DDR2      in use 0   DIMM3                          NODE0
unknown   empty  0   DIMM0                          NODE1
unknown   empty  0   DIMM1                          NODE1
DDR2      in use 0   DIMM2                          NODE1
DDR2      in use 0   DIMM3                          NODE1

==== On-Board Devices =====
LSI serial-SCSI #1
Gigabit Ethernet #1
ATI Rage XL VGA

==== Upgradeable Slots =====
ID  Status  Type           Description
---
1   available PCI Express    PCIExp SLOT0
2   available PCI Express    PCIExp SLOT1
3   available PCI-X         PCIX SLOT2
4   available PCI Express    PCIExp SLOT3
5   available PCI Express    PCIExp SLOT4
$

```

チップマルチスレッド化機能に関する情報の識別

`psrinfo` コマンドが変更され、仮想プロセッサの情報だけでなく、物理プロセッサの情報も返すようになりました。この拡張機能の追加によって、チップマルチスレッド化 (CMT) 機能を識別できるようになりました。-p オプションは、システム内の物理プ

ロセッサの合計数を報告します。-t オプションは、システムのプロセッサとそれに関連付けられたソケット、コア、および CPU ID のツリーを表示します。

`psrinfo -pv` コマンドを使用すると、システム内の全物理プロセッサのほか、各物理プロセッサに関連した仮想プロセッサも表示されます。`psrinfo` コマンドのデフォルト出力は、これまでどおりシステムの仮想プロセッサ情報を表示します。

詳細は、[psrinfo\(1M\)](#) のマニュアルページを参照してください。

システムの物理プロセッサタイプの表示

`psrinfo -p` コマンドを使用し、システム上の物理プロセッサの合計数を表示します。

```
$ psrinfo -p
1
```

-v オプションを追加して、各物理プロセッサに関連付けられている仮想プロセッサの情報も表示します。例:

```
$ psrinfo -pv
The physical processor has 8 cores and 32 virtual processors (0-31)
  The core has 4 virtual processors (0-3)
  The core has 4 virtual processors (4-7)
  The core has 4 virtual processors (8-11)
  The core has 4 virtual processors (12-15)
  The core has 4 virtual processors (16-19)
  The core has 4 virtual processors (20-23)
  The core has 4 virtual processors (24-27)
  The core has 4 virtual processors (28-31)
  UltraSPARC-T1 (chipid 0, clock 1000 MHz)
```

次の例は、x86 ベースのシステムでの `psrinfo -pv` コマンドのサンプル出力を示しています。

```
$ psrinfo -pv
The physical processor has 2 virtual processors (0 1)
  x86 (AuthenticAMD 40F13 family 15 model 65 step 3 clock 2793 MHz)
    Dual-Core AMD Opteron(tm) Processor 2220      [ Socket: F(1207) ]
The physical processor has 2 virtual processors (2 3)
  x86 (AuthenticAMD 40F13 family 15 model 65 step 3 clock 2793 MHz)
    Dual-Core AMD Opteron(tm) Processor 2220      [ Socket: F(1207) ]
```

システムの仮想プロセッサタイプを表示する

システムの仮想プロセッサタイプに関する情報を表示するには、`psrinfo -v` コマンドを使用します。

```
$ psrinfo -v
```

x86 ベースのシステムでは、`isalist` コマンドを使用し、仮想プロセッサタイプを表示します。例:

```
$ isalist
amd64 pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

例 6 SPARC: システムの仮想プロセッサタイプを表示する

この例は、SPARC ベースのシステムの仮想プロセッサタイプに関する情報を表示する方法を示しています。

```
$ psrinfo -v
Status of virtual processor 28 as of: 09/13/2010 14:07:47
on-line since 04/08/2010 21:27:56.
The sparcv9 processor operates at 1400 MHz,
and has a sparcv9 floating point processor.
Status of virtual processor 29 as of: 09/13/2010 14:07:47
on-line since 04/08/2010 21:27:56.
The sparcv9 processor operates at 1400 MHz,
and has a sparcv9 floating point processor.
```

例 7 SPARC: システムの各物理プロセッサに関連付けられている仮想プロセッサを表示する

次の例は、Oracle SPARC T4-4 サーバーで `-pv` オプションとともに実行された場合の、`psrinfo` コマンドの出力を示しています。出力には、チップ (物理プロセッサ) と、スレッド位置に関するコア情報の両方が表示されます。この情報は、スレッドがどの物理 CPU 上にあるか、およびコアレベルでどのようにマップされているかを調べるのに役立ちます。

```
$ psrinfo -pv
The physical processor has 8 cores and 64 virtual processors (0-63)
The core has 8 virtual processors (0-7)
The core has 8 virtual processors (8-15)
The core has 8 virtual processors (16-23)
The core has 8 virtual processors (24-31)
The core has 8 virtual processors (32-39)
The core has 8 virtual processors (40-47)
The core has 8 virtual processors (48-55)
The core has 8 virtual processors (56-63)
SPARC-T4 (chipid 0, clock 2998 MHz)
The physical processor has 8 cores and 64 virtual processors (64-127)
The core has 8 virtual processors (64-71)
The core has 8 virtual processors (72-79)
The core has 8 virtual processors (80-87)
The core has 8 virtual processors (88-95)
The core has 8 virtual processors (96-103)
The core has 8 virtual processors (104-111)
The core has 8 virtual processors (112-119)
The core has 8 virtual processors (120-127)
SPARC-T4 (chipid 1, clock 2998 MHz)
```

システム情報の変更

このセクションでは、一般的なシステム情報を変更するコマンドを説明します。

システム情報の変更のタスクマップ

タスク	説明	手順の参照先
システムの日付と時間を手動で設定します。	システムの日付と時間を手動で設定するには、 <code>date mmddHHMM[[cc]yy]</code> コマンド行構文を使用します。	22 ページの「システムの日付と時間を手動で設定する方法」
その日のメッセージを設定します。	システムその日のメッセージを設定するには、 <code>/etc/motd</code> ファイルを編集します。	23 ページの「その日のメッセージを設定する方法」
システムのアイデンティティを変更します。	<code>hostname</code> コマンドを使用して、システムのアイデンティティを変更します。	23 ページの「システムのアイデンティティの変更方法」

▼ システムの日付と時間を手動で設定する方法

1. 管理者になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. 新しい日付と時間を指定します。

```
$ date mmddHHMM[[cc]yy]
```

mm 月 (2 桁を使用)

dd 日 (2 桁を使用)

HH 時 (2 桁で 24 時間制を使用)

MM 分 (2 桁を使用)

cc 世紀 (2 桁を使用)

yy 年 (2 桁を使用)

詳細は、[date\(1\)](#) のマニュアルページを参照してください。

3. オプションを付けずに `date` コマンドを使用し、システム日付が正しく設定されたことを確認します。

例 8 システムの日付と時間を手動で設定する

次の例は、`date` コマンドを使用して、システムの日付と時間を手動で設定する方法を示しています。

```
# date
Monday, September 13. 2010 02:00:16 PM MDT
# date 0921173404
Thu Sep 17:34:34 MST 2010
```

▼ その日のメッセージを設定する方法

その日のメッセージファイル `/etc/motd` を編集して、システム的全ユーザーに対して、ログイン時に通知または問い合わせる内容を書き込むことができます。ただし、この機能を使用するときは、必要なメッセージだけを送るようにします。メッセージファイルは定期的に編集して、不用になったメッセージを削除することをお勧めします。

1. **Administrator Message Edit** プロファイルが割り当てられている役割になります。
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. `pfedit` コマンドを使用して `/etc/motd` ファイルを編集し、必要なメッセージを追加します。

```
$ pfedit /etc/motd
```

テキストを編集して、ユーザーのログイン時に表示されるメッセージを記述します。スペース、タブ、リターンも含めます。

3. `/etc/motd` ファイルの内容を表示して、変更内容を確認します。

```
$ cat /etc/motd
Welcome to the UNIX universe. Have a nice day.
```

▼ システムのアイデンティティーの変更方法

1. 管理者になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. システムのホスト名を設定します。

```
# hostname name
```

`hostname` および `domainname` コマンドでは、ホスト名とドメイン名を永続的に設定できます。これらのコマンドを使用すると、対応する SMF プロパティおよび関連する SMF サービスも自動的に更新されます。

詳細は、[hostname\(1\)](#)、[domainname\(1M\)](#)、および [nodename\(4\)](#) のマニュアルページを参照してください。

◆◆◆ 第 2 章

システムプロセスの管理

この章では、システムプロセスを管理する手順について説明します。
この章で扱う内容は、次のとおりです。

- 25 ページの「管理が不要なシステムプロセス」
- 26 ページの「システムプロセスの管理」
- 35 ページの「プロセスクラス情報の表示と管理」
- 42 ページの「システムのプロセスに関するトラブルシューティング方法」

管理が不要なシステムプロセス

Oracle Solaris 10 および Oracle Solaris 11 リリースには、特定のタスクを実行するシステムプロセスが含まれますが、管理の必要はありません。

プロセス	説明
fsflush	ページをディスクにフラッシュするシステムデーモン
init	その他のプロセスおよび SMF コンポーネントを起動および再起動する初期システムプロセス
intrd	割り込みによるシステム負荷をモニターおよび分散するシステムプロセス
kmem_task	メモリーキャッシュのサイズをモニターするシステムプロセス
pageout	ディスクへのメモリーページングを制御するシステムプロセス
sched	OS スケジューリングとプロセススワップを担当するシステムプロセス
vm_tasks	パフォーマンスを改善するために複数の CPU にわたる仮想メモリー関連のワークロードの負荷分散を行う、プロセッサごとに 1 つのスレッドを持つシステムプロセス。
zpool-pool-name	関連するプールに対応する I/O タスクスレッドを含む ZFS ストレージプールごとのシステムプロセス

システムプロセスの管理

このセクションでは、システムプロセスを管理するためのさまざまなタスクについて説明します。

システムプロセスの管理のタスクマップ

タスク	説明	手順の参照先
プロセスを表示する。	ps コマンドを使用して、システム上のすべてのプロセスを表示する	30 ページの「プロセスを表示する方法」
プロセスについての情報を表示する	pgrep コマンドを使用して、詳細情報を表示したいプロセスのプロセス ID を取得する	31 ページの「プロセスに関する情報を表示する方法」
プロセスを制御する。	pgrep コマンドを使用して、プロセスを見つける。その後、適切な pcommand (/proc) を使用し、プロセスを制御する。(/proc) コマンドについては、表3を参照してください。	32 ページの「プロセスを制御する方法」
プロセスを強制終了する。	プロセス名かプロセス ID を使用し、プロセスを見つける。pkill コマンドまたは kill コマンドを使用し、プロセスを終了する	32 ページの「プロセスを終了させる方法 (pkill)」 33 ページの「プロセスを終了させる方法 (kill)」

システムプロセスを管理するコマンド

次の表では、システムプロセスを管理するためのコマンドについて説明します。

表 2 プロセスを管理するためのコマンド

コマンド	説明	マニュアルページ
ps, pgrep, prstat, pkill	システム上のアクティブなプロセスのステータスをチェックし、プロセスについての詳細な情報も表示します。	ps(1)、pgrep(1)、および prstat(1M)
pkill	pgrep と同様に機能するが、名前または他の属性によってプロセスを検索またはシグナルを送信して、プロセスを終了します。一致したプロセスのプロセス ID を出力するのではなく、kill コマンドと同様に、各プロセスにシグナルを送信します。	pgrep(1)、および pkill(1) kill(1)

コマンド	説明	マニュアルページ
pargs、preap	プロセスのデバッグを支援します。	pargs(1)、および preap(1)
dispadmin	デフォルトのプロセススケジューリングポリシーを表示します。	dispadmin(1M)
priocntl	プロセスに優先順位クラスを割り当てて、プロセスの優先度を管理します。	priocntl(1)
nice	タイムシェアリングプロセスの優先度を変更します。	nice(1)
psrset	特定のプロセスグループを、1つのプロセッサではなく、プロセッサのグループに結合します。	psrset(1M)

ps コマンドの使用

ps コマンドを使用すると、システム上で活動中のプロセスのステータスをチェックでき、プロセスについての技術的な情報も表示できます。このデータは、プロセスの優先順位の設定方法を決定するときなど、各種の管理タスクに利用できます。

使用するオプションに応じて、ps コマンドは次の情報を表示します。

- プロセスの現在のステータス
- プロセス ID
- 親プロセス ID
- ユーザー ID
- スケジューリングクラス
- 優先順位
- プロセスのアドレス
- 使用したメモリー
- 使用した CPU 時間

次の表では、ps コマンドによって報告されるいくつかのフィールドについて説明します。表示されるフィールドは、選択するオプションによって異なります。使用可能なすべてのオプションについては、ps(1)のマニュアルページを参照してください。

UID	プロセス所有者の実効ユーザー ID。
PID	プロセス ID。
PPID	親プロセス ID
C	スケジューリングのためのプロセッサ使用率。このフィールドは -c オプションを使用すると表示されない

CLS	プロセスが所属するスケジューリングクラス。リアルタイム、システム、またはタイムシェアリングのいずれか。このフィールドは、 <code>-c</code> オプションを指定した場合にのみ表示される
PRI	カーネルスレッドのスケジューリング優先順位。番号が大きいほど優先順位が高い
NI	プロセスの nice 値。これは、スケジュール優先順位に影響する。プロセスの nice 値を大きくすると、その優先順位が下がる
ADDR	proc 構造体のアドレス
SZ	プロセスの仮想アドレスサイズ
WCHAN	プロセスが休眠中のイベントまたはロックのアドレス
STIME	プロセスの起動時間 (時、分、秒)
TTY	プロセス (またはその親プロセス) が起動された端末。疑問符は、制御端末がないことを示す
TIME	プロセスの起動以降に使用した合計 CPU 時間
CMD	プロセスを生成したコマンド

/proc ファイルシステムとコマンド

プロセスコマンドを使用すると、/proc ディレクトリにあるプロセスに関する詳細情報を表示できます。次の表に、/proc プロセスコマンドを示します。/proc ディレクトリは プロセスファイルシステム (PROCFS) とも呼ばれます。アクティブなプロセスのイメージは、そのプロセス ID 番号を使って PROCFS に格納されます。

表 3 プロセスコマンド (/proc)

プロセスコマンド	説明
psched	プロセスの資格情報を表示する
pfiles	プロセス内で開いているファイルに関する fstat 情報と fcntl 情報を表示する
pflags	/proc 追跡フラグ、保留状態のシグナルと保持状態のシグナル、その他のステータス情報を表示する
pldd	プロセスにリンクされている動的ライブラリを表示する
pmap	各プロセスのアドレス空間マップを表示する
psig	各プロセスのシグナルの動作とハンドラを表示する
prun	各プロセスを開始する

プロセスコマンド	説明
pstack	各プロセス内の軽量プロセスの 16 進とシンボルのスタックトレースを表示する
pstop	各プロセスを停止する
ptime	microstate アカウントを使用してプロセスの時間を測定する
ptree	プロセスを含むプロセスツリーを表示する
pwait	プロセス終了後のステータス情報を表示する
pwdx	プロセスの現在の作業ディレクトリを表示する

詳細は、[proc\(1\)](#) のマニュアルページを参照してください。

プロセスツールは `ps` コマンドの一部のオプションに似ていますが、このツールから提供される出力の方が詳細です。

プロセスコマンドには次の機能があります。

- `fstat` や `fcnt1`、作業ディレクトリ、親プロセスと子プロセスのツリーなど、プロセスに関する詳細情報を表示します。
- ユーザーがプロセスを停止または再開できるように、プロセスに対する制御を提供します。

プロセスコマンド (/proc) を使用したプロセスの管理

一部のプロセスコマンドを使用して、プロセスに関する詳細な技術情報を表示したり、アクティブなプロセスを制御したりできます。[表3](#)に、一部の `/proc` コマンドを示しています。

プロセスが無限ループに陥った場合や、実行時間が長すぎる場合は、プロセスを終了(強制終了)できます。`kill` または `pkill` コマンドを使用したプロセスの停止の詳細は、[第2章「システムプロセスの管理」](#)を参照してください。

`/proc` ファイルシステムは、状態情報と制御機能のためのサブディレクトリを含むディレクトリ階層です。

`/proc` ファイルシステムは、`xwatchpoint` 機能も提供し、この機能は、プロセスのアドレス空間の個々のページの読み取り権または書き込み権を再マップするために使用されます。この機能は制限がなく、MT-safe です。

デバッグ用ツールは、`xwatchpoint` 機能を使用するように変更されており、`xwatchpoint` プロセス全体がより高速になっています。

`dbx` デバッグ用ツールを使用して `xwatchpoints` を設定する際の、次の制限はなくなりました。

- SPARC ベースシステムのレジスタウィンドウのため、スタック上のローカル変数に `xwatchpoints` を設定する。

- マルチスレッド化されたプロセスに `xwatchpoints` を設定する。

詳細は、[proc\(4\)](#) および [mdb\(1\)](#) のマニュアルページを参照してください。

▼ プロセスを表示する方法

- `ps` コマンドを使用して、システム上のすべてのプロセスを表示します。

```
$ ps [-efc]
```

`ps` ログインセッションに関連するプロセスのみを表示する

`-ef` システム上で実行中のすべてのプロセスに関する詳細情報を表示する

`-c` プロセススケジューラ情報を表示する

例 9 プロセスを表示する

次の例は、オプションを指定しないときの `ps` コマンドからの出力を示します。

```
$ ps
  PID TTY          TIME CMD
 1664 pts/4        0:06 csh
 2081 pts/4        0:00 ps
```

次の例は、`ps -ef` コマンドからの出力を示しています。この出力例は、システムのブート時に最初に実行されたプロセスが `sched` (スワップ) であり、それに続いて `init` プロセス、`pageout` の順に実行されたことを示しています。

```
$ ps -ef
  UID  PID  PPID  C   STIME TTY          TIME CMD
  root   0    0    0  18:04:04 ?        0:15 sched
  root   5    0    0  18:04:03 ?        0:05 zpool-rpool
  root   1    0    0  18:04:05 ?        0:00 /sbin/init
  root   2    0    0  18:04:05 ?        0:00 pageout
  root   3    0    0  18:04:05 ?        2:52 fsflush
  root   6    0    0  18:04:05 ?        0:02 vmtasks
 daemon 739  1    0  19:03:58 ?        0:00 /usr/lib/nfs/nfs4cbd
  root   9    1    0  18:04:06 ?        0:14 /lib/svc/bin/svc.startd
  root  11    1    0  18:04:06 ?        0:45 /lib/svc/bin/svc.configd
 daemon 559  1    0  18:04:49 ?        0:00 /usr/sbin/rpcbind
 netcfg 47    1    0  18:04:19 ?        0:01 /lib/inet/netcfgd
 dladm  44    1    0  18:04:17 ?        0:00 /sbin/dlmgmt
 netadm 51    1    0  18:04:22 ?        0:01 /lib/inet/ipmgmt
  root  372  338  0  18:04:43 ?        0:00 /usr/lib/hal/hald-addon-cpufreq
  root   67    1    0  18:04:30 ?        0:02 /lib/inet/in.mpathd
  root  141    1    0  18:04:38 ?        0:00 /usr/lib/pfexecd
 netadm 89    1    0  18:04:31 ?        0:03 /lib/inet/nwamd
  root  602    1    0  18:04:50 ?        0:02 /usr/lib/inet/inetd start
  root  131    1    0  18:04:35 ?        0:01 /sbin/dhccpagent
 daemon 119  1    0  18:04:33 ?        0:00 /lib/crypto/kcfd
  root  333    1    0  18:04:41 ?        0:07 /usr/lib/hal/hald --daemon=yes
  root  370  338  0  18:04:43 ?        0:00 /usr/lib/hal/hald-addon-network-
discovery
```

```

root 159 1 0 18:04:39 ? 0:00 /usr/lib/sysevent/syseventd
root 236 1 0 18:04:40 ? 0:00 /usr/lib/ldoms/drd
root 535 1 0 18:04:46 ? 0:09 /usr/sbin/nscd
root 305 1 0 18:04:40 ? 0:00 /usr/lib/zones/zonestatd
root 326 1 0 18:04:41 ? 0:03 /usr/lib/devfsadm/devfsadm
root 314 1 0 18:04:40 ? 0:00 /usr/lib/dbus-daemon --system

```

▼ プロセスに関する情報を表示する方法

1. 詳細を表示したいプロセスのプロセス ID を表示します。

```
# pgrep process
```

プロセス ID は、出力の第 1 列目に表示されます。

2. プロセス情報を表示します。

```
# /usr/bin/pcommand PID
```

pcommand 実行するプロセスコマンド。表3 に、これらのコマンドの一覧と説明を示しています。

PID プロセス ID

例 10 プロセスに関する情報を表示する

次の例は、プロセスコマンドを使用して cron プロセスに関する詳細情報を表示する方法を示しています。

```

# pgrep cron      cron プロセスのプロセス ID を表示する。
4780
# pwdx 4780      cron プロセスの現在の作業ディレクトリを表示する
4780: /var/spool/cron/atjobs
# ptree 4780     cron プロセスを含むプロセスツリーを表示する
4780 /usr/sbin/cron
# pfiles 4780    fstat と fcntl の情報を表示する
4780: /usr/sbin/cron
Current rlimit: 256 file descriptors
0: S_IFCHR mode:0666 dev:290,0 ino:6815752 uid:0 gid:3 rdev:13,2
  O_RDONLY|O_LARGEFILE
  /devices/pseudo/mm@0:null
1: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
  O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
  /var/cron/log
2: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
  O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
  /var/cron/log
3: S_IFIFO mode:0600 dev:32,128 ino:42049 uid:0 gid:0 size:0
  O_RDWR|O_LARGEFILE
  /etc/cron.d/FIFO
4: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
  O_RDWR|O_NONBLOCK
5: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0

```

```
O_RDWR
```

▼ プロセスを制御する方法

1. 制御するプロセスのプロセス ID を表示します。

```
# pgrep process
```

プロセス ID は、出力の第 1 列目に表示されます。

2. 適切なプロセスコマンドを使用してプロセスを制御します。

```
# /usr/bin/pcommand PID
```

pcommand 実行するプロセスコマンド。表3 に、これらのコマンドの一覧と説明を示しています。

PID プロセス ID

3. プロセスステータスを確認します。

```
# ps -ef | grep PID
```

プロセスの終了 (pkill、kill)

無限ループのプロセスを停止 (強制終了) したり、大規模なジョブの完了前に停止したりする必要が生じる場合があります。所有しているプロセスであれば、どれでも終了できます。システム管理者は、プロセス ID が 0、1、2、3、および 4 のプロセス以外のすべてのプロセスを強制終了できます。プロセス ID が 0、1、2、3、4 のプロセスを終了させると、システムがクラッシュする可能性があります。

詳細は、[pgrep\(1\)](#)、[pkill\(1\)](#)、および [kill\(1\)](#) のマニュアルページを参照してください。

▼ プロセスを終了させる方法 (pkill)

1. ほかのユーザーのプロセスを終了するには、root 役割になります。
2. 終了したいプロセスのプロセス ID を表示します。

```
$ pgrep process
```

例:

```
$ pgrep netscape
587
566
```

プロセス ID が出力に表示されます。

注記 - Sun Ray™ システムのプロセス情報を取得するには、次のコマンドを使用します。

すべてのユーザープロセスを一覧表示するには:

```
# ps -fu user
```

ユーザーによって所有される特定のプロセスを特定するには:

```
# ps -fu user | grep process
```

3. プロセスを終了します。

```
$ pkill [signal] PID
```

signal pkill コマンド行構文にシグナルが何も含まれない場合は、使用されるデフォルトシグナルは -15 (SIGKILL)。-9 シグナル (SIGTERM) を pkill コマンドで使用すると、プロセスをただちに終了できる。ただし -9 シグナルは、データが失われる可能性があるため、データベースプロセスや LDAP サーバープロセスなどのプロセスを強制終了するために使用してはならない。

PID 停止するプロセスの名前。

ヒント - pkill コマンドを使用してプロセスを終了する場合は、まず、シグナルオプションを使用せずにコマンドだけで試行してみます。プロセスが数分後に終了しない場合は、-9 シグナルを付けて pkill コマンドを使用します。

4. プロセスが終了したことを確認します。

```
$ pgrep process
```

終了したプロセスは、pgrep コマンドの出力に表示されないはずです。

▼ プロセスを終了させる方法 (kill)

1. ほかのユーザーのプロセスを終了するには、root 役割になります。
2. 終了したいプロセスのプロセス ID を表示します。

```
# ps -fu user
```

ここで、*user* はプロセスの所有者です。

プロセス ID は、出力の第 1 列目に表示されます。

3. プロセスを終了します。

```
# kill [signal-number] PID
```

signal pkill コマンド行構文にシグナルが何も含まれない場合は、使用されるデフォルトシグナルは -15 (SIGKILL)。-9 シグナル (SIGTERM) を pkill コマンドで使用すると、プロセスをただちに終了できる。ただし -9 シグナルは、データが失われる可能性があるため、データベースプロセスや LDAP サーバープロセスなどのプロセスを強制終了するために使用してはならない。

PID 終了したいプロセスのプロセス ID

ヒント - kill コマンドを使用してプロセスを終了する場合は、まず、シグナルオプションを使用せずにコマンドだけで試行してみます。数分待ってプロセスが終了しなければ、kill コマンドに -9 シグナルを付けて使用します。

4. プロセスが終了したことを確認します。

```
$ ps
```

終了したプロセスは、ps コマンドの出力に表示されないはずです。

プロセスのデバッグ (pargs、preap)

pargs コマンドと preap コマンドは、プロセスのデバッグを改善します。pargs コマンドを使用すると、動作中のプロセスまたはコアファイルに関連付けられた引数と環境変数を表示できます。preap コマンドを使用すると、終了した (ゾンビ) プロセスを削除できます。ゾンビプロセスとは、その終了ステータスがまだ親に回収されていないプロセスをいいます。これらのプロセスは概して無害ですが、数が多ければシステムリソースを消費します。pargs コマンドと preap コマンドを使用して、ユーザーの検査権限に含まれるすべてのプロセスを検査できます。管理者になると、すべてのプロセスを検査できます。

preap コマンドの使用については、[preap\(1\)](#) のマニュアルページを参照してください。pargs コマンドの使用については、[pargs\(1\)](#) のマニュアルページを参照してください。[proc\(1\)](#) のマニュアルページも参照してください。

例 11 プロセスをデバッグする (pargs)

pargs コマンドによって、プロセスに渡された引数を ps コマンドで一部しか表示できないという問題がようやく解決されました。次の例では、pargs コマンドを pgrep コマンドと併用して、プロセスに渡された引数を表示する方法を示します。

```
# pargs `pgrep ttymon`
579: /usr/lib/saf/ttymon -g -h -p system-name console login:
-T sun -d /dev/console -l
```

```
argv[0]: /usr/lib/saf/ttymon
argv[1]: -g
argv[2]: -h
argv[3]: -p
argv[4]: system-name console login:
argv[5]: -T
argv[6]: sun
argv[7]: -d
argv[8]: /dev/console
argv[9]: -l
argv[10]: console
argv[11]: -m
argv[12]: ldterm,ttcompat
548: /usr/lib/saf/ttymon
argv[0]: /usr/lib/saf/ttymon
```

次の例では、`pargs -e` コマンドを使用して、プロセスに関連付けられた環境変数を表示する方法を示します。

```
$ pargs -e 6763
6763: tcsh
envp[0]: DISPLAY=:0.0
```

プロセスクラス情報の表示と管理

プロセススケジューリングクラスをシステムに構成し、タイムシェアリングクラスのユーザー優先順位の範囲を構成できます。

プロセススケジューリングクラスの種類は次のとおりです。

- 公平配分 (FSS)
- 固定 (FX)
- システム (SYS)
- 対話型 (IA)
- リアルタイム (RT)
- タイムシェアリング (TS)
 - ユーザーが指定する -60 から +60 までの優先順位の範囲。
 - プロセスの優先順位は、親プロセスから継承されます。この優先順位をユーザーモードの優先順位と呼びます。
 - システムは、ユーザーモードの優先順位をタイムシェアリングディスパッチパラメータテーブル内で検索します。次にシステムは、`nice` または `pricntl` (ユーザー指定) 優先順位に追加し、0-59 の範囲を確保してグローバル優先順位を作成します。

プロセスクラス情報の表示

このセクションの内容は次のとおりです。

36 ページの「プロセスの優先順位情報を表示する」

`prionctl -l` コマンドを使用し、プロセスのスケジューリングクラスと優先順位の範囲を表示する。

36 ページの「プロセスのグローバル優先順位を表示する」

`ps -ec1` コマンドを使用し、プロセスのグローバル優先順位を表示する

プロセスの優先順位情報を表示する

`prionctl -l` コマンドを使用して、スケジューリングクラスおよび優先順位の範囲を表示します。

```
$ prionctl -l
```

次の例は、`prionctl -l` コマンドからの出力を示しています。

```
# prionctl -l
CONFIGURED CLASSES
=====

SYS (System Class)

TS (Time Sharing)
    Configured TS User Priority Range: -60 through 60

FX (Fixed priority)
    Configured FX User Priority Range: 0 through 60

IA (Interactive)
    Configured IA User Priority Range: -60 through 60
```

プロセスのグローバル優先順位を表示する

`ps` コマンドを使用して、プロセスのグローバル優先順位を表示します。

```
$ ps -ec1
```

グローバル優先順位は、PRI カラムの下に表示されます。

次の例は、`ps -ec1` コマンドの出力を示しています。PRI 列の値は、各プロセスの優先順位を示しています。

```
$ ps -ec1
 F S   UID   PID  PPID  CLS  PRI   ADDR   SZ   WCHAN  TTY          TIME CMD
 1 T    0     0    0   SYS  96    ?     0     ?     ?     0:11 sched
 1 S    0     5    0   SDC  99    ?     0     ? ?    ?     0:01 zpool-rp
 0 S    0     1    0   TS   59    ?    688    ? ?    ?     0:00 init
 1 S    0     2    0   SYS  98    ?     0     ? ?    ?     0:00 pageout
 1 S    0     3    0   SYS  60    ?     0     ? ?    ?     2:31 fsflush
```

1 S	0	6	0	SDC	99	?	0	??	0:00	vmtasks
0 S	16	56	1	TS	59	?	1026	??	0:01	ipmgmt
0 S	0	9	1	TS	59	?	3480	??	0:04	svc.star
0 S	0	11	1	TS	59	?	3480	??	0:13	svc.conf
0 S	0	162	1	TS	59	?	533	??	0:00	pfexecd
0 S	0	1738	1730	TS	59	?	817	? pts/ 1	0:00	bash
0 S	1	852	1	TS	59	?	851	??	0:17	rpcbind
0 S	17	43	1	TS	59	?	1096	??	0:01	netcfgd
0 S	15	47	1	TS	59	?	765	??	0:00	d1mgmt
0 S	0	68	1	TS	59	?	694	??	0:01	in.mpath
0 S	1	1220	1	FX	60	?	682	??	0:00	nfs4cbd
0 S	16	89	1	TS	59	?	1673	??	0:02	nwamd
0 S	0	146	1	TS	59	?	629	??	0:01	dhcpgen
0 S	1	129	1	TS	59	?	1843	??	0:00	kcfd
0 S	1	1215	1	FX	60	?	738	??	0:00	lockd
0 S	0	829	828	TS	59	?	968	??	0:00	hald-run
0 S	0	361	1	TS	59	?	1081	??	0:01	devfsadm
0 S	0	879	1	TS	59	?	1166	??	0:01	inetd
0 0	119764	1773	880	TS	59	?	557	cons ole	0:00	ps
0 S	0	844	829	TS	59	?	996	??	0:00	hald-add
0 S	0	895	866	TS	59	?	590	??	0:00	tymon
0 S	0	840	1	TS	59	?	495	??	0:00	cron
0 S	0	874	1	TS	59	?	425	??	0:00	utmpd
0 S	0	1724	956	TS	59	?	2215	??	0:00	sshd
0 S	119764	880	9	TS	59	?	565	? cons ole	0:00	csch
0 S	0	210	1	TS	59	?	1622	??	0:00	sysevent
0 S	0	279	1	TS	59	?	472	??	0:00	iscsid
0 S	1	1221	1	TS	59	?	1349	??	0:00	nfsmapid
1 S	0	374	0	SDC	99	?	0	??	0:00	zpool-us
0 S	0	1207	1	TS	59	?	1063	??	0:00	rmvolmgr
0 S	0	828	1	TS	59	?	1776	??	0:03	hald
0 S	0	853	829	TS	59	?	896	??	0:02	hald-add
0 S	0	373	1	TS	59	?	985	??	0:00	picld
0 S	0	299	1	TS	59	?	836	??	0:00	dbus-dae
0 S	12524	1730	1725	TS	59	?	452	? pts/ 1	0:00	csch
0 S	0	370	1	TS	59	?	574	??	0:00	powerd
0 S	0	264	1	FX	60	?	637	??	0:00	zonestat
0 S	0	866	9	TS	59	?	555	??	0:00	sac
0 S	0	851	829	TS	59	?	998	??	0:00	hald-add
0 S	12524	1725	1724	TS	59	?	2732	??	0:00	sshd
0 S	1	1211	1	TS	59	?	783	??	0:00	statd
0 S	0	1046	1	TS	59	?	1770	??	0:13	intrd
0 S	0	889	1	TS	59	?	1063	??	0:00	syslogd
0 S	0	1209	1	TS	59	?	792	??	0:00	in.ndpd
0 S	0	1188	1186	TS	59	?	951	??	0:15	automoun
0 S	0	1172	829	TS	59	?	725	??	0:00	hald-add
0 S	0	1186	1	TS	59	?	692	??	0:00	automoun
0 S	101	1739	1738	TS	59	?	817	? pts/ 1	0:00	bash
0 S	0	1199	1	TS	59	?	1495	??	0:02	sendmail
0 S	0	956	1	TS	59	?	1729	??	0:00	sshd
0 S	25	1192	1	TS	59	?	1528	??	0:00	sendmail
0 S	0	934	1	TS	59	?	6897	??	0:14	fmd
0 S	0	1131	1	TS	59	?	1691	??	0:07	nscd
0 S	1	1181	1	TS	59	?	699	??	0:00	ypbind

プロセスクラス情報の管理のタスクマップ

次の手順を使用して、プロセスクラスを管理します。

タスク	説明	手順の参照先
プロセスに優先順位を割り当てる	<code>priocntl -e -c</code> コマンドを使用し、割り当てた優先順位でプロセスを開始する。	38 ページの「プロセスの優先順位を指定する方法 (priocntl)」
タイムシェアリングプロセスのスケジューリングパラメータを変更する	<code>priocntl -s -m</code> コマンドを使用し、タイムシェアリングプロセスのスケジューリングパラメータを変更する	39 ページの「タイムシェアリングプロセスのスケジューリングパラメータを変更する方法 (priocntl)」
プロセスのクラスを変更する	<code>priocntl -s -c</code> コマンドを使用し、プロセスのクラスを変更する	40 ページの「プロセスのクラスを変更する方法 (priocntl)」
プロセスの優先順位を変更する	<code>/usr/bin/nice</code> コマンドを適切なオプションとともに使用し、プロセスの優先順位を低くしたり高くしたりする	41 ページの「プロセスの優先順位の変更 (nice)」

プロセスのスケジューリング優先順位の変更 (priocntl)

プロセスのスケジューリング優先順位とは、スケジューリングポリシーに従ってプロセススケジューラによって割り当てられる優先順位のことです。 `dispadmin` コマンドを使用すると、デフォルトのスケジューリングポリシーを表示できます。詳細は、 [dispadmin\(1M\)](#) のマニュアルページを参照してください。

次の手順に示すように、 `priocntl` コマンドを使用すると、プロセスを優先順位クラスに割り当てたり、プロセスの優先順位を管理したりできます。

▼ プロセスの優先順位を指定する方法 (priocntl)

1. **root** 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. 指定した優先順位でプロセスを起動します。

```
# priocntl -e -c class -m user-limit -p PRI command-name
```

<code>-e</code>	コマンドを実行する。
<code>-c class</code>	プロセスを実行する範囲のクラスを指定する。有効なクラスは TS (タイムシェアリング)、RT (リアルタイム)、IA (対話型)、FSS (公平配分)、および FX (固定優先順位)
<code>-m user-limit</code>	このオプションと共に <code>-p</code> オプションを使用すると、優先順位を上下できる最大範囲も指定できる。

`-p PRI` リアルタイムスレッド用に RT クラス内で相対優先順位を指定できるようにする。タイムシェアリングプロセスの場合は、`-p` オプションを使用すると -60 から +60 までのユーザー指定の優先順位を指定できる。

`command-name` 実行されるコマンドの名前を指定する。

3. プロセスステータスを確認します。

```
# ps -ec1 | grep command-name
```

例 12 プロセスの優先順位を指定する (priocntl)

次の例では、ユーザーが指定できる最上位の優先順位を使用して `find` コマンドを開始する方法を示します。

```
# priocntl -e -c TS -m 60 -p 60 find . -name core -print
# ps -ec1 | grep find
```

▼ タイムシェアリングプロセスのスケジューリングパラメータを変更する方法 (priocntl)

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. 実行中のタイムシェアリングプロセスのスケジューリングパラメータを変更します。

```
# priocntl -s -m user-limit [-p user-priority] -i ID type ID list
```

`-s` ユーザー優先順位の範囲について上限を設定し、現在の優先順位を変更する

`-m user-limit` `-p` オプションを使用するときに、優先順位を上下できる最大範囲を指定する

`-p user-priority` 優先順位を指定する

`-i ID type ID list` `ID type` と `ID list` の組み合わせを使用してプロセスを識別する。`ID` タイプではプロセス ID やユーザー ID など、ID のタイプを指定する。`ID` リストではプロセス ID またはユーザー ID のリストを識別する。

3. プロセスステータスを確認します。

```
# ps -ecl | grep ID list
```

例 13 タイムシェアリングプロセスのスケジューリングパラメータを変更する (priocntl)

次の例では、500 ミリ秒のタイムスライス、クラス RT 内の優先順位 20、グローバル優先順位 120 を指定して、コマンドを実行します。

```
# priocntl -e -c RT -m 500 -p 20 myprog
# ps -ecl | grep myprog
```

▼ プロセスのクラスを変更する方法 (priocntl)

1. (オプション) root 役割になります。

注記 - プロセスをリアルタイムプロセスに変更したり、リアルタイムプロセスから変更したりするには、ユーザーはroot役割になるか、リアルタイムシェル内で作業中である必要があります。root 役割でユーザープロセスをリアルタイムクラスに変更すると、そのユーザーは `priocntl -s` コマンドを使用して、リアルタイムのスケジューリングパラメータを変更できなくなります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. プロセスのクラスを変更する

```
# priocntl -s -c class -i ID type ID list
```

<code>-s</code>	ユーザー優先順位の範囲について上限を設定し、現在の優先順位を変更する
<code>-c class</code>	クラス TS (タイムシェアリング) または RT (リアルタイム) を指定して、プロセスのクラスを変更する
<code>-i ID type ID list</code>	<code>ID type</code> と <code>ID list</code> の組み合わせを使用してプロセスを識別する。 <code>ID</code> タイプではプロセス ID やユーザー ID など、ID のタイプを指定する。 <code>ID</code> リストではプロセス ID またはユーザー ID のリストを識別する。

3. プロセスステータスを確認します。

```
# ps -ecl | grep ID list
```

例 14 プロセスのクラスを変更する (priosctl)

次の例では、ユーザー 15249 が所有するすべてのプロセスをリアルタイムプロセスに変更します。

```
# priosctl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

タイムシェアリングプロセスの優先順位の変更 (nice)

nice コマンドは、以前のリリースとの下位互換性を保つためののみサポートされません。priosctl コマンドを使用する方がプロセスを柔軟に管理できます。

プロセスの優先順位は、そのスケジューリングクラスポリシーと nice 値によって決定されます。各タイムシェアリングプロセスは、グローバル優先順位を持っています。グローバル優先順位は、ユーザーが指定した優先順位 (nice コマンドまたは priosctl コマンドの影響を受ける) とシステムで計算された優先順位を加算して算出されます。

プロセスの実行優先順位番号は、オペレーティングシステムによって割り当てられます。優先順位番号は、プロセスのスケジューリングクラス、使用される CPU 時間、nice 値 (タイムシェアリングプロセスの場合) などの、複数の要素によって決定されます。

各タイムシェアリングプロセスは、親プロセスから継承したデフォルトの nice 値で起動します。nice 値は、ps レポートの NI カラムに表示されます。

ユーザーは、自分が与える nice 値優先順位を大きくしてプロセスの優先順位を下げるすることができます。ただし、nice 値を小さくしてプロセスの優先順位を上げることができるのは、管理者だけです。これは、ユーザーが各自のプロセスの優先順位を大きくして CPU の独占比率を高めるのを防ぐためです。

nice 値の範囲は 0 - +39 (0 が最高の優先順位) です。各タイムシェアリングプロセスのデフォルトの nice 値は 20 です。このコマンドには、利用できるバージョンが 2 つ (標準バージョンの /usr/bin/nice と、C シェルの組み込みコマンド) があります。

プロセスの優先順位の変更 (nice)

ユーザーとして、プロセスの優先順位を下げるすることができます。管理者になり、プロセスの優先順位を上げたり下げたりします。

- ユーザーとして、nice 値を大きくすることでコマンドの優先順位を下げるすることができます。

次の `nice` コマンドは、`nice` 値を 5 単位分大きくすることで、`command-name` を実行する優先順位を下げます。

```
$ /usr/bin/nice -5 command-name
```

このコマンドでは、マイナス記号は次にくるものがオプションであることを表します。このコマンドは、次のように指定することもできます。

```
$ /usr/bin/nice -n 5 command-name
```

次の `nice` コマンドは、`nice` 値をデフォルトの 10 単位分大きくすることで、`command-name` の優先順位を下げますが、最大値の 39 を超えさせることはできません。

```
$ /usr/bin/nice command-name
```

- 管理者として、`nice` 値を変更してコマンドの優先順位を上げたり下げたりできません。

次の `nice` コマンドは、`nice` 値を 10 単位下げることによって、`command-name` の優先順位を上げます。最小値の 0 を下回することはできません。

```
# /usr/bin/nice --10 command-name
```

このコマンドでは、最初のマイナス記号は次にくるものがオプションであることを表します。2 番目のマイナス記号は負の数を表します。

次の `nice` コマンドは、`nice` 値を 5 単位上げることで、`command-name` の優先順位を下げます。最大値の 39 を超えることはできません。

```
# /usr/bin/nice -5 command-name
```

詳細は、[nice\(1\)](#) のマニュアルページを参照してください。

システムのプロセスに関するトラブルシューティング方法

発生する可能性のある一般的なシステムプロセスの問題は次のとおりです。

- 同じユーザーが所有する複数の同じジョブがないかどうかを調べます。ジョブが終了するまで待たずに多数のバックグラウンドジョブを起動するスクリプトを実行した場合に、この問題が発生することがあります。
- CPU 時間が大量に増えているプロセスがないかどうかを調べます。この問題を調べるには、`ps` 出力の `TIME` フィールドを確認します。この値は、プロセスが無限ループになっていることを示す場合があります。
- 実行中のプロセスの優先順位が高すぎないかどうかを調べます。`ps -c` コマンドを使用して `CLS` フィールドを調べると、各プロセスのスケジューラクラスが表示されます。リアルタイム (RT) プロセスとして実行中のプロセスが CPU を独占している可能性があります。また、`nice` 値の高いタイムシェアリング (TS) プロセスがな

いかどうかを調べます。管理者がプロセスの優先順位を上げている可能性があります。システム管理者は、`nice` コマンドを使用して優先順位を下げるができます。

- CPU 時間がしだいに増加し、制御がきかなくなったプロセスを調べます。プロセスの開始時間 (STIME) と、その後の CPU 時間 (TIME) の累積を調べると、この問題を特定できます。

◆◆◆ 第 3 章

システムパフォーマンスのモニタリング

コンピュータやネットワークのパフォーマンスを十分に引き出すことは、システム管理における重要な作業です。この章では、コンピュータシステムのパフォーマンスの管理に影響するいくつかの要素について説明します。また、この章では `vmstat`、`iostat`、`df`、および `sar` のコマンドを使用してシステムパフォーマンスをモニタリングする手順についても説明します。

この章で扱う内容は、次のとおりです。

- 45 ページの「システムパフォーマンスのモニタリングに関する情報の検索場所」
- 46 ページの「システムパフォーマンスに影響を与えるシステムリソースについて」
- 47 ページの「プロセスとシステムのパフォーマンスについて」
- 48 ページの「システムパフォーマンスのモニタリング」
- 49 ページの「システムパフォーマンス情報の表示」
- 56 ページの「システム動作のモニタリング」

システムパフォーマンスのモニタリングに関する情報の検索場所

システムパフォーマンスタスク	詳細情報
プロセスの管理	第2章「システムプロセスの管理」
システムパフォーマンスのモニター	第3章「システムパフォーマンスのモニタリング」
調整可能パラメータの変更	『Oracle Solaris 11.3 カーネルのチューンアップ・リファレンスマニュアル』
システムパフォーマンスタスクの管理	『Oracle Solaris 11.3 でのリソースの管理』の第2章、「プロジェクトとタスクについて」
FX および FS スケジューラを使用したプロセス管理	『Oracle Solaris 11.3 でのリソースの管理』の第8章、「公平配分スケジューラについて」

Oracle Enterprise Manager Ops Center を使用したパフォーマンスの管理

単に個々のシステム内のパフォーマンスをモニターするのではなく、大規模な配備内の物理および仮想オペレーティングシステム、サーバー、およびストレージデバイスのパフォーマンスをモニター、分析、および改善する必要がある場合は、Oracle Enterprise Manager Ops Center で使用可能な包括的なシステム管理ソリューションを使用できます。

Enterprise Manager Ops Center のモニター機能は、大規模な配備でモニターされるオペレーティングシステムとゾーンに関する広範な情報を提供します。この情報を使用して、パフォーマンスの評価、問題の特定、およびチューニングの実行が可能になります。Oracle Solaris Zones、Oracle VM Server for SPARC、および Oracle VM Server for x86 ゲストを含む、Linux および OS 仮想化テクノロジー用に、Oracle Solaris オペレーティングシステムで分析が可能です。

詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=oc122>を参照してください。

システムパフォーマンスに影響を与えるシステムリソースについて

コンピュータシステムのパフォーマンスは、システムがリソースをどのように使用し、割り当てるかによって変わります。したがって、通常の条件下でどのように動作するかを知るために、システムパフォーマンスを定期的にモニターする必要があります。期待できるパフォーマンスについてよく把握し、問題が発生したときに分析できなければなりません。

パフォーマンスに影響を与えるシステムリソースは次のとおりです。

中央処理装置 (CPU)	CPU は、命令をメモリーからフェッチして実行します。
入出力 (I/O) デバイス	I/O デバイスは、コンピュータとの間で情報をやりとりします。この種のデバイスには、端末とキーボード、ディスクドライブ、プリンタなどがあります。
メモリー	物理 (またはメイン) メモリーは、システム上のランダムアクセスメモリー (RAM) の容量を示します。

[第3章「システムパフォーマンスのモニタリング」](#)では、システムの動作とパフォーマンスに関する統計を表示するツールについて説明します。

プロセスとシステムのパフォーマンスについて

プロセスに関連する用語は次のとおりです。

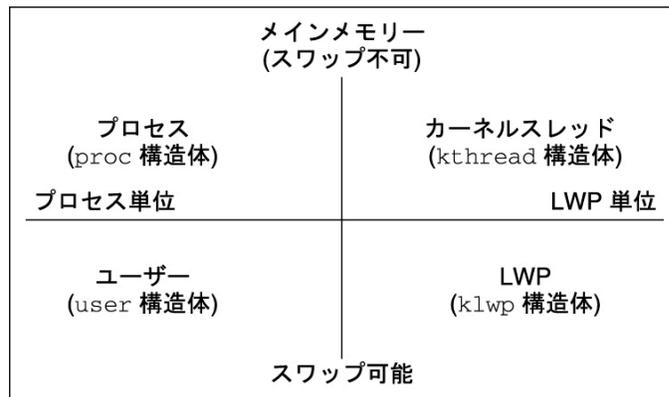
プロセス	システムの動作またはジョブ。システムをブートしてコマンドを実行するか、アプリケーションを起動するたびに、システムは1つ以上のプロセスをアクティブにする
軽量プロセス (LWP)	仮想 CPU または実行リソース。LWP は、利用できる CPU リソースをスケジューリングクラスと優先順位に基づいて使用するよう、カーネルによってスケジューリングされる。LWP には、スワップ可能な情報と、メモリーに常駐する情報を含むカーネルスレッドが含まれます。
アプリケーションスレッド	ユーザーのアドレス空間内で独立して実行できる別個のスタックを持った一連の命令。アプリケーションスレッドは LWP の最上部で多重化できる

1つのプロセスは、複数の LWP と複数のアプリケーションスレッドで構成できます。カーネルは、Oracle Solaris 環境内のスケジューリングエンティティとなる、カーネルスレッド構造をスケジューリングします。各種のプロセス構造体は次のとおりです。

proc	プロセス全体に関連し、メインメモリーに常駐しなければならない情報が入っている
kthread	1つの LWP に関連し、メインメモリーに常駐しなければならない情報が入っている
user	スワップ可能な、プロセス単位の情報が入っている
k1wp	スワップ可能な、LWP プロセス単位の情報が入っている

次の図に、これらのプロセス構造体の関係を示します。

図 1 プロセス構造体の関係



プロセス内のすべてのスレッドは、ほとんどのプロセスリソースにアクセスできません。ほとんどすべてのプロセスの仮想メモリーが共有されます。あるスレッドが共有データを変更すると、その変更結果をプロセス内の他のスレッドが利用できます。

システムパフォーマンスのモニタリング

コンピュータの稼働中は、各種のシステム動作を追跡するためにオペレーティングシステムのカウンタが増分されます。

追跡されるシステム動作は次のとおりです。

- 中央処理装置 (CPU) の使用状況
- バッファの使用状況
- ディスクとテープの入出力 (I/O) 動作
- 端末デバイスの動作
- システムコールの動作
- コンテキスト切り替え
- ファイルアクセス
- 待ち行列の動作
- カーネルテーブル
- プロセス間通信
- ページング
- 空きメモリーとスワップ空間

- カーネルメモリー割り当て (KMA)

モニタリングツール

Oracle Solaris ソフトウェアでは、システムの実行状況を追跡できる複数のツールが用意されています。

表 4 パフォーマンスモニタリングツール

コマンド	説明	詳細情報
cpustat と cputrack コマンド	CPU パフォーマンスカウンタを使用し、システムのパフォーマンスまたはプロセスをモニターする	cpustat(1M) および cputrack(1)
netstat コマンドと nfsstat コマンド	ネットワークパフォーマンスについての情報を表示する	netstat(1M) および nfsstat(1M)
ps コマンドと prstat コマンド	活動中のコマンドについての情報を表示する	第2章「システムプロセスの管理」
sar コマンドと sadc コマンド	システム動作データを収集および報告する	第3章「システムパフォーマンスのモニタリング」
swap コマンド	ユーザーのシステムで利用可能なスワップ領域についての情報を表示する	『Oracle Solaris 11.3 でのファイルシステムの管理』の第3章、「追加スワップ空間の構成」
vmstat コマンドと iostat コマンド	システム動作データの要約。仮想メモリーの統計、ディスクの使用率、CPU の動作など	第3章「システムパフォーマンスのモニタリング」
kstat コマンドと mpstat コマンド	システムで使用可能なカーネル統計 (kstats) を検査し、コマンド行で指定された基準に一致する統計を報告する。mpstat コマンドは、プロセッサ統計を表形式で報告する	kstat(1M) および mpstat(1M) のマニュアルページ。

システムパフォーマンス情報の表示

このセクションでは、システムパフォーマンス情報のモニタリングおよび表示のタスクについて説明します。

仮想メモリーの統計情報を表示する

vmstat コマンドを使用すると、仮想メモリーの統計情報と、CPU のロード、ページング、コンテキスト切り替え数、デバイス割り込み、システムコールなどの、システムイベントに関する情報を表示できます。また、vmstat コマンドを使用すると、スワップ、キャッシュフラッシュ、および割り込みに関する統計情報も表示できます。

表 5 vmstat コマンドからの出力

カテゴリ	フィールド名	説明
procs	r	ディスパッチ待ち行列内のカーネルスレッド数
	b	リソースを待機中のブロックされたカーネルスレッド数
	w	リソース処理の完了を待機中のスワップアウトされた軽量プロセス数
memory		実メモリと仮想メモリの使用状況を表示します
	swap	使用可能なスワップ空間
	free	空きリストのサイズ
page		ページフォルトとページング動作を 1 秒当たりの単位数として表示します
	re	回収されたページ数
	mf	軽度の障害と重大な障害
	pi	ページインされた K バイト数
	po	ページアウトされた K バイト数
	fr	解放された K バイト数
	de	最後にスワップインされたプロセスに必要なだと予想されるメモリ
	sr	page デーモンによって走査され、現在は使用されていないページ数。sr が 0 以外の値であれば、page デーモンが実行されています。
	disk	最高 4 台のディスク上のデータを示す、1 秒当たりのディスク処理数を表示します
faults		トラップ/割り込み率 (1 秒当たり) を表示します
	in	1 秒当たりの割り込み数
	sy	1 秒当たりのシステムコール数
	cs	CPU のコンテキスト切替え率
cpu		CPU 時間の使用状況を表示します
	us	ユーザー時間
	sy	システム時間
	id	アイドル時間

このコマンドの詳細については、[vmstat\(1M\)](#) のマニュアルページを参照してください。

仮想メモリの統計情報の表示 (vmstat)

仮想メモリの統計を表示するには、秒単位で時間間隔を指定して vmstat コマンドを使用します。

```
$ vmstat n
```

n は、秒単位で表した報告間隔です。

次の例に、5 秒間隔で収集された統計情報に関する `vmstat` の表示を示します。

```
$ vmstat 5
kthr      memory          page                disk                faults              cpu
r  b  w    swap free re  mf pi po fr de sr dd f0 s1 --  in  sy  cs us sy id
0  0  0  863160 365680 0   3  1  0  0  0  0  0  0  0  406 378 209  1  0 99
0  0  0  765640 208568 0  36  0  0  0  0  0  0  0  0  479 4445 1378  3  3 94
0  0  0  765640 208568 0   0  0  0  0  0  0  0  0  0  423  214  235  0  0 100
0  0  0  765712 208640 0   0  0  0  0  0  0  0  3  0  412  158  181  0  0 100
0  0  0  765832 208760 0   0  0  0  0  0  0  0  0  0  402  157  179  0  0 100
0  0  0  765832 208760 0   0  0  0  0  0  0  0  0  0  403  153  182  0  0 100
0  0  0  765832 208760 0   0  0  0  0  0  0  0  0  0  402  168  177  0  0 100
0  0  0  765832 208760 0   0  0  0  0  0  0  0  0  0  402  153  178  0  0 100
0  0  0  765832 208760 0  18  0  0  0  0  0  0  0  0  407  165  186  0  0 100
```

システムイベント情報の表示 (`vmstat -s`)

`vmstat -s` コマンドを実行すると、システムを前回ブートしたあとに発生したシステムイベントの数が表示されます。

```
$ vmstat -s
 0 swap ins
 0 swap outs
 0 pages swapped in
 0 pages swapped out
522586 total address trans. faults taken
17006 page ins
 25 page outs
23361 pages paged in
 28 pages paged out
45594 total reclaims
45592 reclaims from free list
 0 micro (hat) faults
522586 minor (as) faults
16189 major faults
98241 copy-on-write faults
137280 zero fill page faults
45052 pages examined by the clock daemon
 0 revolutions of the clock hand
 26 pages freed by the clock daemon
2857 forks
 78 vforks
1647 execs
34673885 cpu context switches
65943468 device interrupts
711250 traps
63957605 system calls
3523925 total name lookups (cache hits 99%)
 92590 user cpu
 65952 system cpu
16085832 idle cpu
 7450 wait cpu
```

スワッピング統計の表示 (vmstat -s)

vmstat -s を実行すると、スワップの統計情報が表示されます。

```
$ vmstat -s
kthr      memory          page        disk        faults       cpu
 r  b  w  swap  free  si  so pi po fr de sr dd f0 s1 --  in  sy  cs us sy id
  0  0  0 862608 364792  0  0  1  0  0  0  0  0  0  0  406 394 213  1  0 99
```

スワッピング統計情報のフィールドを次のリストに示します。その他のフィールドの説明については、[表5](#)を参照してください。

si	1 秒あたりにスワップされた平均軽量プロセス数
so	スワップアウトされた全プロセス数

注記 - vmstat コマンドは、 si フィールドおよび so フィールドの出力値を切り捨てます。スワップ統計情報の詳細情報を表示するには、sar コマンドを使用してください。

デバイス当たりの割り込み数の表示 (vmstat -i)

vmstat -i コマンドを実行すると、デバイス当たりの割り込み数が表示されます。

次の例は、vmstat -i コマンドからの出力を示します。

```
$ vmstat -i
interrupt      total      rate
-----
clock          52163269    100
esp0            2600077      4
zsc0            25341        0
zsc1            48917        0
cgsixc0         459          0
lec0            400882       0
fdc0            14           0
bppc0           0            0
audiocs0        0            0
-----
Total          55238959    105
```

ディスクの使用状況を表示する

iostat コマンドを使用すると、ディスクの入出力に関する統計情報を表示したり、スループット、使用率、待ち行列の長さ、トランザクション率、およびサービス時間の統計情報を表示したりできます。このコマンドの詳細については、[iostat\(1M\)](#)のマニュアルページを参照してください。

ディスク使用状況の情報の表示 (iostat)

秒単位で時間間隔を指定して `iostat` コマンドを使用すると、ディスクの使用状況の情報を表示できます。

```
$ iostat 5
      tty          fd0          sd3          nfs1          nfs31          cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
  0   1   0   0 410    3   0  29    0   0   9    3   0  47    4  2  0 94
```

出力の 1 行目は、今回のブート以降の統計情報を示します。2 行目以降は、時間間隔ごとの統計情報を示します。デフォルトでは、端末 (`tty`)、ディスク (`fd` と `sd`)、および CPU (`cpu`) の統計情報が表示されます。

次の例は、5 秒間隔で収集されたディスク統計情報を示します。

```
$ iostat 5
tty          sd0          sd6          nfs1          nfs49          cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
  0   0   1   0  49    0   0   0    0   0   0    0   0  15    0  0  0 100
  0  47   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  100
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  100
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  100
  0  16  44   6 132    0   0   0    0   0   0    0   0   0    0  0  0  1 99
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  0 100
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  0 100
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  0 100
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  0 100
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  0 100
  0  16  3   1  23    0   0   0    0   0   0    0   0   0    0  0  0  1 99
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  0 100
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  0 100
  0  16   0   0   0    0   0   0    0   0   0    0   0   0    0  0  0  0 100
```

次の表に、`iostat n` コマンド出力内のフィールドを示します。

デバイスタイプ	フィールド名	説明
端末	<code>tin</code>	端末の入力待ち行列内の文字数
	<code>tout</code>	端末の出力待ち行列内の文字数
ディスク	<code>bps</code>	1 秒当たりのブロック数
	<code>tps</code>	1 秒当たりのトランザクション数
	<code>serv</code>	ミリ秒単位で表した平均サービス時間
CPU	<code>us</code>	ユーザーモード
	<code>sy</code>	システムモード
	<code>wt</code>	入出力待機中
	<code>id</code>	アイドル状態

拡張ディスク統計情報の表示 (iostat -xtc)

iostat −xt コマンドを実行して、拡張ディスクの統計情報を表示します。

```
$ iostat &minus;xt
device      r/s      w/s      kr/s      kw/s  wait  actv  wsvc_t  asvc_t  %w  %b  tin tout
blkdev0     0.0      0.0      0.0      0.0  0.0  0.0   0.0    0.0    0  0   0   1
sd0         0.1     19.3      1.4     92.4  0.0  0.0   0.2    1.6    0  1
sd1         0.0      0.0      0.0      0.0  0.0  0.0   0.0    0.0    0  0
nfs9        0.0      0.0      0.0      0.0  0.0  0.0   0.0    1.0    0  0
nfs10       0.0      0.0      0.0      0.0  0.0  0.0   0.0    7.6    0  0
nfs11       0.0      0.0      0.0      0.0  0.0  0.0   0.0   15.6    0  0
nfs12       0.3      0.0      1.9      0.0  0.0  0.0   0.0   30.5    0  1
```

iostat −xt コマンドを使用すると、ディスクごとに 1 行ずつ出力が表示されます。出力フィールドは次のとおりです。

r/s	1 秒当たりの読み取り数
w/s	1 秒当たりの書き込み数
kr/s	1 秒当たりの読み取りキロバイト数
kw/s	1 秒当たりの書き込みキロバイト数
wait	サービス (待ち行列の長さ) を待機中の平均トランザクション数
actv	サービス中の平均トランザクション数
svc_t	ミリ秒単位で表した平均サービス時間
%w	待ち行列が空でない時間の割合
%b	ディスクがビジーである時間の割合

ディスク容量統計情報の表示 (df)

df コマンドを使用すると、マウントされている各ディスク上の空きディスク容量が表示されます。レポート用の統計情報では使用可能容量の合計の内先頭に 10% の空き容量を残しておくので、df から報告される usable ディスク容量は全容量の 90% のみに相当します。この「先頭の空き容量」は、パフォーマンスを高めるために常に空になっています。

実際に df コマンドからレポートされるディスク容量の割合は、使用済み容量を使用可能容量で割った値です。

ファイルシステムの容量が90%を超える場合、`cp` コマンドを使用して空いているディスクにファイルを転送できます。また、`tar` コマンドか `cpio` コマンドを使用してテープにファイルを転送することもできます。また、ファイルを削除することもできます。

このコマンドの詳細については、[df\(1M\)](#) のマニュアルページを参照してください。

ディスク容量の情報の表示 (`df -k`)

`df -k` コマンドを使用すると、ディスク容量情報がキロバイト単位で表示されます。

```
$ df -k
Filesystem            kbytes    used  avail capacity  Mounted on
/dev/dsk/c0t3d0s0    192807   40231  133296    24%      /
```

例 15 ファイルシステム情報を表示する

次の例は、SPARC システムの `df -k` コマンドの出力を示しています。

```
$ df -k
Filesystem            1024-blocks    Used  Available Capacity  Mounted on
rpool/ROOT/solaris-161  191987712    6004395  140577816    5%      /
/devices                0            0          0    0%    /devices
/dev                    0            0          0    0%    /dev
ctfs                    0            0          0    0%    /system/contract
proc                    0            0          0    0%    /proc
mnttab                  0            0          0    0%    /etc/mnttab
swap                    4184236      496      4183740    1%    /system/volatile
objfs                    0            0          0    0%    /system/object
sharefs                  0            0          0    0%    /etc/dfs/sharetab
/usr/lib/libc/libc_hwcapi.so.1  146582211    6004395  140577816    5%    /lib/libc.so.1
fd                       0            0          0    0%    /dev/fd
swap                     4183784      60      4183724    1%    /tmp
rpool/export            191987712     35     140577816    1%    /export
rpool/export/home      191987712     32     140577816    1%    /export/home
rpool/export/home/123  191987712   13108813  140577816    9%    /export/home/123
rpool/export/repo      191987712  11187204  140577816    8%    /export/repo
rpool/export/repo2010_11  191987712     31     140577816    1%    /export/repo2010_11
rpool                    191987712   5238974  140577816    4%    /rpool
/export/home/123       153686630   13108813  140577816    9%    /home/123
```

`df -k` コマンドの出力フィールドは次のとおりです。

1024-blocks	ファイルシステム内の使用可能容量の合計
Used	使用されている容量
Available	使用可能容量
Capacity	使用されている容量が全容量に占める割合
Mounted on	マウントポイント

例 16 オプションを指定しない `df` コマンドを使用してファイルシステム情報を表示する

オペランドやオプションを指定せずに `df` コマンドを使用すると、次の例に示すように、マウントされているすべてのファイルシステムが報告されます。

```
$ df
/ (rpool/ROOT/solaris):100715496 blocks 100715496 files
/devices (/devices): 0 blocks 0 files
/dev (/dev): 0 blocks 0 files
/system/contract (ctfs): 0 blocks 2147483601 files
/proc (proc): 0 blocks 29946 files
/etc/mnttab (mnttab): 0 blocks 0 files
/system/volatile (swap):42257568 blocks 2276112 files
/system/object (objfs): 0 blocks 2147483441 files
/etc/dfs/sharetab (sharefs): 0 blocks 2147483646 files
/dev/fd (fd): 0 blocks 0 files
/tmp (swap):42257568 blocks 2276112 files
/export (rpool/export):100715496 blocks 100715496 files
/export/home (rpool/export/home):100715496 blocks 100715496 files
/export/home/admin (rpool/export/home/admin):100715496 blocks 100715496 files
/rpool (rpool):100715496 blocks 100715496 files
/export/repo2010_11(rpool/export/repo2010_11):281155639 blocks 281155639 files
/rpool (rpool):281155639 blocks 281155639 files
```

システム動作のモニタリング

このセクションでは、システム動作のモニタリングのアクティビティについて説明します。

システム動作のモニタリング (sar)

次のタスクを実行するには、`sar` コマンドを使用します。

- システム動作についてのデータを編成し表示します。
- 特別な要求に基づいて、システム動作データにアクセスします。
- システムパフォーマンスを測定およびモニターするレポートを自動的に生成し、特定のパフォーマンス障害を正確に突き止めるための、特別なリクエストレポートも生成します。`sar` コマンドを設定してシステム上で実行する方法と、これらのツールの説明については、[72 ページの「システム動作データの自動収集 \(sar\)」](#)を参照してください。

このコマンドの詳細については、[sar\(1\)](#) のマニュアルページを参照してください。

ファイルアクセスをチェックする (sar -a)

`sar -a` コマンドでファイルアクセス操作の統計情報を表示します。

```

$ sar -a

SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v    ...

00:00:00  iget/s namei/s dirbk/s
01:00:00      0      3      0
02:00:00      0      3      0
03:00:00      0      3      0
04:00:00      0      3      0
05:00:00      0      3      0
06:00:00      0      3      0
07:00:00      0      3      0
08:00:00      0      3      0
08:20:01      0      3      0
08:40:00      0      3      0
09:00:00      0      3      0
09:20:01      0     10      0
09:40:01      0      1      0
10:00:02      0      5      0

Average      0      4      0

```

`sar -a` コマンドによって報告されるオペレーティングシステムのルーチンは、次のとおりです。

`iget/s` ディレクトリ名検索キャッシュ (DNLC) 内に入っていない `i` ノードに対して出された要求数

`namei/s` 1 秒当たりのファイルシステムパスの検索数。 `namei` で DNLC 内にディレクトリ名が見つからない場合は、 `iget` が呼び出され、ファイルまたはディレクトリの `i` ノードが取得される。したがって、ほとんどの `igets` は DNLC が欠落した結果である

`dirbk/s` 1 秒間に実行されたディレクトリブロックの読み取り回数

これらのオペレーティングシステムルーチンに対してレポートされる値が大きい程、カーネルはユーザーファイルへのアクセスに長い時間を費やしています。この時間には、プログラムとアプリケーションによるファイルシステムの使用量が反映されません。 `-a` オプションを使用すると、アプリケーションのディスク依存度を表示できるので便利です。

バッファー動作をチェックする (`sar -b`)

`sar -b` コマンドを使用すると、バッファー動作の統計情報が表示されます。

バッファーは、メタデータをキャッシュするために使用されます。メタデータには、 `i` ノード、シリンダグループブロック、間接ブロックなどがあります。

```

$ sar -b
00:00:00  bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0     100      0      0     55      0      0

```

次の表は、`-b` オプションを指定したときに表示されるバッファ動作を示します。

フィールド名	説明
<code>bread/s</code>	ディスクからバッファークャッシュに投入された 1 秒当たりの平均読み取り数
<code>lread/s</code>	バッファークャッシュからの 1 秒当たりの平均論理読み取り数
<code>%rcache</code>	バッファークャッシュ内で見つかった論理読み込み数の割合 (<code>lread/s</code> に対する <code>bread/s</code> の比を 100% から差し引いた値)
<code>bwrit/s</code>	バッファークャッシュからディスクに書き込まれた 1 秒当たりの平均物理ブロック数 (512 バイト)
<code>lwrit/s</code>	バッファークャッシュへの 1 秒当たりの平均論理書き込み数
<code>%wcache</code>	バッファークャッシュ内で見つかった論理書き込み数の割合 (<code>lwrit/s</code> に対する <code>bwrit/s</code> の比を 100% から差し引いた値)
<code>pread/s</code>	キャラクタ型デバイスインタフェースを使用する 1 秒当たりの平均物理読み取り数
<code>pwrit/s</code>	キャラクタ型デバイスインタフェースを使用する 1 秒当たりの平均物理書き込みリクエスト数

もっとも重要なエントリは、キャッシュヒット率 `%rcache` と `%wcache` です。これらのエントリは、システムバッファリングの有効性を測定します。`%rcache` が 90% 未満の場合や、`%wcache` が 65% 未満の場合は、バッファ領域を大きくすればパフォーマンスを改善できる可能性があります。

例 17 バッファ動作をチェックする (`sar -b`)

次の `sar -b` コマンド出力の例は、`%rcache` および `%wcache` バッファが速度低下を引き起こしていないことを示しています。すべてのデータは許容範囲に収まっています。

```
$ sar -b
SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v ...
00:00:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0    100      0      0      94      0      0
02:00:01      0      0    100      0      0      94      0      0
03:00:00      0      0    100      0      0      92      0      0
04:00:00      0      1    100      0      1      94      0      0
05:00:00      0      0    100      0      0      93      0      0
06:00:00      0      0    100      0      0      93      0      0
07:00:00      0      0    100      0      0      93      0      0
08:00:00      0      0    100      0      0      93      0      0
08:20:00      0      1    100      0      1      94      0      0
08:40:01      0      1    100      0      1      93      0      0
09:00:00      0      1    100      0      1      93      0      0
09:20:00      0      1    100      0      1      93      0      0
09:40:00      0      2    100      0      1      89      0      0
10:00:00      0      9    100      0      5      92      0      0
10:20:00      0      0    100      0      0      68      0      0
10:40:00      0      1     98      0      1      70      0      0
```

```
11:00:00      0      1    100      0      1     75      0      0
Average      0      1    100      0      1     91      0      0
```

システムコールの統計情報をチェックする (sar -c)

sar -c コマンドを使用すると、システムコールの統計情報が表示されます。

```
$ sar -c
00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      38      2      2   0.00   0.00   149   120
```

次のリストは、-c オプションによって報告されるシステムコールカテゴリの説明です。一般に、読み取りと書き込みがシステムコール合計の約半分を占めます。ただし、システムで実行中の動作によってこの割合は大幅に変動します。

scall/s	1 秒当たりのすべてのタイプのシステムコール数 (通常は、4 ユーザーから 6 ユーザーのシステム上で 1 秒当たり約 30)。
sread/s	1 秒当たりの read システムコール数
swrit/s	1 秒当たりの write システムコール数
fork/s	1 秒当たりの fork システムコール数 (4 ユーザーから 6 ユーザーのシステム上で毎秒約 0.5)。この数値は、シェルスクリプトの実行中は大きくなる
exec/s	1 秒当たりの exec システムコール数。exec/s を fork/s で割った値が 3 より大きい場合は、効率の悪い PATH 変数を調べる。
rchar/s	read システムコールによって転送される 1 秒当たりの文字 (バイト) 数
wchar/s	write システムコールによって転送される 1 秒当たりの文字 (バイト) 数

例 18 システムコールの統計情報をチェックする (sar -c)

次の例は、sar -c コマンドからの出力を示します。

```
$ sar -c
SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:04 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      89      14      9   0.01   0.00   2906   2394
02:00:01      89      14      9   0.01   0.00   2905   2393
03:00:00      89      14      9   0.01   0.00   2908   2393
04:00:00      90      14      9   0.01   0.00   2912   2393
05:00:00      89      14      9   0.01   0.00   2905   2393
06:00:00      89      14      9   0.01   0.00   2905   2393
07:00:00      89      14      9   0.01   0.00   2905   2393
```

```

08:00:00      89      14       9    0.01    0.00    2906    2393
08:20:00      90      14       9    0.01    0.01    2914    2395
08:40:01      90      14       9    0.01    0.00    2914    2396
09:00:00      90      14       9    0.01    0.01    2915    2396
09:20:00      90      14       9    0.01    0.01    2915    2396
09:40:00     880     207     156    0.08    0.08   26671    9290
10:00:00    2020     530     322    0.14    0.13   57675   36393
10:20:00     853     129      75    0.02    0.01   10500    8594
10:40:00    2061     524     450    0.08    0.08  579217  567072
11:00:00    1658     404     350    0.07    0.06 1152916 1144203

Average      302      66      49    0.02    0.01   57842   55544

```

ディスク動作をチェックする (sar -d)

sar -d コマンドを使用すると、ディスク動作の統計情報が表示されます。

```
$ sar -d
```

```
00:00:00 device          %busy  avque  r+w/s  blks/s  await  avserv
```

次のリストは、-d オプションを指定したときに報告されるディスクデバイスの動作を示します。

device	モニター中のディスクデバイス名
%busy	デバイスが転送要求のサービスでビジーであった時間
avque	デバイスが転送要求のサービスでビジーであった時間の間の平均要求数
r+w/s	デバイスへの 1 秒当たりの読み取り転送数と書き込み転送数
blks/s	デバイスに転送される 1 秒当たりの 512 バイトブロック数
await	待ち行列内の転送リクエストの平均時間 (ミリ秒単位)。これは、待ち行列が占有されている場合のみ計測される
avserv	デバイスが 1 転送要求の処理に要する平均時間 (ミリ秒単位)。ディスクの場合は、この値にシークタイム、回転待ち時間、およびデータ転送時間が含まれる

例 19 ディスク動作をチェックする

次の例は、sar -d コマンドからの一部省略した出力を示します。

```
$ sar -d
```

```
SunOS balmy 5.10 Generic_144500-10 sun4v ...
```

```
12:36:32 device          %busy  avque  r+w/s  blks/s  await  avserv
```

```

12:40:01  dad1          15    0.7    26    399    18.1    10.0
          dad1,a       15    0.7    26    398    18.1    10.0
          dad1,b        0    0.0     0     1     1.0     3.0
          dad1,c        0    0.0     0     0     0.0     0.0
          dad1,h        0    0.0     0     0     0.0     6.0
          fd0          0    0.0     0     0     0.0     0.0
          nfs1         0    0.0     0     0     0.0     0.0
          nfs2         1    0.0     1    12     0.0    13.2
          nfs3         0    0.0     0     2     0.0     1.9
          nfs4         0    0.0     0     0     0.0     7.0
          nfs5         0    0.0     0     0     0.0    57.1
          nfs6         1    0.0     6   125     4.3     3.2
          nfs7         0    0.0     0     0     0.0     6.0
          sd1          0    0.0     0     0     0.0     5.4
          ohci0,bu     0    0.0     0     0     0.0     0.0
          ohci0,ct     0    0.0     0     0     0.0     0.0
          ohci0,in     0    0.0     7     0     0.0     0.0
          ohci0,is     0    0.0     0     0     0.0     0.0
          ohci0,to     0    0.0     7     0     0.0     0.0

```

待ち行列内に何かがあるときは、待ち行列の長さや待ち時間が計測されるので注意してください。%busy の値が小さい場合に、待ち行列とサービス時間が大きければ、変更されたブロックをディスクに随時書き込むために、システムが定期的に処理していることを示す場合があります。

ページアウトとメモリーをチェックする (sar -g)

sar -g コマンドを使用すると、ページアウト動作の平均値とメモリー解放動作の平均値とが表示されます。

```

$ sar -g
00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00  0.00    0.00    0.00    0.00    0.00

```

sar -g コマンドで表示される出力は、より多くのメモリーが必要かどうかを判断するのに役立ちます。ps -elf コマンドを使用すると、page デーモンに使用される CPU サイクル数が表示されます。サイクル数が大きく、pgfree/s フィールドと pgscan/s フィールドの値が大きければ、メモリー不足を示します。

また、sar -g を使用すると、i ノードの再利用間隔が短すぎるために、再利用可能なページが失われているかどうか也表示されます。

次のリストでは -g オプションからの出力について説明します。

```

pgout/s          1 秒間にページアウトされた要求数

ppgout/s         1 秒間に実際にページアウトされたページ数。1つのページアウト
                  要求で複数のページがページアウトされることがあります。

pgfree/s         空きリストに配置された1秒当たりのページ数。

pgscan/s         page デーモンによって走査された1秒当たりのページ数。この値
                  が大きい場合は、page デーモンが空きメモリーのチェックに大量

```

の時間を費やしています。これは、メモリーを増やす必要があることを示します。

%ufs_ipf

ufs の i ノードがそれに関連付けられた再使用可能ページを持つ iget によって、空きリストから取り出された割合。これらのページはフラッシュされ、プロセスが回収できなくなります。したがって、このフィールドはページフラッシュを伴う igets の割合です。値が大きければ、i ノードの空きリストがページ境界であり、ufs の i ノード数を増やす必要があることを示します。

例 20 ページアウトとメモリーをチェックする (sar -g)

次の例は、sar -g コマンドからの出力を示します。

\$ sar -g

SunOS balmy 5.10 Generic_144500-10 sun4v ...

00:00:00	pgout/s	ppgout/s	pgfree/s	pgscan/s	%ufs_ipf
01:00:00	0.00	0.00	0.00	0.00	0.00
02:00:00	0.01	0.01	0.01	0.00	0.00
03:00:00	0.00	0.00	0.00	0.00	0.00
04:00:00	0.00	0.00	0.00	0.00	0.00
05:00:00	0.00	0.00	0.00	0.00	0.00
06:00:00	0.00	0.00	0.00	0.00	0.00
07:00:00	0.00	0.00	0.00	0.00	0.00
08:00:00	0.00	0.00	0.00	0.00	0.00
08:20:01	0.00	0.00	0.00	0.00	0.00
08:40:00	0.00	0.00	0.00	0.00	0.00
09:00:00	0.00	0.00	0.00	0.00	0.00
09:20:01	0.05	0.52	1.62	10.16	0.00
09:40:01	0.03	0.44	1.47	4.77	0.00
10:00:02	0.13	2.00	4.38	12.28	0.00
10:20:03	0.37	4.68	12.26	33.80	0.00
Average	0.02	0.25	0.64	1.97	0.00

カーネルメモリーの割り当てをチェックする

カーネルメモリーの割り当て (KMA) では、カーネルサブシステムが必要に応じてメモリーを割り当てたり解放したりします。

KMA では、ロードのピーク時に必要と思われる最大メモリー容量を静的に割り当てることをせず、メモリーのリクエストを次の 3 つのカテゴリに分けます。

- 「小」 (256 バイト未満)
- 「大」 (512 バイト - 4K バイト)
- 「サイズ超過」 (4K バイト超)

KMA は、2 つのメモリープールを管理して、「小」要求と「大」要求を満たします。「サイズ超過」要求は、システムページアロケータからメモリーを割り当てることで満たされます。

KMA リソースを使用するドライバや STREAMS の書き込みに使用中のシステムを調査する場合は、`sar -k` コマンドを使用すると便利です。KMA リソースを使用するが、終了前には特にリソースを返さないドライバやモジュールがあると、メモリーのリークが生じることがあります。メモリーリークが発生すると、KMA によって割り当てられるメモリーは時間が経つにつれて増大します。したがって、`sar -k` コマンドの `alloc` フィールドの値が時間が経つにつれ増える場合は、メモリーリークの可能性ががあります。メモリーリークのもう 1 つの兆候は、要求が失敗することです。この問題が発生した場合は、メモリーリークのために KMA がメモリーを予約したり割り当てたりできなくなっている可能性があります。

メモリーリークが発生した場合は、KMA からメモリーを要求したが返していないドライバや STREAMS がないかどうかをチェックする必要があります。

カーネルメモリーの割り当てをチェックする(`sar -k`)

`sar -k` コマンドを使用すると、KMA の動作が報告されます。

```
$ sar -k
00:00:00 sml_mem  alloc  fail  lg_mem  alloc  fail  ovsz_alloc  fail
01:00:00 2523136 1866512    0 18939904 14762364    0    360448    0
02:00:02 2523136 1861724    0 18939904 14778748    0    360448    0
```

次のリストに `-k` オプションからの出力について説明します。

<code>sml_mem</code>	KMA が小メモリーリクエストプール内で使用できるメモリーのバイト数。このプールでは、小リクエストは 256 バイト未満。
<code>alloc</code>	KMA が小メモリーリクエストプールから小メモリーリクエストに割り当てたメモリーのバイト数。
<code>fail</code>	失敗した小メモリー要求数
<code>lg_mem</code>	KMA が大メモリーリクエストプール内で使用できるメモリーのバイト数。(このプールでは、大要求は 512 バイトから 4K バイトまで)
<code>alloc</code>	KMA が大メモリーリクエストプールから大メモリーリクエストに割り当てたメモリーのバイト数。
<code>fail</code>	失敗した大メモリー要求数
<code>ovsz_alloc</code>	サイズ超過要求 (4K バイトを超える要求) に割り当てられたメモリーの容量。これらの要求はページアロケータによって満たされる。このため、プールはない
<code>fail</code>	失敗したサイズ超過メモリー要求数

例 21 カーネルメモリーの割り当てをチェックする(sar -k)

次の例は、簡略化された sar -k 出力を示します。

```
$ sar -k
SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:04 sml_mem alloc fail lg_mem alloc fail ovsz_alloc fail
01:00:00 6119744 4852865 0 60243968 54334808 156 9666560 0
02:00:01 6119744 4853057 0 60243968 54336088 156 9666560 0
03:00:00 6119744 4853297 0 60243968 54335760 156 9666560 0
04:00:00 6119744 4857673 0 60252160 54375280 156 9666560 0
05:00:00 6119744 4858097 0 60252160 54376240 156 9666560 0
06:00:00 6119744 4858289 0 60252160 54375608 156 9666560 0
07:00:00 6119744 4858793 0 60252160 54442424 156 9666560 0
08:00:00 6119744 4858985 0 60252160 54474552 156 9666560 0
08:20:00 6119744 4858169 0 60252160 54377400 156 9666560 0
08:40:01 6119744 4857345 0 60252160 54376880 156 9666560 0
09:00:00 6119744 4859433 0 60252160 54539752 156 9666560 0
09:20:00 6119744 4858633 0 60252160 54410920 156 9666560 0
09:40:00 6127936 5262064 0 60530688 55619816 156 9666560 0
10:00:00 6545728 5823137 0 62996480 58391136 156 9666560 0
10:20:00 6545728 5758997 0 62996480 57907400 156 9666560 0
10:40:00 6734144 6035759 0 64389120 59743064 156 10493952 0
11:00:00 6996288 6394872 0 65437696 60935936 156 10493952 0
Average 6258044 5150556 0 61138340 55609004 156 9763900 0
```

プロセス間通信をチェックする (sar -m)

sar -m コマンドを使用し、プロセス間通信の動作を報告します。

```
$ sar -m
00:00:00 msg/s sema/s
01:00:00 0.00 0.00
```

通常、これらの数字は、メッセージやセマフォアを使用するアプリケーションを実行していない限りゼロ (0.00) です。

-m オプションからの出力は次のとおりです。

msg/s 1 秒当たりのメッセージ処理 (送受信) 数

sema/s 1 秒当たりのセマフォア処理数

次の例は、sar -m コマンドからの一部省略した出力を示します。

```
$ sar -m
SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:00 msg/s sema/s
01:00:00 0.00 0.00
02:00:02 0.00 0.00
03:00:00 0.00 0.00
04:00:00 0.00 0.00
05:00:01 0.00 0.00
```

```
06:00:00    0.00    0.00
Average      0.00    0.00
```

ページイン動作をチェックする (sar -p)

sar -p コマンドを使用して、保護および変換フォルトを含むページイン動作を報告します。

```
$ sar -p
00:00:00 atch/s  pgin/s  ppgin/s  pflt/s  vflt/s  slock/s
01:00:00   0.07   0.00   0.00   0.21   0.39   0.00
```

次のリストに -p オプションから報告される統計情報を示します。

atch/s	現在メモリーに入っているページを回収して満たされる 1 秒当たりのページフォルト数 (1 秒当たりの付加数)。この例には、空きリストから無効なページを回収し、別のプロセスに現在使用中のテキストページを共有する処理が含まれます。たとえば、複数のプロセスが同じプログラムテキストにアクセスしている場合などです。
pgin/s	ファイルシステムがページインリクエストを受信する 1 秒当たりの回数。
ppgin/s	ページインされる 1 秒当たりのページ数。ソフトロック要求 (slock/s を参照) などの 1 つのページイン要求や、大型ブロックサイズでは、複数のページがページインされることがあります。
pflt/s	保護エラーによるページフォルト数。保護フォルトの例には、ページへの不正なアクセスや、書き込み時コピーなどがあります。通常、この数値は主に書き込み時コピーからなっています。
vflt/s	1 秒当たりのアドレス変換ページフォルト数。これらのフォルトは有効性フォルトと呼ばれ、仮想アドレスに有効なプロセステーブルエントリが存在しないときに発生します。
slock/s	物理入出力を必要とするソフトウェアロックリクエストによって発生する 1 秒当たりのフォルト数。ソフトロック要求の発生例には、ディスクからメモリーへのデータ転送などがあります。システムはデータを受信するページをロックするので、別のプロセスはそれを回収して使用できません。

例 22 ページイン動作をチェックする (sar -p)

次の例は、sar -p コマンドからの出力を示します。

```

$ sar -p
SunOS balmy 5.10 Generic_144500-10 sun4v  ...
00:00:04 atch/s  pgin/s  ppgin/s  pflt/s  vflt/s  slock/s
01:00:00  0.09   0.00   0.00   0.78   2.02   0.00
02:00:01  0.08   0.00   0.00   0.78   2.02   0.00
03:00:00  0.09   0.00   0.00   0.81   2.07   0.00
04:00:00  0.11   0.01   0.01   0.86   2.18   0.00
05:00:00  0.08   0.00   0.00   0.78   2.02   0.00
06:00:00  0.09   0.00   0.00   0.78   2.02   0.00
07:00:00  0.08   0.00   0.00   0.78   2.02   0.00
08:00:00  0.09   0.00   0.00   0.78   2.02   0.00
08:20:00  0.11   0.00   0.00   0.87   2.24   0.00
08:40:01  0.13   0.00   0.00   0.90   2.29   0.00
09:00:00  0.11   0.00   0.00   0.88   2.24   0.00
09:20:00  0.10   0.00   0.00   0.88   2.24   0.00
09:40:00  2.91   1.80   2.38   4.61  17.62   0.00
10:00:00  2.74   2.03   3.08   8.17  21.76   0.00
10:20:00  0.16   0.04   0.04   1.92   2.96   0.00
10:40:00  2.10   2.50   3.42   6.62  16.51   0.00
11:00:00  3.36   0.87   1.35   3.92  15.12   0.00
Average   0.42   0.22   0.31   1.45   4.00   0.00

```

待ち行列動作をチェックする (sar -q)

sar -q コマンドを使用し、次の情報を報告します。

- 待ち行列にリクエストが入っている間の平均待ち行列の長さ。
- 待ち行列に要求が入っている時間の割合

```

$ sar -q
00:00:00 runq-sz %runocc swpq-sz %swpocc

```

-q オプションからの出力は次のとおりです。

runq-sz	CPU を実行するためにメモリー内で待機中のカーネルスレッド数。通常、この値は 2 未満になる。値が常に 2 より大きい場合は、システムが CPU の限界に到達している可能性がある
%runocc	ディスパッチ待ち行列が使用されている時間の割合
swpq-sz	スワップアウトされるプロセスの平均数。
%swpocc	プロセスがスワップアウトされている時間の割合。

例 23 待ち行列動作をチェックする

次の例は、sar -q コマンドからの出力を示します。%runocc の値が大きく (90 パーセント超)、runq-sz の値が 2 より大きい場合は、CPU の負荷が大きく、応答速度が低下しています。この場合は、CPU の容量を増やしてシステムの応答速度を適正化する必要があります。

```
# sar -q
SunOS balmy 5.10 Generic_144500-10 sun4v  ...

00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00 1.0 7 0.0 0
02:00:00 1.0 7 0.0 0
03:00:00 1.0 7 0.0 0
04:00:00 1.0 7 0.0 0
05:00:00 1.0 6 0.0 0
06:00:00 1.0 7 0.0 0

Average 1.0 7 0.0 0
```

未使用のメモリーをチェックする (sar -r)

sar -r コマンドを使用し、現在使用されていないメモリーページ数とスワップファイルのディスクブロック数を報告します。

```
$ sar -r
00:00:00 freemem freeswap
01:00:00 2135 401922
```

-r オプションからの出力は次のとおりです。

freemem コマンドによるサンプル収集間隔の間にユーザープロセスに利用できる平均メモリーページ数。ページサイズはシステムに依存します。

freeswap ページスワップに使用可能な 512 バイトのディスクブロック数

例 24 未使用のメモリーをチェックする (sar -r)

次の例は、sar -r コマンドからの出力を示します。

```
$ sar -r
SunOS balmy 5.10 Generic_144500-10 sun4v  ...

00:00:04 freemem freeswap
01:00:00 44717 1715062
02:00:01 44733 1715496
03:00:00 44715 1714746
04:00:00 44751 1715403
05:00:00 44784 1714743
06:00:00 44794 1715186
07:00:00 44793 1715159
08:00:00 44786 1714914
08:20:00 44805 1715576
08:40:01 44797 1715347
09:00:00 44761 1713948
09:20:00 44802 1715478
09:40:00 41770 1682239
10:00:00 35401 1610833
10:20:00 34295 1599141
10:40:00 33943 1598425
11:00:00 30500 1561959
```

Average 43312 1699242

CPU の使用状況をチェックする (sar -u)

sar -u コマンドを使用し、CPU の使用状況の統計情報を表示します。

```
$ sar -u
00:00:00 %usr %sys %wio %idle
01:00:00 0 0 0 100
```

オプションを指定しない sar コマンドは、sar -u コマンドと同じです。プロセッサの状態には、「ビジー」と「アイドル」があります。ビジー状態のときは、プロセッサはユーザーモードまたはシステムモードになっています。アイドル状態のときは、プロセッサは入出力の完了を待っているか、何も処理することがない状態です。

-u オプションからの出力は次のとおりです。

%usr	プロセッサがユーザーモードになっている時間の割合
%sys	プロセッサがシステムモードになっている時間の割合
%wio	プロセッサがアイドル状態で入出力の完了を待っている時間の割合
%idle	プロセッサがアイドル状態で入出力を待っていない時間の割合

一般に、%wio の値が大きい場合は、ディスクの処理速度が低下していることを意味します。

例 25 CPU の使用状況をチェックする (sar -u)

次の例は、sar -u コマンドからの出力を示します。

```
$ sar -u
00:00:04 %usr %sys %wio %idle
01:00:00 0 0 0 100
02:00:01 0 0 0 100
03:00:00 0 0 0 100
04:00:00 0 0 0 100
05:00:00 0 0 0 100
06:00:00 0 0 0 100
07:00:00 0 0 0 100
08:00:00 0 0 0 100
08:20:00 0 0 0 99
08:40:01 0 0 0 99
09:00:00 0 0 0 99
09:20:00 0 0 0 99
09:40:00 4 1 0 95
10:00:00 4 2 0 94
10:20:00 1 1 0 98
```

```

10:40:00    18     3     0    79
11:00:00    25     3     0    72

Average      2     0     0    98

```

システムテーブルのステータスをチェックする (sar -v)

sar -v コマンドを使用して、プロセステーブル、i ノードテーブル、ファイルテーブル、および共有メモリーレポートテーブルのステータスを報告します。

```

$ sar -v
00:00:00  proc-sz   ov  inod-sz   ov  file-sz   ov  lock-sz
01:00:00  43/922    0 2984/4236  0 322/322   0   0/0

```

次のリストに -v オプションからの出力について説明します。

proc-sz	現在カーネル内で使用されているか割り当てられている、プロセスエントリ (proc 構造体) の数。
inod-sz	メモリー内の合計 i ノード数とカーネル内で割り当て済みの最大 i ノード数の比。これは厳密な上限ではありません。ここからオーバーフローすることもあります。
file-sz	開いているシステムファイルテーブルのサイズ。ファイルテーブルには領域が動的に割り当てられるので、sz は 0 として表示されます。
ov	各テーブルのサンプルポイント間で発生しているオーバーフロー。
lock-sz	カーネルで現在使用中または割り当てられている共有メモリーレコードテーブルのエントリの数。共有メモリーレコードテーブルには領域が動的に割り当てられるので、sz は 0 として表示されます。

例 26 システムテーブルのステータスをチェックする (sar -v)

次の例は、sar -v コマンドからの一部省略した出力を示します。この例は、すべてのテーブルに十分なサイズがあり、オーバーフローは発生しないことを示します。これらのテーブルには、いずれも物理メモリーの容量に基づいて領域が動的に割り当てられます。

```

$ sar -v
00:00:04  proc-sz   ov  inod-sz   ov  file-sz   ov  lock-sz
01:00:00  69/8010   0 3476/34703  0   0/0       0   0/0
02:00:01  69/8010   0 3476/34703  0   0/0       0   0/0
03:00:00  69/8010   0 3476/34703  0   0/0       0   0/0
04:00:00  69/8010   0 3494/34703  0   0/0       0   0/0

```

```

05:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
06:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
07:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
08:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
08:20:00 69/8010 0 3494/34703 0 0/0 0 0/0
08:40:01 69/8010 0 3494/34703 0 0/0 0 0/0
09:00:00 69/8010 0 3494/34703 0 0/0 0 0/0
09:20:00 69/8010 0 3494/34703 0 0/0 0 0/0
09:40:00 74/8010 0 3494/34703 0 0/0 0 0/0
10:00:00 75/8010 0 4918/34703 0 0/0 0 0/0
10:20:00 72/8010 0 4918/34703 0 0/0 0 0/0
10:40:00 71/8010 0 5018/34703 0 0/0 0 0/0
11:00:00 77/8010 0 5018/34703 0 0/0 0 0/0

```

スワッピング動作をチェックする (sar -w)

sar -w コマンドを使用すると、スワッピングと切り替え動作が表示されます。

```

$ sar -w
00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00 0.00 0.0 0.00 0.0 22

```

次に、sar -w コマンド出力の対象となる値と説明を示します。

swpin/s	メモリーに転送される 1 秒当たりの軽量プロセス数
bswin/s	スワップイン用に転送される 1 秒当たりのブロック数。/* (float)PGTOBLK(xx->cvmi.pgswpin) / sec_diff */。
swpot/s	メモリーからスワップアウトされる 1 秒当たりの平均プロセス数。この数値が 1 より大きい場合は、メモリーを増やす必要がある
bswot/s	スワップアウト用に転送される 1 秒当たりのブロック数
pswch/s	1 秒当たりのカーネルスレッド切り替え数。

注記 - すべてのプロセスのスワップインには、プロセスの初期化が含まれます。

例 27 スワップ動作をチェックする (sar -w)

次の例は、sar -w コマンドからの出力を示します。

```

$ sar -w
00:00:04 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00 0.00 0.0 0.00 0.0 132
02:00:01 0.00 0.0 0.00 0.0 133
03:00:00 0.00 0.0 0.00 0.0 133
04:00:00 0.00 0.0 0.00 0.0 134
05:00:00 0.00 0.0 0.00 0.0 133
06:00:00 0.00 0.0 0.00 0.0 133

```

```

07:00:00  0.00  0.0  0.00  0.0  132
08:00:00  0.00  0.0  0.00  0.0  131
08:20:00  0.00  0.0  0.00  0.0  133
08:40:01  0.00  0.0  0.00  0.0  132
09:00:00  0.00  0.0  0.00  0.0  132
09:20:00  0.00  0.0  0.00  0.0  132
09:40:00  0.00  0.0  0.00  0.0  335
10:00:00  0.00  0.0  0.00  0.0  601
10:20:00  0.00  0.0  0.00  0.0  353
10:40:00  0.00  0.0  0.00  0.0  747
11:00:00  0.00  0.0  0.00  0.0  804

Average  0.00  0.0  0.00  0.0  198

```

端末動作をチェックする (sar -y)

sar -y コマンドを使用して、端末デバイスの動作をモニターします。

```

$ sar -y
00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00      0      0      0      0      0      0

```

大量の端末入出力がある場合は、このレポートを使用して不良な回線がないかどうかを判別できます。次に、記録される動作を示します。

rawch/s	1 秒当たりの入力文字数 (raw 待ち行列)
canch/s	標準待ち行列で処理される 1 秒当たりの入力文字数
outch/s	1 秒当たりの出力文字数 (出力待ち行列)
rcvin/s	1 秒当たりの受信側ハードウェア割り込み数
xmtin/s	1 秒当たりの送信側ハードウェア割り込み数
mdmin/s	1 秒当たりのモデム割り込み数

1 秒当たりのモデム割り込み数 (mdmin/s) は、0 に近い値になります。また、1 秒当たりの送受信側割り込み数 (xmtin/s と rcvin/s) は、それぞれ着信または発信文字数以下になります。そうでない場合は、不良回線がないかどうかをチェックしてください。

例 28 端末動作をチェックする (sar -y)

次の例は、sar -y コマンドからの出力を示します。

```

$ sar -y
00:00:04 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00      0      0      0      0      0      0
02:00:01      0      0      0      0      0      0
03:00:00      0      0      0      0      0      0

```

04:00:00	0	0	0	0	0	0
05:00:00	0	0	0	0	0	0
06:00:00	0	0	0	0	0	0
07:00:00	0	0	0	0	0	0
08:00:00	0	0	0	0	0	0
08:20:00	0	0	0	0	0	0
08:40:01	0	0	0	0	0	0
09:00:00	0	0	0	0	0	0
09:20:00	0	0	0	0	0	0
09:40:00	0	0	1	0	0	0
10:00:00	0	0	37	0	0	0
10:20:00	0	0	0	0	0	0
10:40:00	0	0	3	0	0	0
11:00:00	0	0	3	0	0	0
Average	0	0	1	0	0	0

システム全体のパフォーマンスをチェックする (sar -A)

sar -A コマンドを使用して、すべてのオプションからの統計情報を表示し、システム全体のパフォーマンスを表示します。

このコマンドを使用すると、全体像を把握できます。複数のタイムセグメントからのデータが表示される場合は、レポートに平均値が含まれます。

システム動作データの自動収集 (sar)

システム動作データを自動的に収集するには、sadc、sa1、および sa2 の 3 つのコマンドを使用します。

sadc データ収集ユーティリティは、システム動作に関するデータを定期的に収集し、24 時間ごとに 1 つのファイルに 2 進形式で保存します。sadc コマンドを定期的に (通常は 1 時間ごとに) 実行したり、システムがマルチユーザーモードでブートするときにも実行するように設定できます。データファイルは、/var/adm/sa ディレクトリに格納されます。各ファイルには sadd という名前が与えられ、この場合、dd は現在の日付です。このコマンドの書式は次のとおりです。

```
/usr/lib/sa/sadc [t n] [ofile]
```

このコマンドは、*t* 秒 (5 秒より長くする必要がある) 間隔でサンプルデータを *n* 回収集します。このコマンドは次に、2 進形式の *ofile* ファイルまたは標準出力に書き込みます。

ブート時の sadc コマンドの実行

カウンタが 0 にリセットされたときから統計情報を記録するには、sadc コマンドをシステムのブート時に実行するようにしてください。sadc コマンドをブート時に確実に

実行するために、`svcadm enable system/sar:default` コマンドで日次データファイルにレコードを書き込みます。

コマンドエントリの書式は次のとおりです。

```
/usr/bin/su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`"
```

sa1 スクリプトを使用した `sadc` コマンドの定期的な実行

定期的にレコードを生成するには、`sadc` コマンドを定期的に行う必要があります。そのためには、`/var/spool/cron/crontabs/sys` ファイルの次の行をコメント解除するのがもっとも簡単な方法です。

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

デフォルトの `sys` の `crontab` エントリによって、次のように動作します。

- 最初の2つの `crontab` エントリによって、月曜から金曜までの午前8時から午後5時までは20分ごとに、それ以外では1時間ごとに、レコードが `/var/adm/sa/sadd` ファイルに書き込まれます。
- 3番目のエントリは、月曜から金曜までは1時間ごとに、レコードを `/var/adm/sa/saradd` ファイルに書き込み、すべての `sar` オプションが含まれます。

これらのデフォルトは、必要に応じて変更できます。

sa2 スクリプトを使用したレポートの生成

もう1つのシェルスクリプト `sa2` は、2進データファイルでないレポートを生成します。`sa2` コマンドは `sar` コマンドを呼び出して、レポートファイルに ASCII 出力を書き込みます。

データの自動収集を設定する (`sar`)

`sar` コマンドを使用すると、システム動作データそのものを収集するか、`sadc` コマンドで作成された日次動作ファイルに収集された情報を報告できます。

`sar` コマンドの書式は次のとおりです。

```
sar [-aAbcdgkmpqruvw] [-o file] t [n]
sar [-aAbcdgkmpqruvw] [-s time] [-e time] [-i sec] [-f file]
```

最初の書式は、オペレーティングシステム内の累積動作カウンタから t 秒間隔で n 回データを収集します。 t は、5秒以上の値にします。それ以外の値にすると、コマンド

そのものがサンプルに影響を与えることがあります。また、サンプルの収集間隔を指定する必要があります。指定しないと、このコマンドは第2の書式に従って動作します。*n* のデフォルト値は1です。

次の例では、2番目の書式を使用して、10秒間隔で2つのサンプルが収集されます。`-o` オプションを指定すると、サンプルは2進形式で保存されます。

```
$ sar -u 10 2
```

2番目の書式を指定した `sar` コマンドは、サンプリング間隔または指定されたサンプルの数なしで、前に記録されたファイルからデータを抽出します。このファイルは、`-f` オプションで指定したファイル、またはデフォルトでは最新日付分の標準日次動作ファイル `/var/adm/sa/sadd` です。

`-s` および `-e` オプションでは、レポートの開始時間と終了時間を定義します。開始時間と終了時間の書式は `hh[:mm[:ss]]` です (この場合、*hh*、*mm*、*ss* はそれぞれ時間、分、秒を表します)。

`-i` オプションでは、レコード選択の間隔を秒単位で指定します。`-i` オプションを指定しなければ、日次動作ファイル内で見つかったすべての間隔がレポートされます。

`sar` オプションおよびそのアクションは次のとおりです。

注記 - オプションを使用しなければ、`sar` コマンドを `-u` オプションを指定して呼び出すのと同じです。

<code>-a</code>	ファイルアクセス操作をチェックする
<code>-b</code>	バッファ動作をチェックする
<code>-c</code>	システムコールをチェックする
<code>-d</code>	各ブロックデバイスの動作をチェックする
<code>-g</code>	ページアウトとメモリの解放をチェックする
<code>-k</code>	カーネルメモリの割り当てをチェックする
<code>-m</code>	プロセス間通信をチェックする
<code>-nv</code>	システムテーブルのステータスをチェックする
<code>-p</code>	スワップとディスパッチ動作をチェックする
<code>-q</code>	待ち行列動作をチェックする
<code>-r</code>	未使用メモリーをチェックする

-u	CPU の使用率をチェックする
-w	ボリュームのスワッピングと切り替えをチェックする
-y	端末動作をチェックする
-A	システム全体のパフォーマンスをレポートする (すべてのオプションを入力した場合と同じです)

▼ 自動データ収集を設定する方法

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. `svcadm enable system/sar:default` コマンドを実行します。

このバージョンの `sadc` コマンドは、カウンタが 0 にリセットされる時間 (ブート時) を示す特殊なレコードを書き込みます。

3. `crontab` ファイル `/var/spool/cron/crontabs/sys` を編集します。

注記 - `crontab` ファイルは直接編集しないでください。既存の `crontab` ファイルを変更するときは、代わりに `crontab -e` コマンドを使用してください。

```
# crontab -e sys
```

4. 次の行のコメントを解除します。

```
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

詳細は、[crontab\(1\)](#) のマニュアルページを参照してください。

システムタスクのスケジュール設定

この章では、Oracle Solaris サービス管理機能 (SMF)、`crontab` コマンド、および `at` コマンドを使用して、ルーティングシステムタスクや 1 回かぎりのタスクをスケジュールする方法について説明します。

内容は次のとおりです。

- 77 ページの「スケジュールされているシステムタスクの概要」
- 81 ページの「SMF を使用した繰り返されるシステムタスクのスケジューリング」
- 83 ページの「`crontab` を使用した繰り返されるシステムタスクのスケジューリング」
- 93 ページの「`at` コマンドを使用した 1 つのシステムタスクのスケジューリング」

スケジュールされているシステムタスクの概要

自動的に実行するシステムタスクを多数設定できます。これらのタスクには、定期的に発生するものと特定の期間のみ発生するものがあります。夜間や週末などのピーク時間外に 1 回だけ実行する必要があるタスクもあります。

このセクションには、Oracle Solaris サービス管理機能 (SMF)、`crontab` コマンド、または `at` コマンドを使用したシステムタスクのスケジュールの概要情報が記載されています。それぞれに、使用するべき主な機能があります。通常、SMF は、リソースモニタリングのためのより単純なスケジューリングメカニズムを提供しますが、`cron` のようなスケジューリング機能を備えています。SMF を使用して、定期的なサービスとスケジュールされているサービスの両方を設定できます。SMF の定期的なサービスは、指定された時間の枠または「間隔」内にルーチンタスクを実行します。スケジュールされている SMF サービスは常に、割り当てられた開始時間に従って実行されます。繰り返されるジョブまたはルーチンジョブを (特定の開始時間に) 実行するには、`crontab` コマンドを使用します。単一のジョブをスケジュールするか、タスクを 1 回 (特定の時間に) 実行するには、`at` コマンドを使用します。

次の表は、SMF、`crontab` コマンド、`at` コマンド、およびこれらの関数のファイルまたはプロパティをまとめたものです。

表 6 タスクを自動的に実行するためのツール

ツール	スケジューリングの対象	使用制御用ファイルまたはプロパティ
定期的な SMF サービス	定期的な間隔での複数のシステムタスク (前の起動の時間に相対的)	<code>solaris.smf.modify</code> <code>solaris.smf.manage</code>
スケジュールされている SMF サービス	特定の時間に実行する複数のシステムタスク	<code>scheduled/schedule</code> は、実行する頻度を設定します。ほかの <code>scheduled</code> プロパティは、実行する時間を指定します。
<code>crontab</code> コマンド	一定間隔で (ただし特定の時間に) 実行する複数のシステムタスク	<code>/var/spool/cron/crontabs</code>
<code>at</code> コマンド	特定の時間に実行する単一のシステムタスク	<code>/var/spool/cron/atjobs</code>

SMF を使用した定期的なタスクまたはスケジュールされているタスクのスケジューリング

Solaris サービス管理機能 (SMF) を使用して、実行中のアプリケーションまたはサービスを管理できます。サービスは、サービスとインスタンスオブジェクトおよびその構成設定によって SMF フレームワークで表されます。ローカルの Oracle Solaris インスタンスの構成は `localhost` スコープと呼ばれ、これが現在サポートされている唯一のスコープです。詳細は、[smf\(5\)](#) のマニュアルページを参照してください。

システムによるルーチン保守タスクを一定間隔で実行するために役立つ SMF サービスは、*periodic* サービスまたは *scheduled* サービスと呼ばれます。スケジュールされているサービスは、定期的なサービスのタイプの 1 つで、特定の時間に発生します。場合によっては実行されるタスクや、特定のスケジュール (ピーク時間外など) で実行されるタスクには、スケジュールされているサービスを使用します。スケジュールされているサービスの詳細は、『[Oracle Solaris 11.3 でのシステムサービスの開発](#)』の第 4 章、「[特定のスケジュールで実行するサービスの作成](#)」を参照してください。

また定期的なサービスでは、最後の実行に相対的な時間に起動メソッドを開始します。これは、より頻繁に、またはより定期的に発生する保守タスクに使用されません。SMF では、定期的なサービスは、委任されたりリスタータ `svc:/system/svc/periodic-restarter` によって管理されます。このリスタータは、管理するインスタンスの開始メソッドのみを実行します。つまり、スケジュールされているタスクは、そのインスタンスがオンラインになっている期間の指定された間隔にのみ開始されます。定期的なサービスの詳細は、『[Oracle Solaris 11.3 でのシステムサービスの開発](#)』の第 3 章、「[定期的に行うサービスの作成](#)」を参照してください。

SMF を使用してタスクをスケジュールする利点は、次のとおりです。

- 失敗したサービスは、依存関係の順序で自動的に再起動されます (一方、cron は通常それ自体を再起動しません)
- サービスはオペレーティングシステムと適切に統合され、依存関係で簡単に制御できます
- サービスの問題をデバッグして、スケジュールされているサービスがなぜ、どのように失敗したかを詳細に示したレポートを作成します
- (必要な) IPS パッケージソフトウェアが実行されている場合にのみタスクが実行されるようにします
- スケジュールされているタスクを削除するための追加の手順は不要です (IPS パッケージで自動的にアンインストールされます)
- プロパティを変更してサービスを管理できる (cron では不可能) root 以外のユーザーにタスクを委任します
- 異なるタイムゾーンと設定で複数のユーザーを管理します

注記 - root アクセス権がないユーザーは、スケジュールされている独自のサービスを SMF で作成できません。そのため、cron または at が、特権が不足しているユーザーの唯一のオプションです。

SMF タスクをスケジュールする手順については、[81 ページの「SMF を使用した線り返されるシステムタスクのスケジューリング」](#)を参照してください。

crontab を使用したルーチンシステムタスクのスケジューリング

cron は、指定された日付と時間にコマンドを実行するプロセスを実行します。SMF とは異なり、これは、独自のプロセス初期化フェーズ中または 2 つのコマンドの実行時に、crontab または at コマンドファイルのみを調べます。定期的にスケジュールされた間隔では新しいファイルや変更されたファイルの有無をチェックしません。

crontab ファイルにある指示に従って、定期的にスケジュールされたコマンドを cron に指定できます。ユーザーは、crontab コマンドを使用して独自のシステムタスクを cron で送信できます。root アクセスは SMF 用であるため、必要ありません。また、crontab コマンドを使用すると、複数回実行する必要があるタスクをスケジュールできますが、at コマンドでは 1 回かぎりの実行だけが可能です。cron が終了することはないため、一度だけ実行してください。

crontab ジョブをスケジュールする手順については、[86 ページの「crontab ファイルを作成または編集する方法」](#)を参照してください。

at を使用した 1 つのシステムタスクのスケジューリング

at コマンドを使用すると、1 回かぎりのタスクまたはまれにしか発生しないタスクを特定の時間に実行するようにスケジュールできます。ジョブは 1 つのコマンドやスクリプトで構成されます。

crontab と同様に、at コマンドでは、システムタスクの自動実行をスケジュールできます。ただし、crontab ファイルとは異なり、at ファイルはタスクを 1 回だけ実行します。その後はディレクトリから削除されます。したがって、at コマンドが役立つのは、単純なコマンドまたはスクリプトを実行して、別ファイルに書き出した出力をあとから調べるような場合です。

at ジョブの実行を指定するには、コマンドを入力してから、at コマンド構文に従ってオプションで実行時間を指定してください。at ジョブの実行の詳細は、[93 ページの「at ジョブファイルの送信」](#)を参照してください。

at コマンドは、実行したコマンドまたはスクリプトを、現在の環境変数のコピーとともに /var/spool/cron/atjobs ディレクトリに格納します。at ジョブのファイル名は、at 待ち行列の場所を指定する長い数値と、.a 拡張子から構成されます (例: 793962000.a)。

cron デーモンは、起動時に at ジョブをチェックし、新しく実行されるジョブを待機します。cron デーモンが at ジョブを実行すると、atjobs ディレクトリから at ジョブのファイルが削除されます。詳細は、[at\(1\)](#) のマニュアルページを参照してください。

at ジョブをスケジュールする手順については、[94 ページの「at ジョブを作成する方法」](#)を参照してください。

繰り返されるシステムタスクの例

毎日、毎週、または毎月実行するルーチンシステム管理タスクをスケジュールできます。タスクの要件または割り当てられているアクセス制御権利に応じて、前述のスケジューリングツールの 1 つを Solaris で使用できます。

毎日のシステム管理タスクには、次のようなものがあります。

- 作成後、数日以上経過したファイルを一時ディレクトリから削除する
- アカウンティングサマリーコマンドを実行する
- df コマンドおよび ps コマンドを使用してシステムのスナップショットを取る
- 日常のセキュリティーモニタリングを実行する

- システムのバックアップを実行する

毎週のシステム管理タスクには、次のようなものがあります。

- `man -k` コマンドで処理する `catman` データベースを再構築する
- `fsck -n` コマンドを実行してディスク問題があれば表示する

毎月のシステム管理タスクには、次のようなものがあります。

- 指定月に使用されなかったファイルをリストする
- 月次アカウントングレポートを生成する

さらに、リマインダの送信やバックアップファイルの削除など、ほかのルーチンシステムタスクをスケジュールできます。

SMF を使用した繰り返されるシステムタスクのスケジューリング

次のセクションには、SMF サービスインスタンスを作成、変更、および表示する方法に関する情報が記載されています。また、SMF アクセス制御についても説明します。

- [81 ページの「SMF のスケジューリング管理方法」](#)
- [82 ページの「SMF のメソッドおよび時間値のスケジューリング」](#)
- [81 ページの「SMF のスケジューリング管理方法」](#)

SMF のスケジューリング管理方法

スケジュールされている各 SMF サービスは、リスタータによって管理されます。マスターリスタータ `svc.startd` は、サービスインスタンスの完全なセットおよびその依存関係の状態を管理します。マスターリスタータは、自身のサービスに代って各種処理を行うほか、特定アプリケーションクラス向けの特定実行環境を提供できる委任リスタータを制御します。たとえば、`inetd` は、初期環境でサービスインスタンスを提供する委任されたリスタータです。`inetd` に委任されたそれぞれのサービスインスタンスがオンライン状態になっています。ある特定のインスタンスのデーモンが実行されていなくても、そのインスタンスを実行することは可能です。インスタンスがオンライン状態に移行したときに依存関係が満たされていると、`svc.startd` は、重複している可能性があるほかのインスタンスの起動メソッドを呼び出します。すべての SMF リスタータの構成設定を表示するには、[smf_restarter\(5\)](#) のマニュアルページを参照してください。

各サービスまたはサービスインスタンスは、サービスの起動、停止、およびリフレッシュを行う一連のメソッドを定義する必要があります。`svc.startd` および類似した

リスタータのメソッド表記に関する詳細は、[smf_method\(5\)](#) のマニュアルページを参照してください。レガシー構成情報のリポジトリへの取得などの管理メソッドについては、[svccfg\(1M\)](#) のマニュアルページで説明されています。

SMF のメソッドおよび時間値のスケジューリング

`scheduled_method` 要素によって委任される SMF のスケジュールされているサービスインスタンスには、特定の時間が必要です。`scheduled_method` 要素は、スケジュールされているサービスのメソッド情報とスケジューリング情報の両方を指定します。

表 7 SMF のスケジューリングに指定できる数値

時間フィールド	数値
分	0-59
時	0-23
日	1-31
月	1-12
曜日	0 - 6 (0 は日曜日)

制約および `scheduled_method` 要素の指定方法の詳細は、『[Oracle Solaris 11.3 でのシステムサービスの開発](#)』の「[scheduled_method 要素の指定](#)」を参照してください。

SMF マニフェストの例

例 29 スケジュールされている SMF サービスの作成

次の例は、毎週日曜日の午前 1 時 00 分に自動的に実行するようにスケジュールされている SMF サービスインスタンスを作成する方法を示しています。

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle
  SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='site/sample-periodic-svc'>
  <service type='service' version='1' name='site/sample-periodic-svc'>

    <instance name='default' enabled='false'>

      <scheduled_method
        schedule='month'
        day='0'
        hour='1'
        minute='0'
        exec='/usr/bin/scheduled_service_method'
      >
    </instance>
  </service>
</service_bundle>
```

```

        timeout_seconds='0'>
        <method_context>
        <method_credential user='root' group='root' />
        </method_context>
    </scheduled_method>

</instance>
</service>
</service_bundle>

```

定期的なサービスの作成方法のステップごとの詳細は、『Oracle Solaris 11.3 でのシステムサービスの開発』の第3章、「定期的に実行するサービスの作成」を参照してください。

crontab を使用した繰り返されるシステムタスクのスケジューリング

次のセクションでは、crontab コマンドを使用してルーチンシステムタスクをスケジューリングする方法について説明します。crontab ファイルの作成、編集、表示、および削除を行う方法に関する情報が次のセクションに記載されています。

- 86 ページの「crontab ファイルを作成または編集する方法」
- 87 ページの「crontab ファイルが存在するかどうかを確認する」
- 88 ページの「crontab ファイルを表示する」
- 89 ページの「crontab ファイルを削除する方法」
- 91 ページの「crontab コマンドの使用を拒否する方法」
- 91 ページの「crontab コマンドの使用を特定のユーザーに限定する方法」

crontab ファイルの内容

cron デーモンは、各 crontab ファイル内にあるコマンドに従ってシステムタスクをスケジューリングします。crontab ファイルには、それぞれ一定間隔で実行されるコマンドが1行に1つずつ入っています。各行の先頭は cron デーモンが各コマンドを実行する日時情報です。

たとえば、Oracle Solaris ソフトウェアのインストール中に、root という名前の crontab ファイルが提供されます。このファイルの内容には、次のコマンド行が含まれています。

```

10 3 * * * /usr/sbin/logadm          (1)
15 3 * * 0 /usr/lib/fs/nfs/nfsfind    (2)
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1    (3)
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean  (4)

```

これらの各コマンド行の出力は次のようになります。

- 最初の行は、毎日午前 3 時 10 分に `logadm` コマンドを実行します。
- 2 行目は、毎週日曜日の午前 3 時 15 分に `nfsfind` スクリプトを実行します。
- 3 行目は、毎日午前 2 時 10 分に、サマータイムをチェック (して必要に応じて修正) するスクリプトを実行します。
RTC タイムゾーンも `/etc/rtc_config` ファイルもない場合、このエントリは何もしません。

x86 のみ - `/usr/sbin/rtc` スクリプトは、x86 ベースのシステムでのみ実行できません。

- 4 行目は、毎日午前 3 時 30 分に Generic Security Service テーブル `/etc/gss/gsscred_db` の重複エントリをチェック (重複エントリがある場合は削除) します。

`crontab` ファイル内のコマンド行の構文の詳細は、[85 ページの「crontab ファイルエントリの構文」](#) を参照してください。

`crontab` ファイルは `/var/spool/cron/crontabs` ディレクトリに保存されます。Oracle Solaris ソフトウェアのインストール時には、`root` 以外にもいくつかの `crontab` ファイルが提供されます。

<code>adm</code>	アカウントティング
<code>root</code>	一般的なシステム機能とファイルシステムの整理
<code>sys</code>	パフォーマンスデータの収集
<code>uucp</code>	一般的な <code>uucp</code> の整理

デフォルトの `crontab` ファイルに加えて、`crontab` ファイルを作成して、自身のシステムタスクをスケジュールできます。カスタムの `crontab` ファイルは、作成したユーザーのアカウントに基づいて、`bob`、`mary`、`smith`、`jones` などのように命名されます。

`root` またはほかのユーザーの `crontab` ファイルを使用するには、スーパーユーザーの特権が必要です。

cron デーモンのスケジューリング管理

`cron` デーモンは、`crontab` コマンドの自動スケジューリングを管理します。`cron` デーモンは、`/var/spool/cron/crontab` ディレクトリに `crontab` ファイルがあるかどうかをチェックします。

cron デーモンは、起動時に次のタスクを実行します。

- 新しい crontab ファイルがないかを確認する
- ファイル内のリストから実行時間を読み取る
- 正しい時間にコマンドを実行する
- 更新された crontab ファイルに関する crontab コマンドからの通知を待機する

ほとんど同様に、cron デーモンは at ファイルのスケジューリングを制御します。これらのファイルは /var/spool/cron/atjobs ディレクトリに格納されています。cron デーモンは、実行された at ジョブに関する crontab コマンドからの通知も待機します。

crontab ファイルエントリの構文

crontab ファイルには、スペースで区切られた各コマンド行の最初の 5 つのフィールドで指定された時間に自動的に実行されるコマンドが 1 行に 1 つずつ入っています。

表 8 指定できる crontab 時間フィールドの値

時間フィールド	値
分	0-59
時	0-23
日	1-31
月	1-12
曜日	0 - 6 (0 は日曜日)

次に、crontab 時間フィールドで特殊文字を使用する際のガイドラインを示します。

- 各フィールドはスペースで区切る
- 複数の値の間はコンマで区切る
- 値の範囲はハイフンを使用して指定する
- 取り得るすべての値を含むには、ワイルドカードとしてアスタリスクを使用する
- コメントまたは空白行を示すには、行の先頭にコメント記号 (#) を使用する

たとえば、次の crontab コマンドエントリは、毎月 1 日と 15 日の午後 4 時に、ユーザーのコンソールウィンドウに注意を促すメッセージを表示します。

```
0 16 1,15 * * echo Timesheets Due > /dev/console
```

crontab ファイル内の各コマンドは、非常に長い行であっても 1 行で構成する必要があります。crontab ファイルは余分なキャリッジリターンを認識しません。crontab

エントリとコマンドオプションの詳細は、[crontab\(1\)](#) のマニュアルページを参照してください。

crontab ファイルの作成と編集

crontab ファイルを作成するもっとも簡単な方法は、`crontab -e` コマンドを使用することです。このコマンドは、`EDITOR` 環境変数でシステム環境に定義されているテキストエディタを起動します。この環境変数が設定されていない場合は、`crontab` コマンドはデフォルトのエディタ `ed` を使用します。

次の例は、エディタが定義されているかどうかを判別する方法を示しており、`vi` をデフォルトエディタとして設定します。

```
$ which SEDITOR
$
$ EDITOR=vi
$ export EDITOR
```

crontab ファイルを作成すると、自動的に `/var/spool/cron/crontabs` ディレクトリ内に格納され、作成者のユーザー名で命名されます。`root` 特権があれば、別のユーザーや `root` の crontab ファイルを作成または編集できます。

▼ crontab ファイルを作成または編集する方法

始める前に 別のユーザーに属する crontab ファイルを作成または編集する場合は、`root` 役割になる必要があります。『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

自分の crontab ファイルを編集する場合は、`root` 役割になる必要はありません。

1. 新しい crontab ファイルを作成するか、既存の crontab ファイルを編集します。

```
# crontab -e [username]
```

`username` は、crontab ファイルを作成または編集するユーザーのアカウント名を指定します。自分の crontab ファイルを作成するにはスーパーユーザー特権は必要ありませんが、`root` または別のユーザーの crontab ファイルを作成したり編集したりするには、スーパーユーザー特権が必要です。



注意 - 誤ってオプションを指定しないで `crontab` コマンドを入力した場合は、変更を保存せずに終了できるようにする、エディタの中断文字を入力してください。この場合に変更を保存してファイルを終了すると、既存の crontab ファイルが空のファイルで上書きされます。

2. コマンド行を crontab ファイルに追加します。

85 ページの「[crontab ファイルエントリの構文](#)」に記載されている構文に従ってください。crontab ファイルは、`/var/spool/cron/crontabs` ディレクトリに保存されます。

3. crontab ファイルの変更箇所を確認します。

```
# crontab -l [username]
```

例 30 crontab ファイルを作成する

次の例は、他のユーザーのための crontab ファイルをどのように作成するかを示します。

```
# crontab -e mary
```

新しい crontab ファイルに追加された次のコマンドエントリは、毎週日曜日の午前 1 時 00 分にすべてのログファイルを Mary のホームディレクトリから自動的に削除します。このコマンドエントリは出力先を変更しないので、出力先変更文字がコマンド行の `*.log` のあとに追加されます。このためコマンドが正しく実行されます。

```
# This command helps clean up user accounts.
1 0 * * 0 rm /home/mary/*.log > /dev/null 2>&1
```

crontab ファイルの表示と確認

`crontab -l` コマンドを使用して、crontab ファイルの内容を表示および確認できます。

crontab ファイルが存在するかどうかを確認する

ユーザーの crontab ファイルが存在することを確認するには、`/var/spool/cron/crontabs` ディレクトリ内の `ls -l` コマンドを使用します。たとえば、次のサンプル出力は、システムにさまざまなユーザーの crontab ファイルが存在していることを示しています。

```
$ ls -l /var/spool/cron/crontabs
drwxr-xr-x  2 root    sys          12 Nov 26 16:55 ./
drwxr-xr-x  4 root    sys          4 Apr 28 2012 ../
-rw-r----- 1 root    sys         190 Jun 28 2011 adm
-rw-r----- 1 root    staff        0 Nov 13 2012 mary
-rw-r----- 1 root    un           437 Oct  8 2012 johndoe
-r----- 1 root    root         453 Apr 28 2012 lp
-rw-r----- 1 root    sparccad    63 Jul 17 10:39 mary2
-rw-r----- 1 root    sparccad    387 Oct 14 15:15 johndoe2
-rw-r----- 1 root    other       2467 Nov 26 16:55 root
-rw-r----- 1 root    sys         308 Jun 28 2011 sys
-rw-r----- 1 root    siete       163 Nov 20 10:40 mary3
-r----- 1 root    sys         404 Jan 24 2013 uucp
```

crontab ファイルを表示する

`crontab -l` コマンドは、`cat` コマンドがほかのタイプのファイルの内容を表示する方法とほとんど同じ方法で `crontab` ファイルの内容を表示します。このコマンドを使用するために、(`crontab` ファイルが入っている) `/var/spool/cron/crontabs` ディレクトリに変更する必要はありません。

デフォルトでは、`crontab -l` コマンドは独自の `crontab` ファイルを表示します。ほかのユーザーに属する `crontab` ファイルを表示するには、`root` 役割になる必要があります。

You can use the `crontab` command as follows:

```
# crontab -l [username]
```

`username` は、`crontab` ファイルを表示するユーザーのアカウント名を指定します。別のユーザーの `crontab` ファイルを表示するには、`root` 特権が必要です。



注意 - 誤ってオプションを指定しないで `crontab` コマンドを入力した場合は、使用しているエディタの中断文字を入力して、変更を保存せずに終了します。この場合に変更を保存してファイルを終了すると、既存の `crontab` ファイルが空のファイルで上書きされます。

例 31 crontab ファイルを表示する

この例は、`crontab -l` コマンドを使用して、デフォルトの `crontab` ファイルの内容を表示する方法を示しています。

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
```

例 32 デフォルトの root の crontab ファイルを表示する

次の例は、デフォルトの `root` の `crontab` ファイルを表示する方法を示します。

```
$ su
Password:

# crontab -l
#ident "@(#)root      1.19   98/07/06 SMI"   /* SVr4.0 1.1.3.1   */
#
# The root crontab should be used to perform accounting data collection.
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

例 33 他ユーザーの crontab ファイルを表示する

次の例は、他ユーザーの crontab ファイルを表示する方法を示します。

```
$ su
Password:
# crontab -l jones
13 13 * * * cp /home/jones/work_files /usr/backup/. > /dev/null 2>&1
```

crontab ファイルの削除

デフォルトでは、rm コマンドを使用して誤って crontab ファイルを削除できないように、crontab ファイルの保護が設定されます。crontab ファイルを削除する場合は、rm コマンドではなく crontab -r コマンドを使用してください。

デフォルトでは、crontab -r コマンドは自分自身の crontab ファイルを削除します。

このコマンドを使用するために、/var/spool/cron/crontabs ディレクトリ (crontab ファイルが入っている) に変更する必要はありません。

▼ crontab ファイルを削除する方法

始める前に root またはほかのユーザーに属する crontab ファイルを削除するには、root 役割になります。役割には、認証と特権コマンドが含まれます。『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

自分の crontab ファイルを削除する場合は、root 役割になる必要はありません。

1. 次のように入力して、crontab ファイルを削除します。

```
# crontab -r [username]
```

username は、crontab ファイルを削除するユーザーのアカウント名を指定します。別のユーザーの crontab ファイルを削除するには、root 役割になります。



注意 - 誤ってオプションを指定しないで crontab コマンドを入力した場合は、使用しているエディタの中断文字を入力して、変更を保存せずに終了します。この場合に変更を保存してファイルを終了すると、既存の crontab ファイルが空のファイルで上書きされます。

2. crontab ファイルが削除されたことを確認します。

```
# ls /var/spool/cron/crontabs
```

例 34 crontab ファイルを削除する

次の例では、ユーザー smith が `crontab -r` コマンドを使用して自分の crontab ファイルを削除します。

```
$ ls /var/spool/cron/crontabs
adm   jones   root    smith   sys     uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm   jones  root    sys     uucp
```

crontab コマンドの使用制御

`/etc/cron.d` ディレクトリ内の `cron.deny` と `cron.allow` の 2 つのファイルによって、`crontab` コマンドの使用を制御できます。これらのファイルによって、指定したユーザーだけが、それぞれ自分の crontab ファイルの作成、編集、表示、または削除などの crontab コマンドのタスクを実行できるようにします。

`cron.deny` ファイルおよび `cron.allow` ファイルは、それぞれ 1 行に 1 ユーザー名が入ったリストからなります。

これらの使用制御用ファイルは、次のように連携して機能を果たします。

- `cron.allow` が存在する場合、このファイルに一覧表示されているユーザーのみが、`crontab` ファイルを作成、編集、表示、または削除できます。
- `cron.allow` が存在しない場合は、`cron.deny` にリストされているユーザーを除くすべてのユーザーが `crontab` ファイルを送信できます。
- `cron.allow` も `cron.deny` も存在しない場合、`crontab` コマンドを実行するには、root 役割になる必要があります。
- `cron.deny` および `cron.allow` ファイルを編集または作成するには、root 役割になる必要があります。

Oracle Solaris ソフトウェアのインストール時に作成される `cron.deny` ファイルには、次のユーザー名が含まれます。

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

デフォルトの `cron.deny` ファイル内のユーザー名は、いずれも `crontab` コマンドを使用できません。このファイルを編集して、`crontab` コマンドの使用を拒否したいユーザーを追加できます。

デフォルトの `cron.allow` ファイルは指定されていないため、デフォルトの `cron.deny` ファイルに一覧表示されているユーザーを除くすべてのユーザーが `crontab` コ

マンドを使用できます。cron.allow ファイルを作成した場合、そのユーザーだけが crontab コマンドを使用できます。

▼ crontab コマンドの使用を拒否する方法

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. /etc/cron.d/cron.deny ファイルを編集し、crontab コマンドの使用が拒否されるユーザーの名前を 1 行ずつ追加します。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

3. /etc/cron.d/cron.deny ファイルに新しいエントリが含まれているか確認します。

```
# cat /etc/cron.d/cron.deny
daemon
bin
nuucp
listen
nobody
noaccess
```

▼ crontab コマンドの使用を特定のユーザーに限定する方法

1. root 役割になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. /etc/cron.d/cron.allow ファイルを作成します。

3. root 役割を cron.allow ファイルに追加します。

このファイルに root を追加しないと、crontab コマンドへの root アクセスが拒否されます。

4. crontab コマンドの使用を許可するユーザーの名前を 1 行ずつ追加します。

```
root
username1
username2
username3
.
.
.
```

例 35 crontab コマンドの使用を特定のユーザーに限定する

次は、ユーザー jones、temp、および visitor に crontab コマンドを使用させない cron.deny ファイルの例です。

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
temp
visitor
```

次の例は、cron.allow ファイルを示しています。ユーザー root、jones、および smith だけが、crontab コマンドを使用できます。

```
$ cat /etc/cron.d/cron.allow
root
jones
smith
```

crontab コマンドの使用制限の確認

特定のユーザーが crontab コマンドを使用できるかどうかを確認するには、そのユーザーのアカウントでログインして crontab -l コマンドを使用します。

```
$ crontab -l
```

ユーザーが crontab コマンドを使用でき、crontab ファイルをすでに作成してある場合、このファイルが表示されます。そのユーザーが crontab コマンドを使用できるが、crontab ファイルがない場合は、次のようなメッセージが表示されます。

```
crontab: can't open your crontab file
```

このユーザーは、cron.allow ファイル (が存在する場合) に含まれているか、cron.deny ファイルに含まれていません。

ユーザーが crontab コマンドを使用できない場合は、上記の crontab ファイルの有無にかかわらず、次のメッセージが表示されます。

```
crontab: you are not authorized to use cron. Sorry.
```

このメッセージは、ユーザーが `cron.allow` ファイル (が存在する場合) に含まれていないか、`cron.deny` ファイルに含まれていることを意味します。

at コマンドを使用した 1 つのシステムタスクのスケジューリング

次のセクションでは、`at` コマンドを使用してルーチンシステムタスクをスケジュールする方法について説明します。`at` を使用してシステムタスクをスケジュールするための例は、次のセクションに記載されています。

- [94 ページの「at ジョブを作成する方法」](#)
- [95 ページの「at 待ち行列を表示する」](#)
- [95 ページの「at ジョブを確認する」](#)
- [95 ページの「at ジョブを表示する」](#)
- [96 ページの「at ジョブを削除する方法」](#)
- [97 ページの「at コマンドの使用を拒否する」](#)

デフォルトでは、ユーザーはそれぞれ自分の `at` ジョブファイルを作成、表示、または削除できます。`root` または別のユーザーに属する `at` ファイルにアクセスするには、`root` 役割になる必要があります。

at ジョブファイルの送信

`at` ジョブの実行を設定すると、ジョブ識別番号と `.a` 拡張子が与えられます。これがジョブのファイル名および待ち行列番号になります。

`at` ジョブファイルの送信の手順は次のとおりです。

1. コマンド実行時間を指定して `at` ユーティリティーを起動します。
2. あとで実行させるコマンドまたはスクリプトを入力します。

注記 - このコマンドまたはスクリプトからの出力が重要な場合は、後で調べることができるように、出力内容を必ずファイルに書き込むようにしてください。

たとえば、次の at ジョブは、7月最終日の深夜0時近くにユーザーアカウント smith から core ファイルを削除します。

```
$ at 11:45pm July 31
at> rm /home/smith/*core*
at> Press Control-d
commands will be executed using /bin/csh
job 933486300.a at Tue Jul 31 23:45:00 2004
```

at ジョブを作成する

次のタスクでは、at ジョブの作成方法について説明します。

▼ at ジョブを作成する方法

1. at ユーティリティを起動して、ジョブを実行する時間を指定します。

```
$ at [-m] time [date]
```

-m	ジョブの完了後に電子メールを送信することを指定する。
time	ジョブをスケジュールしたい時刻を指定する。24 時間制を使用しない場合は、am または pm を追加する。使用できるキーワードは、midnight、noon、now。分単位の値の指定はオプション。
date	月または曜日の英語名の最初の 3 文字以上、またはキーワード today または tomorrow を指定する

2. at プロンプトに、実行したいコマンドまたはスクリプトを 1 行に 1 つずつ入力します。
各行の終わりで Return キーを押すことにより、複数のコマンドを入力できます。
3. Control-D キーを押して at ユーティリティを終了し、at ジョブを保存します。
at ジョブは待ち行列番号を割り当てられ、それがそのジョブのファイル名にもなります。この番号は at ユーティリティの終了時に表示されます。

例 36 at ジョブを作成する

次の例は、ユーザー jones がバックアップファイルを午後 7 時 30 分に削除するために作成した at ジョブを示しています。彼女は、ジョブの完了後に電子メールメッセージを受信するように -m オプションを使用しました。

```
$ at -m 1930
at> rm /home/jones/*.*.backup
at> Press Control-D
```

```
job 897355800.a at Thu Jul 12 19:30:00 2004
```

彼女は、at ジョブが実行されたことを確認するメールメッセージを受け取りました。

```
Your "at" job "rm /home/jones/*.backup"
completed.
```

次の例は、jones で日曜日の午前 4 時 00 分に大規模な at ジョブをスケジュールした方法を示しています。ジョブの出力先は big.file という名前のファイルです。

```
$ at 4 am Saturday
at> sort -r /usr/dict/words > /export/home/jones/big.file
```

at 待ち行列を表示する

at 待ち行列で実行を待っているジョブを確認するには、次に示すように atq コマンドを使用します。

```
$ atq
```

このコマンドは、その使用者が作成した at ジョブに関するステータス情報を表示します。

at ジョブを確認する

at ジョブが作成できたかどうかを確認するには、atq コマンドを使用します。次の例の atq コマンドは、jones の at ジョブが待ち行列に入っていることを確認しています。

```
$ atq
Rank  Execution Date      Owner   Job           Queue  Job Name
1st   Jul 12, 2004 19:30   jones  897355800.a   a      stdin
2nd   Jul 14, 2004 23:45   jones  897543900.a   a      stdin
3rd   Jul 17, 2004 04:00   jones  897732000.a   a      stdin
```

at ジョブを表示する

自分の at ジョブの実行時間に関する情報を表示するには、at -l コマンドを使用します。

```
$ at -l [job-id]
```

ここで、-l job-id は、ステータスを表示する特定のジョブのオプションの ID 番号です。ID がないと、コマンドによってユーザーが実行したすべてのジョブのステータスが表示されます。

例 37 at ジョブを表示する

次の例は、at -l コマンドからのサンプル出力を示しており、このコマンドは、ユーザーが実行を依頼したすべてのジョブのステータスに関する情報を表示します。

```
$ at -l
897543900.a Sat Jul 14 23:45:00 2004
897355800.a Thu Jul 12 19:30:00 2004
897732000.a Tue Jul 17 04:00:00 2004
```

次の例は、at -l コマンドを使用して単一のジョブを指定する場合に表示されるサンプル出力を示しています。

```
$ at -l 897732000.a
897732000.a Tue Jul 17 04:00:00 2004
```

▼ at ジョブを削除する方法

始める前に root または別のユーザーに属する at ジョブを削除するには、root 役割になります。『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

自分の at ジョブを削除する場合は、root 役割になる必要はありません。

1. 次のように入力して、at ジョブが実行される前に待ち行列から削除します。

```
# at -r [job-id]
```

-r job-id オプションで、削除したいジョブの識別番号を指定します。

2. at -l (または atq) コマンドを使用して、at ジョブが削除されていることを確認します。

at -l コマンドは、at 待ち行列に残っているジョブを表示します。識別番号を指定したジョブは、このリストに表示されないはずで

```
$ at -l [job-id]
```

例 38 at ジョブを削除する

次の例では、ユーザーは、7月17日午前4時に実行するようにスケジュールされた at ジョブを削除します。まず、このユーザーは at 待ち行列を表示してそのジョブの識別番号を探します。次に、そのジョブを at 待ち行列から削除します。最後に、at 待ち行列をもう一度表示して上記のジョブが削除されていることを確認します。

```
$ at -l
897543900.a Sat Jul 14 23:45:00 2003
897355800.a Thu Jul 12 19:30:00 2003
897732000.a Tue Jul 17 04:00:00 2003
$ at -r 897732000.a
```

```
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

at コマンドの使用制御

at コマンドの使用を制御して、指定したユーザーのみに at ジョブに関するキュー情報の作成、削除、または表示を許可するファイルを設定できます。at コマンドの使用を制御するファイルは `/etc/cron.d/at.deny` で、ここにはユーザー名が列挙 (1 行に 1 人) されています。このファイルに列挙されているユーザーは、at コマンドを使用できません。

Oracle Solaris ソフトウェアのインストール時に作成される `at.deny` ファイルには、次のユーザー名が含まれます。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

スーパーユーザー特権を使用して、`at.deny` ファイルを編集して、at コマンドの使用を制限するほかのユーザー名を追加します。

at コマンドの使用を拒否する

root で、`/etc/cron.d/at.deny` ファイルを編集して、at コマンドを使用させないようにするユーザー名を 1 行に 1 つずつ追加します。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

例 39 at の使用を拒否する

次の例は、ユーザー `smith` および `jones` が at コマンドを使用できないように編集された `at.deny` ファイルを示しています。

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith
```

at コマンドの使用制御が拒否されていることを確認する

ユーザー名が `/etc/cron.d/at.deny` ファイルに正しく追加されたことを確認するには、ユーザーとしてログインしているときに `at -l` コマンドを使用します。たとえば、ログインしているユーザー `smith` は、`at` コマンドにアクセスできず、次のメッセージが表示されます。

```
# su smith
Password:
# at -l
at: you are not authorized to use at. Sorry.
```

同様に、そのユーザーが `at` ジョブの実行を依頼しようとした場合は、次のメッセージが表示されます。

```
# at 2:30pm
at: you are not authorized to use at. Sorry.
```

このメッセージによって、そのユーザーが `at.deny` ファイルに含まれていることが確認されます。

`at` コマンドを使用できる場合、`at -l` コマンドは何も返しません。

システムコンソール、端末デバイス、および電源サービスの管理

この章では、`ttymon` プログラムとシステム電源サービスを使用してシステムコンソールとローカル接続された端末デバイスを管理する方法について説明します。

この章で扱う内容は、次のとおりです。

- 99 ページの「システムコンソールとローカル接続された端末デバイスの管理」
- 102 ページの「システム電源サービスの管理」

システムコンソールとローカル接続された端末デバイスの管理

システムコンソールは、特殊な属性を持つ端末で、特定の用途に使用されます。たとえば、管理者向けのカーネルメッセージはコンソールに送信され、ほかの端末には送信されません。

端末は、Oracle Solaris とやりとりするための手段です。システムのビットマップグラフィックスディスプレイは、英数字端末と同じではありません。英数字端末はシリアルポートに接続され、テキストのみを表示します。グラフィックスディスプレイは、特別な手順に従って管理する必要はありません。

端末をコンピュータの物理的なモニターとキー配列に関連付けることもできます。グラフィックス端末がほかと異なる点は、コンピュータのグラフィックスカードとモニターに関連付ける必要があることです。したがって、シリアルポートに文字が送信されるのではなく、コンピュータに内蔵されているグラフィックスカードのメモリー上に文字が描画されます。

システムコンソールとローカル接続された端末デバイスを管理する SMF サービス

システムコンソールおよびローカルに接続される端末デバイスは、SMF サービスのインスタンスの `svc:/system/console` として表されます。このサービスはほとんどの

動作を定義し、各インスタンスはサービスから継承された設定に対する優先指定値を持っています。ttymon プログラムは、これらの端末にログインサービスを提供するために使用されます。各端末では、ttymon プログラムの個別のインスタンスを使用します。サービスから ttymon プログラムに渡されるコマンド行引数によってプログラムの動作が制御されます。

システムに付属するサービスインスタンスは次のとおりです。

- `svc:/system/console-login:default`
デフォルトインスタンスは、常に ttymon プログラムによってシステムのハードウェアへのログインが提供されることを表します。
- `svc:/system/console-login:{vt2, vt3, vt4, vt5, vt6}`
システムの仮想コンソールには、追加のサービスインスタンスが提供されていません。仮想コンソールが使用できない場合は、これらのサービスが自動的に無効化されます。詳細は、[vtdaemon\(1M\)](#) のマニュアルページを参照してください。
- `svc:/system/console-login:{terma, termb}`
`svc:/system/console-login:terma` および `svc:/system/console-login:termb` サービスは、便宜を図るために提供されます。これらのサービスは、追加の `/dev/term/a` および `/dev/term/b` ポートに対してログインサービスを設定するのに便利です。これらのサービスはデフォルトで無効化されています。

追加のサービスインスタンスは、`svc:system/console-login` サービスの一部として定義できます。たとえば、サポートする必要がある `/dev/term/f` ポートがある場合、`svc:/system/console-login:termf` をインスタンス化して、適切な方法で構成できます。

▼ 補助端末に対してログインサービスを設定する方法

システム上の `/dev/term/a` または `/dev/term/b` シリアルポートに接続された端末に対しては、定義済みのサービスが提供されます。

1. **root 役割になります。**
『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. **サービスインスタンスを有効にします。**
たとえば、`/dev/term/a` に対するログインサービスを有効にするには:

```
# svcadm enable svc:/system/console-login:terma
```
3. **サービスがオンラインであることを確認します。**

```
# svcs svc:/system/console-login:terma
```

サービスがオンラインであることが出力に表示されるはずですが、サービスが保守モードである場合は、サービスのログファイルで詳細を調べます。

▼ コンソールのボーレート速度を設定する方法

x86 ベースのシステムでのコンソール速度のサポートは、特定のプラットフォームに依存します。

SPARC ベースのシステムには、次のコンソール速度がサポートされています。

- 9600 bps
- 19200 bps
- 38400 bps

1. 管理者になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. eeprom コマンドを使用して、システムタイプに適したボーレート速度を設定します。

```
# eeprom ttya-mode=baud-rate,8,n,1,-
```

たとえば、x86 ベースシステムのコンソールのボーレート速度を 38400 に変更するには、次のように入力します。

```
# eeprom ttya-mode=38400,8,n,1,-
```

3. /etc/ttydefs ファイルのコンソール行を次のように変更します。

```
console baud-rate hupcl opost onlcr:baud-rate::console
```

4. システムのタイプに合わせて、次の追加変更を行います。

これらの変更はプラットフォームに依存することに注意してください。

- **SPARC ベースのシステム:** /etc/driver/drv ディレクトリにあるバージョンの options.conf ファイルでボーレート速度を変更します。例:

ボーレートを 9600 に変更するには:

```
# 9600          :bd:
ttymodes="2502:1805:bd:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

ボーレート速度を 19200 に変更するには。

```
# 19200         :be:
ttymodes="2502:1805:be:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

ボーレート速度を 38400 に変更するには:

```
# 38400          :bf:
ttymodes="2502:1805:bf:8a3b:3:1c:7f:15:4:0:0:11:13:1a:19:12:f:17:16";
```

- **x86 ベースのシステム:** BIOS のシリアル切り替えが有効である場合にコンソール速度を変更します。

システム電源サービスの管理

Oracle Solaris 11 オペレーティングシステムでは、電源管理構成が SMF 構成リポジトリに移動しています。電源関連のコマンド、デーモン、および構成ファイルを組み合わせる代わりに、新しい `poweradm` コマンドを使用してシステムの電源管理プロパティを直接管理します。これらの変更は、広範囲の変更の一部で、Oracle Solaris 11 オペレーティングシステムの電源管理フレームワークが最適化されます。

次の電源管理機能は使用できなくなりました。

- `/etc/power.conf`
- `pmconfig` および `powerd`
- デバイス電源管理

次のプロパティは、電源管理コンポーネントを記述します。

- `administrative-authority` – Oracle Solaris 電源管理の管理制御のソースを定義します。このプロパティは `none`、`platform` (デフォルト値)、または `smf` に設定できます。

`platform` に設定した場合、`time-to-full-capacity` と `time-to-minimum-responsiveness` の値はプラットフォームの電源管理コマンドから取得されます。

`smf` に設定した場合、`time-to-full-capacity` と `time-to-minimum-responsiveness` の値は SMF から取得されます。

`time-to-full-capacity` または `time-to-minimum-responsiveness` を逆の設定でプラットフォームのコマンドまたは SMF サービスプロパティから設定しようとすると、値は無視されます。

`administrative-authority` を `none` に設定すると、Oracle Solaris インスタンス内の電源管理はオフになります。

- `time-to-full-capacity` – システムがアクティブなままで低稼働状態または応答性の低い状態からフル稼働状態に達することが許可される最大時間 (ミリ秒単位) を定義します。最大値には、この境界内の PM 機能のいずれかまたはすべてを使用している時間が含まれます。

`administrative-authority` のデフォルト設定は `platform` に設定されているため、デフォルトでは、この値はプラットフォーム (たとえば `i86pc`) から取得されます。

または、`administrative-authority` が `smf` に設定されている場合、この値は SMF 電源サービスによって提供される定義から取得されます。インストール時は、この値は未定義です。このプロパティを変更する場合は、システムのワークロードやアプリケーションのニーズに応じて適切な値を検討するようにしてください。

- `time-to-minimum-responsiveness` – システムがアクティブな状態に戻ることが許可される時間 (ミリ秒単位) を定義します。このパラメータは、`time-to-full-capacity` 制約を満たすために必要な最小容量を指定します。`administrative-authority` のデフォルト設定は、デフォルトで `platform` に設定されているため、このパラメータ値は `i86pc` などのプラットフォームから取得されます。

または、`administrative-authority` が `smf` に設定されている場合、この値は SMF 電源サービスによって提供される定義から取得されます。インストール時は、この値は未定義です。このプロパティを変更する場合は、システムのワークロードやアプリケーションのニーズに応じて適切な値を使用してください。

中程度の値 (たとえば秒単位) を使用すると、プラットフォーム上のハードウェアコンポーネントやサブシステムをより応答性の低い休止状態にすることができます。より大きい値 (たとえば 30 秒から分単位) を使用すると、RAM への保存停止などの手法を使用してシステム全体の保存停止が可能になります。

- `suspend-enable` – デフォルトでは、Oracle Solaris を実行しているシステムで保存停止操作の試行は許可されません。このプロパティを `true` に設定すると、保存停止操作の試行が許可されます。`administrative-authority` の値はこのプロパティには影響を与えません。
- `platform-disabled` – `platform-disabled` を `true` に設定すると、プラットフォームの電源管理は無効になります。デフォルト値の `false` に設定すると、電源管理は上記のプロパティの値によって制御されます。

電源管理ステータスの簡単なサマリーを表示するには、次のコマンドを発行します。

```
$ /usr/sbin/poweradm show
Power management is enabled with the hardware platform as the authority:
time-to-full-capacity set to 250 microseconds
time-to-minimum-responsiveness set to 0 milliseconds
```

電源管理プロパティを表示するには、次のコマンドを発行します。

```
$ /usr/sbin/poweradm list
active_config/time-to-full-capacity          current=250, platform=250
active_config/time-to-minimum-responsiveness current=0, platform=0
active_control/administrative-authority      current=platform, smf=platform
suspend/suspend-enable                       current=false
platform-disabled                             current=false
```

この出力で、`active_control/administrative-authority` は 2 つの設定がある構成のソースを示します。

- `platform` - プラットフォームに由来する電源管理の構成です。これがデフォルト値です。
- `smf - poweradm` コマンドを使用してほかの電源管理プロパティを設定できます。

出力の `platform-disabled` プロパティは、プラットフォームの電源管理が有効になっていることを示しています。

```
platform-disabled          current=false
```

詳細については、[poweradm\(1M\)](#) のマニュアルページを参照してください。

例 40 電源管理有効化と無効化

以前にシステムの保存停止と復元再開を行うために `/etc/power.conf` ファイルで `S3-support` を有効にしていた場合、類似する `poweradm` の構文は:

```
# poweradm set suspend-enable=true
```

`suspend-enable` プロパティはデフォルトで `false` に設定されます。

電源管理を無効にするには、次の構文を使用します。

```
# poweradm set administrative-authority=none
```

次の SMF 電源管理サービスを無効にしても、電源管理は無効になりません。

```
online      Sep_02   svc:/system/power:default
```

保存停止と復元再開を無効にするには、次の構文を使用します。

```
# poweradm set suspend-enable=false
```

例 41 電源管理パラメータの設定および表示

次の例は、`time-to-full-capacity` を 300 ミリ秒、`time-to-minimum-responsiveness` を 500 ミリ秒に設定する方法を示しています。最後に、新しい値が Oracle Solaris インスタンスに通知されます。

```
# poweradm set time-to-full-capacity=300
# poweradm set time-to-minimum-responsiveness=500
# poweradm set administrative-authority=smf
```

次のコマンドは、現在の `time-to-full-capacity` の値を表示します。

```
# poweradm get time-to-full-capacity
300
```

次のコマンドは、プラットフォームによって設定された `time-to-full-capacity` の値を取得します。

```
# poweradm get -a platform time-to-full-capacity
```

この値が現在の値と同じになるのは、`administrative-authority` が `platform` に設定されている場合だけです。詳細については、上記の `administrative-authority` プロパティの説明を参照してください。

▼ 保守モードの電源サービスから回復する方法

`time-to-full-capacity` と `time-to-minimum-responsiveness` の両方を設定する前に `administrative-authority` を `smf` に設定すると、サービスが保守モードに移行します。このシナリオから回復するには、次のタスクを参照してください。

1. 管理者になります。

『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. `administrative-authority` を `none` に設定します。

```
# poweradm set administrative-authority=none
```

3. `time-to-full-capacity` と `time-to-minimum-responsiveness` の両方を必要な値に設定します。

```
# poweradm set time-to-full-capacity=value  
# poweradm set time-to-minimum-responsiveness=value
```

4. サービスをクリアします。

```
# svcadm clear power
```

5. `administrative-authority` を `smf` に設定します。

```
# poweradm set administrative-authority=smf
```


索引

あ

- アドレス空間マップ
 - 表示, 29
- アプリケーションスレッド, 47, 48
- 一覧表示
 - 実行中のプロセス, 30
 - プロセス, 30

か

- カーネルスレッド
 - 構造体, 27, 47
 - スケジューリング, 27
- 強制終了プロセス, 32
- 共有メモリー
 - プロセスの仮想メモリー, 48
- 繰り返されるシステムタスク
 - SMF インスタンス, 81
- 繰り返されるタスク
 - スケジューリング, 83
- コンソール端末
 - ボーレートの設定, 101

さ

- サービス管理機能
 - システムタスク, 77
- 削除
 - crontab ファイル, 90
- 作成
 - at ジョブ, 94
- 時間
 - CPU 時間が大量に増えているプロセス, 42
 - CPU 使用率, 27
 - CPU の使用, 42

- システムコンソール
 - 管理, 99
 - SMF サービスの使用, 99
- システムタスク, 78, 79
 - 参照 crontab コマンド、at コマンド
 - 参照 smf
 - at コマンド, 80
 - crontab コマンド, 80
 - スケジューリング
 - 1 回かぎりのタスク, 80, 93
 - システムタスクの自動実行, 78
- システム動作
 - 追跡される動作のリスト, 48
 - データの自動収集, 72, 73
 - データの手動収集, 73
- システムの製品名
 - prtcnf コマンドでの表示, 13
- システムパフォーマンス 参照 パフォーマンス
- システムリソース
 - 概要, 46
- 自動システム動作
 - データ収集, 72, 73
 - レポート, 72, 73
- 新機能
 - svcadm enable system/sar:default コマンド, 72
- スケジューリング, 78, 79
 - 参照 crontab コマンド、at コマンド
 - 参照 smf
 - 1 回かぎりのシステムタスク, 80
 - at コマンド, 80
 - crontab コマンド, 80
 - SMF インスタンス, 81
 - 繰り返されるシステムタスク, 79, 80
 - 定期的なシステムタスク, 78

- スケジューリングクラス, 35
 - 指定, 39
 - 情報の表示, 27, 36, 36
 - 変更, 40
 - 優先順位の変更, 39, 41
 - 優先順位レベル, 35, 39
- スケジュールされている SMF サービス
 - 作成および編集, 82
 - システムタスクのスケジューリング, 81
- 制御
 - at コマンドの使用, 78
 - プロセス, 32
- 制御がきかなくなったプロセス, 43
- その日のメッセージ (MOTD) 機能, 23

た

- タイムシェアリングプロセス
 - スケジューリングパラメータの変更, 39
 - 優先順位
 - 概要, 35
 - 範囲, 35
 - 変更, 39, 41, 41
- 端末
 - プロセス制御, 27
- 端末デバイス
 - 管理, 99
 - SMF サービスの使用, 99
 - ログインサービスの設定, 100
- ディスクドライブ
 - 情報の表示
 - 空きディスク容量, 54
- ディスク容量
 - 情報の表示
 - df コマンド, 54
 - マウントポイント, 55
- ディレクトリ
 - プロセスの現在の作業ディレクトリ, 29, 29
- 電源サービス
 - 管理, 102
 - トラブルシューティングの問題, 105
- トラブルシューティングプロセス, 42, 43

は

- パフォーマンス
 - Ops Center を使用したモニター, 46
 - 追跡される動作のリスト, 48
 - 動作データの自動収集, 72, 73
 - 動作データの手動収集, 56, 73
 - ファイルアクセス, 56, 57
 - プロセス管理, 29, 41, 47
 - モニタリング用のツール, 49
 - レポート, 56
- 表示
 - LWP 情報, 29
 - アーキテクチャタイプ, 12
 - アドレス空間マップ, 29
 - 拡張ディスクの統計情報, 54
 - 仮想プロセッサタイプ, 20
 - システム情報
 - コマンド, 11
 - システム動作情報, 56, 73
 - システムにインストールされているメモリー, 14
 - 実行中のプロセスの情報, 30
 - 診断情報, 18
 - スケジューリングクラス情報, 27, 36, 36
 - 製品名情報
 - prtconf, 13
 - ディスクの使用状況の情報, 52
 - ディスク容量の統計情報, 54
 - デバイスのプロパティー値, 14
 - 日付と時間, 12
 - 物理プロセッサタイプ
 - psrinfo コマンド, 20
 - プロセス情報, 28, 29, 31
 - プロセスの情報, 30
 - プロセッサタイプ, 13
 - ホスト ID, 12
 - 優先順位情報, 27, 36
 - リリース情報, 12
 - リンクされたライブラリ, 29, 29
- ファイル
 - fstat と fcnt1 情報の表示, 29, 29, 29, 31
 - アクセス操作のチェック, 56, 57
- ファイルシステム
 - ディスク容量の使用状況, 54
 - マウントポイント, 55

フィールドエントリ
 SMF インスタンス, 82
フラグの追跡, 29
プログラム
 ディスク依存関係, 57
プロセス
 nice 値, 27, 41, 41, 42
 proc ツールコマンド, 28
 アドレス空間マップの表示, 29, 29
 アプリケーションスレッド, 47
 一時的に停止, 29
 管理のコマンド, 26
 強制終了, 29, 32
 現在の作業ディレクトリ, 29, 29, 31
 構造体, 27, 47
 再開, 29
 シグナルアクション, 29
 情報の表示, 31
 スケジューリングクラス, 35, 36, 39
 スタックトレース, 29
 制御, 32
 制御がきかない, 43
 ツリー, 29, 29, 31
 定義, 47
 トラブルシューティング, 42, 43
 開いているファイルの `fstat` と `fcntl` の情報,
 29, 29, 29, 31
 フラグの追跡, 29, 29
 優先順位, 41
 概要, 35, 41
 指定, 39, 39
 情報の表示, 27, 36
 スケジューリングクラス, 35, 39
 プロセスクラスのグローバル優先順位, 35,
 36
 変更, 39, 39, 41, 41, 41
 ユーザーモードの優先順位, 35
 用語, 47, 48
 リンクされたライブラリ, 29, 29
プロセスクラスのグローバル優先順位
 定義, 35
 表示, 36
プロセスの強制終了, 29
プロセスの再開, 29
プロセスファイルシステム (PROCFS), 28

プロセスを一時的に停止, 29
変更
 システム情報, 22
 システムのアイデンティティ, 23
 スケジューリングクラス, 40
 日付と時間, 22
 優先順位, 39, 41
 タイムシェアリングプロセス, 41
 変更, 41
ボーレート
 eeprom コマンドでの設定方法, 101
 ttymon 端末での設定方法, 101

ま

毎月のタスク
 crontab を使用したスケジューリング, 80
メモリー
 仮想プロセス, 48
 共有プロセスの仮想メモリー, 48
 情報の表示, 14
 プロセス構造体, 47

や

ユーザー構造体, 47
ユーザープロセス
 優先順位, 35
 優先順位の変更, 41, 41
ユーザーモードの優先順位, 35
優先順位 (プロセス)
 概要, 35, 41
 グローバル
 定義, 35
 表示, 36
 指定, 39, 39
 情報の表示, 27, 36
 スケジューリングクラス, 39
 タイムシェアリングプロセスの変更, 39, 41, 41
 変更, 39, 41
 ユーザーモードの優先順位, 35

ら

リアルタイムプロセス

クラスの変更, 40

A

at.deny ファイル, 78, 97

at コマンド

アクセス制御, 97, 97

エラーメッセージ, 98

概要, 78, 80, 93

システムタスク, 77

自動スケジューリング, 85

使用の制御, 78

ジョブキューの表示, 95, 95

ジョブファイル, 93

ジョブファイルの削除, 96

ジョブファイルの送信, 93

スケジューリング, 93

セキュリティー, 97

電子メールの送信確認, 94

at ジョブファイル

作成, 94

説明, 80

場所, 80

atjobs ディレクトリ, 85

C

CPU (中央処理装置)

情報の表示

使用時間, 27, 42

大量に使用するプロセス, 42

cron.allow ファイル, 90, 92

cron.deny ファイル, 90, 91

cron デーモン, 80, 84

crontab コマンド

crontab ファイルの削除, 90

アクセス制御, 90

エラーメッセージ, 92

削除, 89, 89

作成, 86

システムタスク, 77

使用されるファイル, 84, 84

使用制御, 92

使用の制御, 78, 91

スケジューリング, 83, 84

表示, 87, 88, 88

変更を保存せずに終了, 86

編集, 86

毎日のタスク, 78, 79, 80

crontab ファイル

アクセス制御, 91

確認, 87

構文, 85

削除, 89, 90

作成, 87

使用の拒否, 91

説明, 84, 85

デフォルト, 84

場所, 84

crontab ファイルの削除, 89

crontab を使用した毎日のタスクのスケジューリング, 79

D

df コマンド, 54, 55

-k オプション (キロバイト), 55

概要, 54

例, 55

dispadmin コマンド

概要, 38

E

/etc/cron.d/at.deny ファイル, 97, 97

/etc/cron.d/cron.allow ファイル, 90, 92

/etc/cron.d/cron.deny ファイル, 91

eeprom コマンド

ttymon 端末のボーレートの設定に使用, 101

F

fcnt1 情報, 29, 29, 29, 31

fsck コマンド, 80

fstat 情報, 29, 29, 29, 31

I

iostat コマンド, 52, 53

K

klwp 構造体, 47
kthread 構造体, 47

L

LWP (軽量プロセス)
構造体, 47
情報の表示, 29
定義, 47
プロセス, 47, 47

M

motd ファイル, 23

N

nice コマンド, 41, 41, 42
nice 値, 27, 41

P

/proc ディレクトリ, 28
perf ファイル, 72
pfiles コマンド, 29, 29, 31
pflags コマンド, 29, 29
pkill コマンド, 29, 32
pldd コマンド, 29, 29
pmap コマンド, 29, 29
priocntl コマンド
syntax, 36
概要, 38
構文, 37
proc 構造体, 27, 47
proc ツールコマンド, 29
PROCFS (プロセスファイルシステム), 28
prtconf コマンド, 14
システムの製品名の表示, 13
ps コマンド, 27, 30
概要, 27
グローバル優先順位の表示, 36
スケジューリングクラスの情報の表示, 27, 42

プロセスの完全な情報を表示する, 30
報告されるフィールド, 27

psig コマンド, 29, 29
psrinfo コマンドオプションによるチップマルチスレッド化機能の識別, 19
pstack コマンド, 29, 29
ptime コマンド, 29
ptree コマンド, 29, 29, 31
pwait コマンド, 29
pwdx コマンド, 29, 29, 31

S

sadd ファイル, 73
sa1 コマンド, 72
sa2 コマンド, 72, 73
sadc コマンド
システムデータの自動収集, 72, 73
ブート中に実行, 72
sar コマンド, 56, 73
概要, 56, 73
すべてのオプション, 74, 74
scheduled_method
要素, 82
SMF インスタンス
確認, 82
作成および編集, 82
時間値, 82
スケジューリング, 81
表示, 82
svcadm enable system/sar:default コマンド, 72
svcs を使用した定期的なジョブのスケジューリング, 78
sys crontab, 73

T

ttymon コンソールのボーレート速度の設定, 101

U

/usr/proc/bin ディレクトリ, 28, 29

V

/var/adm/sa/sa dd ファイル, 73

/var/spool/cron/atjobs ディレクトリ, 78, 80,
85, 85

/var/spool/cron/crontabs/root ファイル, 83

/var/spool/cron/crontabs/sys crontab, 73

/var/spool/cron/crontabs ディレクトリ, 84, 84

vmstat コマンド

概要, 49