



# Solaris カーネルのチューンアップ・リファレンスマニュアル

---

Sun Microsystems, Inc.  
901 San Antonio Road  
Palo Alto, CA 94303  
U.S.A. 650-960-1300

Part Number 806-7088-10  
2001 年 2 月

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

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

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

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

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリョーベイマジクス株式会社からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人 日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェイスマスタをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、フォントとして無断複製することは禁止されています。

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

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

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

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

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

Wnn6 は、オムロン株式会社で開発されたソフトウェアです。(Copyright OMRON Co., Ltd. 1999 All Rights Reserved.)

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK8」は株式会社ジャストシステムの著作物であり、「ATOK8」にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。

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

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

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

Unicode は、Unicode, Inc. の商標です。

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

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

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

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

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

原典: Solaris Tunable Parameters Reference Manual

Part No: 806-6779-10

Revision A



# 目次

---

はじめに	13
<b>1. Solaris システムのチューニングの概要</b>	<b>19</b>
Solaris システムのチューニング	19
変数の説明の書式	20
Solaris カーネルのチューニング	22
/etc/system ファイル	23
adb	24
kadb	25
mdb	25
特別な構造	27
システム構成情報の閲覧	27
sysdef	27
kstat 構造体群	28
<b>2. Solaris カーネルのチューニング可能パラメータ</b>	<b>29</b>
一般的なパラメータ	30
physmem	30
lwp_default_stksize	31
fsflush とそれに関連するチューニング可能パラメータ	32
fsflush	32

tune\_t\_fsflushr 33  
autoup 34  
dopageflush 35  
doiflush 36  
プロセスのサイズのチューニング可能パラメータ 37  
maxusers 37  
reserved\_procs 39  
pidmax 40  
max\_nprocs 41  
maxuprc 42  
ページング関連のチューニング可能パラメータ 43  
lotsfree 44  
desfree 45  
minfree 47  
throttlefree 48  
pageout\_reserve 49  
cachefree 51  
priority\_paging 52  
pages\_pp\_maximum 52  
tune\_t\_minarmem 53  
fastscan 54  
slowscan 55  
min\_percent\_cpu 55  
handspreadpages 56  
pages\_before\_pager 57  
maxpgio 58  
スワッピング関連の変数 58  
swapfs\_reserve 59

swapfs_minfree	60
一般的なカーネル変数	60
noexec_user_stack	60
カーネルメモリアロケータ	61
kmem_flags	62
一般的なドライバ	65
moddebug	65
一般的な入出力	67
maxphys	67
rlim_fd_max	68
rlim_fd_cur	69
一般的なファイルシステム	70
ncsize	70
rstchown	71
segkpsize	72
dnlc_dir_enable	74
dnlc_dir_min_size	74
dnlc_dir_max_size	75
UFS	75
bufhwm	75
ndquot	78
ufs_ninode	79
ufs:ufs_WRITES	81
ufs:ufs_LW と ufs:ufs_HW	82
TMPFS	83
tmpfs:tmpfs_maxkmem	83
tmpfs:tmpfs_minfree	84
仮想端末	85

pt_cnt	86
pt_pctofmem	87
pt_max_pty	88
ストリーム	88
nstrpush	88
strmsgsz	89
strctlsz	89
System V メッセージキュー	90
msgsys:msginfo_msgmax	91
msgsys:msginfo_msgmnb	92
msgsys:msginfo_msgmni	92
msgsys:msginfo_msqtql	93
System V セマフォ	93
semsys:seminfo_semmni	94
semsys:seminfo_semmns	94
semsys:seminfo_semvmx	95
semsys:seminfo_semmsl	96
semsys:seminfo_semopm	96
semsys:seminfo_semmnu	97
semsys:seminfo_semume	98
semsys:seminfo_semaem	99
System V 共有メモリー	100
shmsys:shminfo_shmmax	100
shmsys:shminfo_shmmin	101
shmsys:shminfo_shmmni	102
shmsys:shminfo_shmseg	102
segspt_minfree	103
スケジューリング	104

	rechoose_interval	104
タイマー		105
	hires_tick	105
	timer_max	105
Sun4u 特有		106
	consistent_coloring	106
<b>3. NFS チューニング可能パラメータ</b>		<b>109</b>
NFS 環境のチューニング		109
NFS モジュールのパラメータ		110
	nfs:nfs3_pathconf_disable_cache	110
	nfs:nfs_allow_preepoch_time	110
	nfs:nfs_cots_timeo	112
	nfs:nfs3_cots_timeo	113
	nfs:nfs_do_symlink_cache	114
	nfs:nfs3_do_symlink_cache	115
	nfs:nfs_dynamic	115
	nfs:nfs3_dynamic	116
	nfs:nfs_lookup_neg_cache	117
	nfs:nfs3_lookup_neg_cache	118
	nfs:nfs_max_threads	119
	nfs:nfs3_max_threads	120
	nfs:nfs_nra	121
	nfs:nfs3_nra	122
	nfs:nrnode	123
	nfs:nfs_shrinkreaddir	125
	nfs:nfs_write_error_interval	126
	nfs:nfs_write_error_to_cons_only	127
	nfs:nfs_disable_rddir_cache	127

nfs:nfs3_bsize	128
nfs:nfs_async_clusters	130
nfs:nfs3_async_clusters	131
nfs:nfs_async_timeout	132
nfs:nacache	133
nfs:nfs3_jukebox_delay	135
nfs:nfs3_max_transfer_size	136
nfssrv モジュールのパラメータ	137
nfssrv:nfs_portmon	137
nfssrv:rfs_write_async	138
nfssrv:nfsauth_ch_cache_max	139
nfssrv:exi_cache_time	140
rpcmod モジュールのパラメータ	141
rpcmod:clnt_max_conns	141
rpcmod:clnt_idle_timeout	142
rpcmod:svc_idle_timeout	143
rpcmod:svc_default_stksize	143
rpcmod:svc_default_max_same_xprt	144
rpcmod:maxdupreqs	145
rpcmod:cotsmaxdupreqs	146
<b>4. TCP/IP チューニング可能パラメータ</b>	<b>149</b>
TCP/IP パラメータのチューニングの概要	149
TCP/IP パラメータの検査	150
Internet Request for Comments (RFC)	150
IP チューニング可能パラメータ	151
ip_icmp_err_interval と ip_icmp_err_burst	151
ip_forwarding と ip6_forwarding	151
xxx:ip_forwarding	152

ip_respond_to_echo_broadcast と ip6_respond_to_echo_multicast	152
ip_send_redirects と ip6_send_redirects	153
ip_forward_src_routed と ip6_forward_src_routed	153
ip_addrs_per_if	154
ip_strict_dst_multihoming と ip6_strict_dst_multihoming	154
TCP チューニング可能パラメータ	156
tcp_deferred_ack_interval	156
tcp_deferred_acks_max	157
tcp_wscale_always	157
tcp_tstamp_always	158
tcp_xmit_hiwat	159
tcp_recv_hiwat	159
tcp_max_buf	160
tcp_cwnd_max	160
tcp_slow_start_initial	161
tcp_slow_start_after_idle	161
tcp_sack_permitted	162
tcp_rev_src_routes	163
tcp_time_wait_interval	163
tcp_conn_req_max_q	164
tcp_conn_req_max_q0	165
tcp_conn_req_min	166
/etc/system ファイルに設定する TCP パラメータ	167
特別な注意を要する TCP パラメータ	168
UDP チューニング可能パラメータ	173
udp_xmit_hiwat	173
udp_recv_hiwat	174

- 特別な注意を要する UDP パラメータ 174
- ルート別のメトリック 175
- 5. システム機能のパラメータ 177
  - システムのデフォルトのパラメータ 178
    - cron 178
    - devfsadm 178
    - dhcpcagent 178
    - fs 178
    - inetinit 179
    - init 179
    - kbd 179
    - login 179
    - nfslogd 179
    - passwd 180
    - power 180
    - su 180
    - sys-suspend 180
    - tar 180
    - utmpd 181
  - A. チューニング可能パラメータの変更履歴 183
    - カーネルパラメータ 183
      - プロセスのサイズに関するチューニング可能パラメータ 183
      - ページング関連のチューニング可能パラメータ 186
      - 一般的なカーネル変数 188
      - 一般的な入出力 189
      - 仮想端末 190
      - Sun4u 特有 191
    - 機能が削除されたパラメータ 192

ページング関連のチューニング可能パラメータ	192
System V メッセージパラメータ	192
System V セマフォのパラメータ	195
NFS モジュールのパラメータ	195
<b>B. このマニュアルの改訂履歴</b>	<b>197</b>
現行バージョン - Solaris 8 1/01 リリース	197
前のリリースからの変更	197
shmsys:shminfo_shmmin (Solaris 8 1/01 リリース)	197
semsys:seminfo_semmnu (Solaris 8 1/01 リリース)	198
索引	<b>199</b>



## はじめに

---

『Solaris カーネルのチューンアップ・リファレンスマニュアル』では、Solaris™ のカーネルとネットワークのチューニング可能パラメータに関する参照情報を掲載しています。このマニュアルには、CDE や Java™ 環境に関するチューニング可能パラメータの情報は記載していません。

このマニュアルは、SPARC™ ベースシステムと IA ベースシステムの両方の情報を記載しています。

---

注 - Solaris オペレーティング環境は、2 種類のハードウェア (プラットフォーム) 上で動作します。つまり、SPARC と IA (Intel アーキテクチャ) です。Solaris オペレーティング環境は、64 ビットと 32 ビットの両方のアドレス空間で動作し、IA では 32 ビットのアドレス空間でのみ動作します。このマニュアルで説明する情報は、章、節、注、箇条書き、図、表、例、またはコード例において特に明記しない限り、両方のプラットフォームおよびアドレス空間に該当します。

---

---

## 対象読者

Solaris システムの管理者は、状況によっては、カーネルのチューニング可能パラメータを変更しなければならない場合があります。このマニュアルは、このような習熟した Solaris システム管理者を対象としています。Solaris のチューニング可能パラメータの変更に関するガイドラインについては、19ページの「Solaris システムのチューニング」を参照してください。

---

## 内容の紹介

このマニュアルは、次の章から構成されています。

---

章	内容
第 1 章	Solaris システムのチューニングに関する概要、およびこのマニュアルでカーネルのチューニング可能パラメータの説明に使用する形式の説明
第 2 章	Solaris カーネルのチューニング可能パラメータの説明 (カーネルメモリーやファイルシステム、プロセスサイズ、ページングのパラメータなど)
第 3 章	NFS チューニング可能パラメータの説明 (シンボリックリンクのキャッシュや、動的再転送、RPC セキュリティのパラメータなど)
第 4 章	TCP/IP のチューニング可能パラメータの説明 (IP 転送やソースルーティング、バッファサイジングのパラメータなど)
第 5 章	特定のシステム機能のデフォルト値を変更するためのパラメータの説明。変更するには、 <code>/etc/default</code> ディレクトリにあるファイルを変更します。
付録 A	変更や廃止されたパラメータの履歴
付録 B	このマニュアルの改訂履歴 (現行の Solaris リリースバージョンを含む)

---

---

## 関連マニュアル

Solaris システムのチューニングには、次の書籍が参考になります。

- 『*Configuration and Capacity Planning for Solaris Servers*』 Brian L. Wong 著、Sun Microsystems Press、ISBN 0-13-349952-9
- 『*NFS Illustrated*』 Brent Callaghan、Addison Wesley 著、ISBN 0-201-32570-5

- 『Resource Management』 Richard McDougall、Adrian Cockcroft、Evert Hoogendoorn、Enrique Vargas、Tom Bialaski 著、Sun Microsystems Press、ISBN 0-13-025855-5
- 『Sun Performance and Tuning: SPARC and Solaris』 Adrian Cockcroft 著、Sun Microsystems Press/PRT Prentice Hall、ISBN 0-13-149642-3

---

## Solaris のチューニングに関するその他の情報源

さらに、Solaris のチューニングに関する情報として次のものがあります。

目的	参照先
パフォーマンスチューニングクラス	<a href="http://suned.sun.com">http://suned.sun.com</a>
オンラインパフォーマンスチューニング情報	<a href="http://www.sun.com/sun-on-net/performance">http://www.sun.com/sun-on-net/performance</a>
Sun Microsystems Press によるパフォーマンスチューニング関連の文書の注文	<a href="http://www.sun.com/books/blueprints.series.html">http://www.sun.com/books/blueprints.series.html</a>

---

## Sun のマニュアルの注文方法

専門書を扱うインターネットの書店 Fatbrain.com から、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、<http://www1.fatbrain.com/documentation/sun> の Sun Documentation Center をご覧ください。

---

## Sun のオンラインマニュアル

<http://docs.sun.com> では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。

---

## 表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。  ls -a を使用してすべてのファイルを表示します。  system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% <b>su</b>  password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。

表 P-1 表記上の規則 続く

字体または記号	意味	例
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	<code>sun% grep `^#define \ XV_VERSION_STRING`</code>

ただし AnswerBook2™ では、ユーザーが入力する文字と画面上のコンピュータ出力は区別して表示されません。

コード例は次のように表示されます。

■ C シェル

```
machine_name% command y|n [filename]
```

■ C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

■ Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

■ Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

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

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

---

## 一般規則

- このマニュアルでは、「IA」という用語は、Intel 32 ビットのプロセッサアーキテクチャを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Celeron、Pentium III、Pentium III Xeon の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。

## Solaris システムのチューニングの概要

---

この章では、このマニュアルで使用するチューニング情報の記載形式の概要を示します。また、Solaris システムのチューニングの別の方法についても説明します。

- 19ページの「Solaris システムのチューニング」
- 20ページの「変数の説明の書式」
- 22ページの「Solaris カーネルのチューニング」

---

## Solaris システムのチューニング

Solaris は、SPARC および Intel プロセッサで動作する、マルチスレッドでスケールアップ可能な UNIX オペレーティング環境です。Solaris は、システムの負荷に自動的に対応します。したがって、Solaris では最小限のチューニングしか必要ありませんが、それでもチューニングが必要になることがあります。このマニュアルでは、Solaris 環境で公式にサポートされているカーネルのチューニングオプションの詳細を記述します。

Solaris カーネルは、常にロードされているコア部分と、参照が発生するとロードされるロード可能モジュールから構成されています。このマニュアルのカーネル部分で述べる変数の多くはコア部分に存在しますが、一部はロード可能モジュールに存在します。

システムのチューニングをする際に考慮する必要があることは、さまざまなシステム変数を設定する方法は、処理効率を高めるという目的にとって、多くの場合、一番効率の良くない方法であるということです。最も効果的なチューニング方法は、

通常、アプリケーションの動作を変更することです。また、物理メモリーを増やしたり、ディスクの入出力パターンのバランスをとることも効果があります。このマニュアルに記載された変数の値を1つ変更しただけで、システムパフォーマンスに意味のある影響が現れることは、ごく限られた場合にしか起きません。

さらに、もう1つ注意すべきことがあります。あるシステムの `/etc/system` 設定値は、全体的であれ、部分的であれ、別のシステムの環境にも適しているとは限りません。したがって、使用する環境に応じて、このファイルに設定する値を慎重に検討する必要があります。このマニュアルで述べるシステム変数を変更する場合は、システムの動作を前もって理解していなければなりません。



---

**注意** - このマニュアルで記述される変数やその意味は、リリースによって変わる可能性があります。リリースとは、Solaris Update リリースの場合もあり、Solaris 8 などのような新しいバージョンの場合もあります。これらの変数とその意味は、予告なく変更されることがあります。

---

## 変数の説明の書式

各変数の説明は、次の形式に従って行われます。

- 変数名
- 説明
- データ型
- デフォルト
- 単位
- 範囲
- 動的か
- 検査
- 暗黙的制約
- どのような場合に変更するか
- コミットレベル (または安定性レベル)
- 変更履歴

変数名	<p>「変数名」は、<code>/etc/system</code> ファイルに指定したり、<code>/etc/default/facility</code> ファイルに指定されている名前です。</p> <p>ほとんどの変数名は、コロン (:) を伴わない <i>variable</i> の形式をとります。このような名前は、カーネルのコア部分内の変数を表しています。名前にコロンが含まれている場合、コロンの左側の文字列はロード可能モジュールの名前を表し、コロンの右側の文字列はモジュール内の変数名を表します。次はその例です。</p> <div style="border: 1px solid black; padding: 2px; margin: 10px 0; width: fit-content;"> <code>module_name:variable</code> </div>
説明	<p>この変数が何をどのように制御するかを簡単に説明します。</p>
データ型	<p>符号付きまたは符号なしの、<code>short</code>、<code>long</code>、または整数です。ただし、<code>long</code> のサイズは、以下のようにシステムによって異なります。</p> <ul style="list-style-type: none"> <li>■ 32 ビットのカーネルが動作しているシステムでは、<code>long</code> のサイズは整数と同じです。</li> <li>■ 64 ビットのカーネルが動作しているシステムでは、<code>long</code> のビット幅は整数の 2 倍です。たとえば、符号なし整数は 32 ビット、符号なし <code>long</code> は 64 ビットです。</li> </ul>
デフォルト	<p>システムがデフォルト値として使用している値です。</p>
単位	<p>単位の種類を表します。この項目は省略されることがあります。</p>
範囲	<p>システムの検査で取り得る範囲や、データ型の上下限を表す範囲です。</p> <ul style="list-style-type: none"> <li>■ MAXINT — 符号付き整数の最大値 (2,147,483,647) を表します。</li> </ul>

- MAXUINT — 符号なし整数の最大値 (4,294,967,295) を表します。

動的か	動作中のシステムで adb、mdb、kadb デバッガを使用して変更できる場合は「はい」、ブート時の初期設定のみの場合は「いいえ」です。
検査	変数の、 <code>/etc/system</code> ファイルで指定された、またはデフォルトの値に対して、システムが適用する検査内容を示します。また、検査がいつ適用されるかも示します。
暗黙的制約	変数に対する暗黙的な制約事項 (特に他の変数との関係において) を表します。この項目は省略されることがあります。
どのような場合に変更するか	この値の変更が必要になる理由、エラーメッセージ、戻り値を示します。
コミットレベル (安定性レベル)	インタフェースの安定性を表します。このマニュアルで記述するパラメータの多くは「発展中 (Evolving)」または「変更の可能性あり (Unstable)」のいずれかに分類されます。詳細は、 <code>attributes(5)</code> のマニュアルページを参照してください。
変更履歴	該当する場合は、付録の「変更履歴」へのリンクを示します。この項目は省略されることがあります。

---

## Solaris カーネルのチューニング

次の表では、パラメータに適用可能なチューニングの方法を示します。

チューニングパラメータの適用方法	参照先
<code>/etc/system</code> ファイルの変更	23ページの「 <code>/etc/system</code> ファイル」
デバッガ (adb) の使用	24ページの「adb」
カーネルデバッガ (kadb) の使用	25ページの「kadb」
モジュールデバッガ (mdb) の使用	25ページの「mdb」
ndd コマンドによる TCP/IP パラメータの設定	第 4 章
<code>/etc/default</code> 下のファイルの変更	178ページの「システムのデフォルトのパラメータ」

## `/etc/system` ファイル

`/etc/system` ファイルは、カーネル変数の値を静的に調整する機構を提供します。このファイルに指定された値は、ブート時に読み込まれ適用されます。このファイルに対する変更は、システムがリブートされるまでオペレーティングシステムに適用されません。

Solaris 8 より前のリリースでは、システム変数の値を設定する `/etc/system` のエントリは、次の 2 つの処理段階に分けて適用されていました。

- 最初の処理段階では、さまざまなブートストラップ変数 (`maxusers` など) を取得し、基本となるシステムパラメータを初期設定します。
- 次の処理段階では、ブートストラップ変数を使用して基本構成を計算し、`/etc/system` ファイルに指定されているすべての値を適用します。ブートストラップ変数に関しては、この処理段階で計算された値やリセットされた値は、`/etc/system` ファイルに指定されている値で置き換えられます。

2 つめの処理段階では、許容されないような値を変数に設定したり、初期構成で値が上書きされる変数 (`max_nprocs` など) に値を割り当てたりするため、この処理段階はユーザーや管理者にとって混乱を招く場合があります。

Solaris 8 リリースでは、構成パラメータを計算する前にすべての値を設定する 1 つの処理段階が設けられました。

## 例 — /etc/system を使用したパラメータの設定

次の /etc/system エントリでは、NFS バージョン 2 ソフトウェアを使用してマウントされたファイルシステムに対し先読みするブロックの数を指定します。

```
set nfs:nfs_nra=4
```

### 適正でない値からの復元

値を変更する前に /etc/system のコピーを作成しておけば、不正な値を簡単に元の値に戻せます。

```
# cp /etc/system /etc/system.good
```

/etc/system に設定した値が原因でシステムがブートできない状態になった場合は、次のコマンドでブートします。

```
ok boot -a
```

このコマンドを実行すると、ブートプロセスで使用する各ファイルの名前をシステムから要求されます。/etc/system ファイルの名前が要求されるまで Return キーを押して、デフォルトの値を適用しま

す。Name of system file [/etc/system]: プロンプトが表示されたら、正しい /etc/system ファイルのコピーの名前か /dev/null を入力します。

```
Name of system file [/etc/system]: /etc/system.good
```

/dev/null を入力すると、システムは /dev/null から構成情報を読み込もうとします。しかし、/dev/null は空のため、システムはデフォルト値を使用します。システムがブートした後、/etc/system ファイルを修正できます。

システムの回復については、『Solaris のシステム管理 (第 1 巻)』でさらに記述されています。

## adb

adb は実行時デバッガです。スーパーユーザーは、-k オプションを指定して adb を実行し、動作中のカーネルの変数を見ることができます。さらに、-k オプションと共に -w オプションを指定すると、スーパーユーザーは動作中のカーネルメモ

リー内の値を変更することができます。この方法によるすべての変更は、システムがリブートされると失われます。

## 例 — adb を使用した値の変更

整数変数 `maxusers` の現在の値を `0x200` に変更するには、次のようにします。

```
# adb -kw
physmem f7c6
maxusers/D
maxusers:
maxusers:      495
maxusers/W 200
maxusers:      0x1ef          =          0x200
$g
```

実際に変更する場合は、`maxusers` を変更したい変数のアドレスに、値を設定したい値に置き換えて、このコマンドを実行します。

adb コマンドの使い方については、adb(1) のマニュアルページを参照してください。

## kadb

kadb はブート可能なカーネルデバグガであり、その一般的な構文は adb と同じです。例外については、kadb(1M) のマニュアルページを参照してください。kadb の利点の 1 つは、ユーザーがブレイクポイントを設定でき、そのブレイクポイントに達したらデータを調べたり、カーネルコードを 1 つずつ実行できることです。

kadb -d を指定してシステムをブートすると、コアカーネルの変数に値を設定できます。ただし、ロード可能モジュールの値は、そのモジュールが実際にロードされたときに値が設定されます。

『*Writing Device Drivers*』の「Debugging」の節に kadb コマンドの使い方の簡単な説明が記載されています。

## mdb

mdb(1) は Solaris 8 リリースで新たに提供されたモジュラデバグガです。このデバグガは、拡張が容易であるという点で現在の Solaris デバグガの中でも独特のもの

です。adb マクロの作成を試みたことがあるユーザーなら、この作業が簡単でないことがお分かりでしょう。このデバッガのプログラミング API を使用して、モジュールをコンパイルすることによって、デバッガのコンテキスト内で希望する処理を実行することができます。mdb は adb(1) や crash(1M) との下位互換性があります。

さらに、mdb(1) には、コマンド行での編集、コマンド履歴、組み込み出力ページャ、構文チェック、コマンドパイプラインなどの、いくつかの便利な機能があります。カーネルに対する事後検分用のデバッガとしては、このデバッガをお勧めします。

## 例 — mdb を使用した値の変更

整数変数 `maxusers` の値を 5 から 6 に変更するには、次のようにします。

```
# mdb -kw
Loading modules: [ unix krtld genunix ip logindmux ptm nfs ipc lofs ]
> maxusers/D
maxusers:
maxusers:          495
> maxusers/W 200
maxusers:          0x1ef          =          0x200
> $q
```

実際に変更する場合は、`maxusers` を変更したい変数のアドレスに、値を設定したい値に置き換えて、このコマンドを実行します。

モジュールデバッガの詳細は、『Solaris モジュールデバッガ』を参照してください。

adb や kadb、mdb では、モジュール名接頭辞を指定する必要はありません。モジュールが読み込まれると、そのモジュールのシンボルは、コアカーネルのシンボルやすでにロードされているその他のモジュールのシンボルとの共通の名前空間を形成するからです。

たとえば、UFS モジュールがロードされると想定した場合、`ufs:ufs_WRITES` は、個々のデバッガでは `ufs_WRITES` としてアクセスされます。しかし、`/etc/system` ファイルに設定する場合は、`ufs:` 接頭辞が必要です。adb や kadb でモジュール名接頭辞を指定すると、未定義のシンボルのメッセージが出力されます。

---

## 特別な構造

Solaris のチューニング変数にはいろいろな形式があります。tune\_t\_gpgslo、tune\_t\_fsflushr、tune\_t\_minarmem、tune\_t\_minasmem、tune\_t\_flckrec の実行時の状態は、/usr/include/sys/tuneable.h に定義された tune 構造体で表わされます。カーネルが初期設定された後は、これらの変数の値は、この tune 構造体の対応フィールドに入ります。

以前のバージョンの『Solaris のシステム管理 (第 2 巻)』を始め、さまざまなマニュアルには、tune 構造体の変数に値を設定する適切な方法として tune:field-name (field-name には前述の実際の変数名を指定する) という構文が指定されていました。しかし、この設定は何のメッセージもなく失敗します。ブート時にこの構造体に変数を設定するには、必要なフィールド名に対応する特別な変数を初期設定する必要があります。そうすれば、これらの値がシステム初期設定プロセスで tune 構造体にロードされます。

複数のチューニングパラメータが置かれるもう 1 つの構造体に、v という名前の var 構造体があります。var 構造体の定義は /usr/include/sys/var.h にあります。autoup や bufhwm などの変数の実行時の状態はここに格納されます。

システムが動作中に tune 構造体や v 構造体を変更しないでください。システムの動作中にこれらの構造体のフィールドを変更すると、システムがパニックになることがあります。

---

## システム構成情報の閲覧

システム構成を調べるツールはいくつかあります。ツールには、root 特権が必要なものもあれば、非特権ユーザーが実行できるものもあります。すべての構造体やデータ項目はカーネルデバッグで調べることができます (動作中のシステムで adb や mdb を使用したり、kadb のもとでシステムをブートする)。

### sysdef

sysdef (1M) コマンドは、System V IPC 設定や、STREAMS チューニング可能パラメータ、プロセスリソースの制限値、tune 構造体と v 構造体の各部分の値などを

表示します。たとえば、512 バイトの Ultra™ 80 システム上で `sysdef` を実行すると、「チューニング可能パラメータ」セクションは次のように表示されます。

10387456	maximum memory allowed in buffer cache (bufhwm)
7930	maximum number of processes (v.v_proc)
99	maximum global priority in sys class (MAXCLSYSPRI)
7925	maximum processes per user id (v.v_maxup)
30	auto update time limit in seconds (NAUTOUP)
25	page stealing low water mark (GPGSLO)
5	fsflush run rate (FSFLUSHR)
25	minimum resident memory for avoiding deadlock (MINARMEM)
25	minimum swapable memory for avoiding deadlock (MINASMEM)

---

## kstat 構造体群

`kstat` データ構造体群は、さまざまなカーネルのサブシステムやドライバによって維持されています。この構造体群は、カーネル内のデータをユーザープログラムに提供する機構を提供します。この機構を利用する場合、プログラムはカーネルのメモリーを読んだり、`root` 特権を持つ必要はありません。詳細は、`kstat(3KSTAT)` を参照してください。

Solaris 8 リリースでは、コマンド行インタフェースから `kstat` 構造体群を選択して表示できる `kstat(1M)` コマンドが新しく追加されています。`kstat` 情報を処理する Perl モジュール `kstat(3EXT)` も利用できます。

## Solaris カーネルのチューニング可能パラメータ

---

この章では、ほとんどの Solaris のカーネルのチューニング可能パラメータについて記述しています。NFS のチューニング可能パラメータについては第 4 章を、TCP/IP のチューニング可能パラメータについては、第 3 章をそれぞれ参照してください。

- 30ページの「一般的なパラメータ」
- 32ページの「fsflush とそれに関連するチューニング可能パラメータ」
- 37ページの「プロセスのサイズのチューニング可能パラメータ」
- 43ページの「ページング関連のチューニング可能パラメータ」
- 58ページの「スワッピング関連の変数」
- 60ページの「一般的なカーネル変数」
- 61ページの「カーネルメモリアロケータ」
- 65ページの「一般的なドライバ」
- 67ページの「一般的な入出力」
- 70ページの「一般的なファイルシステム」
- 75ページの「UFS」
- 83ページの「TMPFS」
- 85ページの「仮想端末」
- 88ページの「ストリーム」
- 90ページの「System V メッセージキュー」

- 93ページの「System V セマフォ」
- 100ページの「System V 共有メモリー」
- 104ページの「スケジューリング」
- 105ページの「タイマー」
- 106ページの「Sun4u 特有」

## 一般的なパラメータ

この節では、物理メモリーやスタックサイズに関する一般的なカーネルパラメータについて説明します。

### physmem

説明	OS とファームウェアを除いたメモリーの物理ページ数に関するシステムの値を変更します。
データ型	符号なし long
デフォルト	そのシステムで使用できる物理メモリーのページ数。これには、コアカーネルとそのデータが格納されているメモリーは含まれません。
範囲	1 からシステムの物理メモリーの総量まで
単位	ページ
動的か	いいえ
検査	なし
どのような場合に変更するか	より少ない物理メモリーで実行したときの影響を調べたい場合。このパラメータに対しては、コアカーネルやそのデータ、その他のさまざまなデータ構造体(起動処理の初期に割り当て)などのメモリーは考慮されません。したがって、physmem の値は、より小さなメモリー量を

表わすよう、想定したページ数より小さくすべきです。

コミットレベル 変更の可能性あり

## lwp\_default\_stksize

説明	カーネルスレッドが作成されるときに、呼び出しルーチンが使用するサイズを明示的に指定しない場合に使用される、スタックのサイズのデフォルト値。
データ型	整数
デフォルト	32 ビットの SPARC や IA ベースのプラットフォームでは 8192、64 ビットの sun4u プラットフォームでは 16,384
範囲	0 から 262,144
単位	バイト。ただし、getpagesize(3C) から返される値の倍数。
動的か	はい。変数の変更後、作成されるスレッドに影響があります。
検査	8192 以上 262,144 (256 × 1024) 以下で、かつシステムページサイズの倍数でなければなりません。これらの条件が満たされないと、次のメッセージが表示されます。

```
Illegal stack size, Using N
```

N の値は、上述のデフォルト値です。

どのような場合に変更するか

スタック容量が足りないためにシステムがパニックになる場合。この問題を解決する最もよい方法は、システムが容量を使い果たす原因を明らかにし、それを修正することです。デフォルトのスタックサイズを増やすと、ほとんどすべてのカーネルスレッドのスタックが大きくなるた

め、カーネルのメモリー使用量が増加します。増加した容量は一般には使用されず、無駄になります。さらに、カーネルの使用量が増えると、同じメモリープールを使用する他のリソースの容量が少なくなるため、システムの作業を行う能力が低下するおそれがあります。副次的な影響には、カーネルが作成できるスレッドの数が少なくなるということがあります。したがって、この方法は、根本的な原因が解消されるまでの一時的な回避策として使用すべきです。

コミットレベル

変更の可能性あり

---

## fsflush とそれに関連するチューニング可能パラメータ

この節では、fsflush とそれに関連するチューニング可能パラメータについて説明します。

### fsflush

システムデーモン fsflush は定期的に行われ、主に次の3つのことを行います。

- fsflush は、呼び出されるたびに次のことを行います。
  1. 一定の時間を経過したダーティーなファイルシステムページをディスクにフラッシュします。
  2. メモリーの一部を調べ、変更されているページをそれぞれのバッキングストアに書き込みます。ページは、変更されており、かつ次の条件のどれにも該当しない場合書き込まれます。
    - カーネルページである
    - 空いている
    - ロックされている
    - スワップデバイスに関連付けられている
    - 現在入出力操作に使用されている

この結果、書き込み権に基づいて mmap され、かつ実際に変更されているファイルのページがフラッシュされます。

ページはバッキングストアにフラッシュされますが、それを使用しているプロセスとの接続は保たれます。フラッシュしておく、システムのメモリーが不足したときのページの再利用が簡単になります。これは、フラッシュ後にそのページが変更されていなければ、ページを回収する前にそのページをバッキングストアに書き出す必要がなくなり、遅延を避けられるからです。

3. ファイルシステムのメタデータをディスクに書き込みます。この書き込みは  $n$  回目の呼び出しごとに行われます。 $n$  はさまざまな構成変数から計算されます。詳細は、33ページの「`tune_t_fsflushr`」と34ページの「`autoup`」を参照してください。

呼び出しをどのような頻度で行うかや、メモリー走査を実行するかどうか、ファイルシステムデータのフラッシュを行うかどうか、そしてそのフラッシュをどのような頻度で行うかは、構成可能です。

ほとんどのシステムでは、`fsflush` によって、メモリーの走査と、ファイルシステムメタデータの同期化を行うのが一般的です。システムの使用状況によっては、メモリーの走査はほとんど意味がなかったり、CPU 時間を使用しすぎることがあります。

## tune\_t\_fsflushr

説明	<code>fsflush</code> を呼び出す間隔を秒単位で指定します。
データ型	符号付き整数
デフォルト	5
範囲	1 から MAXINT
単位	秒
動的か	いいえ
検査	値がゼロ以下の場合は、値は 5 に再設定され、警告メッセージが出力されます。このチェックが行われるのはブート時だけです。
どのような場合に変更するか	下記の <code>autoup</code> を参照してください。

## autoup

### 説明

個々の呼び出しでダーティページに関して検査するメモリー量と、ファイルシステム同期操作の頻度を、`tune_t_flushr` とともに制御します。

さらに、`autoup` の値は、空リストからバッファを書き出すかどうかの制御にも使用されます。`B_DELWRI` フラグが付いているバッファ (変更されているファイルコンテンツページ) は、空リストに置かれている時間が `autoup` 秒を超えると書き出されます。`autoup` の値を増やすと、バッファがメモリーに置かれている時間が長くなります。

### データ型

符号付き整数

### デフォルト

30

### 範囲

1 から MAXINT

### 単位

秒

### 動的か

いいえ

### 検査

`autoup` がゼロ以下の場合は、30 に再設定され、警告メッセージが出力されます。このチェックが行われるのはブート時だけです。

### 暗黙的制約

`autoup` は `tune_t_fsflushr` の整数倍で、かつ `tune_t_fsflushr` の 6 倍以上であるべきです。そうでないと、`fsflush` が呼び出されるたびに余計なメモリーが走査されます。

`dopageflush` がゼロでない場合にメモリーをチェックするためには、(全体のシステムページ数  $\times$  `tune_t_fsflushr`) が `autoup` 以上でなければなりません。

どのような場合に変更するか

autoup や tune\_t\_fsflushr の変更が必要になる状況はいくつかあります。

- 大きなメモリーをもつシステム — この場合には、autoup を増やすと、fsflush の個々の呼び出しで走査されるメモリー量が少なくなります。
- メモリーの要求量が最小限のシステム — autoup と tune\_t\_fsflushr を両方とも増やすと、走査の回数が減ります。autoup 対 tune\_t\_fsflushr の現在の比率を維持するには autoup も増やす必要があります。
- 一時ファイルの数が多いシステム (メールサーバーやソフトウェアビルドマシンなど) — 多数のファイルが作成されて削除された時、fsflush によって、これらのファイルのデータページがディスクに不必要に書き込まれるおそれがあります。

コミットレベル

変更の可能性あり

## dopageflush

説明

fsflush の呼び出しで、変更されたページをメモリーから探すかどうかを制御します。fsflush の個々の呼び出しでは、システムのメモリーページ数が調べられます (動的再構成の操作によりメモリーページの数が変わっていることがある)。個々の呼び出しでは、(全体のページ数 × tune\_t\_fsflushr) / autoup ページが走査されます。

データ型

符号付き整数

デフォルト

1 (有効)

範囲

0 (無効) または 1 (有効)

単位	切り替え (オン/オフ)
動的か	はい
検査	なし
どのような場合に変更するか	システムページスキャナの実行がまれな場合 (vmstat 出力の <code>sr</code> 欄に値 0 が示される)。
コミットレベル	変更の可能性あり

## doiflush

説明	<code>fsflush</code> 呼び出しでファイルシステムメタデータの同期化を行うかどうかを制御します。同期化は、 <code>fsflush</code> の $N$ 回目の呼び出しごとに行われます。ここで $N$ は $(\text{autoup} / \text{tune\_t\_fsflushr})$ です。これは整数の割り算であるため、 <code>tune_t_fsflushr</code> が <code>autoup</code> より大きいと、コードが反復カウンタが $N$ 以上であるかどうかをチェックするので、同期化は <code>fsflush</code> が呼び出されるたびに行われます。 $N$ は <code>fsflush</code> を実行するときに 1 度だけ計算されるため、その後で <code>tune_t_fsflushr</code> や <code>autoup</code> を変更しても、同期化操作の頻度に影響はありません。
データ型	符号付き整数
デフォルト	1 (有効)
範囲	0 (無効) または 1 (有効)
単位	切り替え (オン/オフ)
動的か	はい
検査	なし
どのような場合に変更するか	一定期間にファイルが頻繁に変更されるため、フラッシュによる負荷がシステムの動作に悪影響を

与える場合。システムがリブートされる際に消えたり状態の一貫性がどうなっても構わないファイルは、TMPFS ファイルシステム (/tmp など) に置いた方がいいでしょう。Solaris 7 や Solaris 8 リリースが動作しているシステムでは、mount の `-noatime` オプションを使用することによって、i ノードトラフィックを減らすことができます。このオプションを使うと、ファイルがアクセスされた時に i ノードの更新が行われません。

リアルタイム処理を行うシステムでは、このオプションを無効にし、アプリケーションによってファイルの同期化を明示的に行い、一貫性を保つことを望むこともあるでしょう。

コミットレベル

変更の可能性あり

---

## プロセスのサイズのチューニング可能パラメータ

システムで使用されるプロセスの数や個々のユーザーが作成できるプロセスの数を制御する変数がいくつかあります。基本となる変数は `maxusers` です。この変数は、`max_nprocs` や `maxuprc` に割り当てられる値の基になります。

### `maxusers`

#### 説明

`maxusers` は、当初、システムがサポートできるログインユーザーの数を定義するものでした。カーネルの生成時には、さまざまなテーブルのサイズがこの設定値に基づいて決められていました。しかし、現在の Solaris リリースでは、サイズの決定は主にシステムのメモリー総量に基づいて行われます。そのため、今までの `maxusers` の使い方は、大部分が変更になっています。ただし、次のサブシステムの数は `maxusers` に基づいています。

- システムで使用できるプロセスの最大数

	<ul style="list-style-type: none"> <li>■ システムに保持される割り当て構造体の数</li> <li>■ ディレクトリ名ルックアップキャッシュ (DNLC) のサイズ</li> </ul>
データ型	符号付き整数
デフォルト	M バイト単位のメモリー総量か 2048 (小さい方)
範囲	1 から 2048
	<hr/> <p>注 - 1024 より大きい値は <code>/etc/system</code> に指定する必要があります。4096 より大きい値は、4096 に設定されます。</p> <hr/>
単位	ユーザー
動的か	いいえ。この変数に依存する変数を計算した後に <code>maxusers</code> が再び参照されることはありません。
検査	なし
どのような場合に変更するか	<p>システムによって計算されたデフォルトのユーザープロセス数が小さすぎる場合。このような状況は、システムコンソールに表示される次のメッセージでわかります。</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 10px auto;">out of processes</div> <p>デフォルトのプロセス数が大きすぎる場合は、次のようにします。</p> <ul style="list-style-type: none"> <li>■ 多量のメモリーがあり、動作しているプロセスの数が比較的少ないデータベースサーバーでは、<code>maxusers</code> のデフォルト値を少なくすることによってシステムメモリーを節約できます。</li> <li>■ 多量のメモリーがあり、動作しているプロセスがほとんどないファイルサーバーでは、この値を減らすことができます。ただし、</li> </ul>

DNLC のサイズを明示的に設定する必要があります (70ページの「ncsize」を参照)。

- 多量のメモリーがあり、動作しているプロセスがほとんどない計算サーバーでは、この値を減らすことができます。

コミットレベル

変更の可能性あり

変更履歴

詳細は、183ページの「maxusers (Solaris 7 リリース)」を参照してください。

## reserved\_procs

説明

UID が root (0) のプロセス (fsflush など) のためにプロセステーブルに予約するシステムプロセススロットの数を指定します。

データ型

符号付き整数

デフォルト

5

範囲

5 から MAXINT

単位

プロセス

動的か

いいえ。最初のパラメータ計算の後には使用されません。

検査

Solaris 8 リリースでは、どのような /etc/system 設定でも有効です。

コミットレベル

変更の可能性あり

どのような場合に変更するか

たとえば、システムの UID 0 (root) のプロセスの数を、通常の数から 10 大きくした場合、この設定をしないとユーザーレベルのプロセスを作れないような状況でも、root でシェルを起動するために必要な余裕が生まれます。

## pidmax

説明	<p>このパラメータは、取り得る最大のプロセス ID の値を指定します。これは Solaris 8 以降のリリースで有効です。</p> <p>pidmax は maxpid 変数の値を設定します。したがって、maxpid がいったん設定されると、pidmax は無視されます。maxpid は、カーネルの別のところで、最大のプロセス ID を判別したり、制約のチェックを行うために使用されます。</p> <p>/etc/system ファイルに maxpid エントリを追加して設定しようとしても、効果はありません。</p>
データ型	符号付き整数
デフォルト	30,000
範囲	266 から 999,999
単位	プロセス
動的か	いいえ。pidmax の値を設定するためにブート時だけ使用されます。
検査	値が reserved_procs の値や 999,999 と比較されます。値が reserved_procs より小さいか 999,999 より大きいと、値は 999,999 に設定されます。
暗黙的制約	max_nprocs に対して範囲の検査が行われ、max_nprocs は常にこの値以下に保たれます。
どのような場合に変更するか	システムで 30,000 を超えるプロセスをサポートするには、このパラメータの変更が必要です (他の処理も必要)。
コミットレベル	変更の可能性あり

## max\_nprocs

### 説明

システム上に作成できるプロセスの最大数。これには、システムプロセスとユーザープロセスが含まれます。/etc/system に指定する任意の値が maxuprc の計算に使用されます。

さらに、この値は、他のシステムデータ構造体のサイズを決める際にも使用されます。この変数が使用される他のデータ構造体は次のとおりです。

- ディレクトリ名ルックアップキャッシュのサイズを決めるとき (ncsize が指定されていない場合)
- UFS のディスク割り当て構造体を割り当てるとき (ndquot が指定されていない場合)
- 構成されたシステム V セマフォによって使用されるメモリの総量がシステム限度を超えていないか確認するとき
- sun4d、sun4m、Intel プラットフォーム向けのハードウェアアドレス変換のリソースを構成するとき

### データ型

符号付き整数

### デフォルト

10 + (16 × maxusers)

### 範囲

266 から maxpid の値

### 動的か

いいえ

### 検査

maxpid と比較し、それよりも大きい場合は、maxpid に設定されます。sun4d と Intel プラットフォームでは、さらにプラットフォーム固有の値と比較されます。max\_nprocs は、max\_nprocs、maxpid、プラットフォーム値のうち最も小さい値に設定されます。両プラットフォームとも、プラットフォーム値として 65,534 を使用します。

どのような場合に変更するか	このパラメータの変更は、1つのシステムで30,000を越えるプロセスを可能にするために必要となる手順の一部です。
コミットレベル	変更の可能性あり
変更履歴	詳細は、184ページの「max_nprocs (Solaris 8より前のリリース)」を参照してください。

## maxuprc

説明	1ユーザーが1つのシステム上に作成できるプロセスの最大数
データ型	符号付き整数
デフォルト	max_nprocs - reserved_procs
範囲	1 から max_nprocs - reserved_procs
単位	プロセス
動的か	いいえ
検査	max_nprocs - reserved_procs と比較され、小さい方に設定されます。
どのような場合に変更するか	1ユーザーが作成できるプロセスの数を強く制限するために、デフォルト値より小さい値を指定したい場合(システムが作成できるプロセスの数が多くて)。この限度を超えると、次の警告メッセージがコンソールかメッセージファイルに出力されます。 <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">out of per-user processes for uid N</div>
コミットレベル	変更の可能性あり

## ページング関連のチューニング可能パラメータ

Solaris 環境は、必要に応じてページングされる仮想メモリーシステムです。システムの稼働に伴ってページが必要になると、そのページがメモリーに読み込まれます。メモリーの占有率が一定のしきい値を超え、さらにメモリーの要求が続くと、ページングが発生します。ページングには、一定の変数で制御されるいくつかのレベルがあります。

一般的なページングアルゴリズムは次のとおりです。

- メモリーの不足が認識されます。ページ走査スレッドが実行され、メモリーのチェックを開始します。この際、2段階のアルゴリズムが使用されます。
  1. 使用されていないページを識別します。
  2. 一定の間隔後にもそのページが使用されていないければ、そのページを再利用の対象とみなします。

ページが変更されていると、そのページの入出力をスケジュールする要求をページアウトスレッドに出し、走査スレッドはメモリーの監視を続けます。ページアウトは、そのページをページのバッキングストアに書き込み、空リストに置くようにします。メモリーの走査では、ページの内容の区別はありません。ページは、データファイルからのものもあれば、実行可能ファイルのテキスト、データ、スタックからのものもあります。

- システムのメモリーの使用が著しくなってくるに従い、このアルゴリズムは、再利用の候補とみなすページや、ページングアルゴリズムを実行する頻度に関する基準を強化します。詳細は、54ページの「fastscan」と55ページの「slowscan」を参照してください。使用可能なメモリーが `lotsfree` と `minfree` 間の範囲に入ると、システムは、ページアウトスレッドの各呼び出しで走査するメモリー総量を `slowscan` の値から `fastscan` の値に線形に増やします。システムは、`desfree` 変数を使用して、リソースの使用や動作に関する決定数を制御します。

システムはまた、ページアウト操作を、1つのCPUの4%以内に限定しようとしません。このアルゴリズムは、`slowscan` と `fastscan` の間のメモリー量の一部を調べ、次の条件のどれかに当てはまると走査を終了します。

- メモリー不足を解消するだけのページが見つかった。
- 予定のページ数を調べた。
- 長すぎる時間が経過した。

ページアウトが走査を終了してもメモリー不足が解消しない場合は、後で別の走査が 1/4 秒間スケジュールされます。



---

**注意** - VM システムのすべてのチューニングパラメータを `/etc/system` から削除することをお勧めします。まずデフォルトの設定値で実行してから、これらのパラメータの調整が必要かどうかを判定してください。さらに、`priority_paging` を有効にしたり、`cachefree` を調整しないでください。これらはカーネルに残っていますが、現在では必要なものではありません。これらを変更すると、ページ走査が実行されたときにほとんどの場合パフォーマンスが低下します。

---

Solaris 7 5/99 リリースから、CPU とメモリーに対する動的再構成 (DR) がサポートされています。メモリーの追加や削除を伴う DR 操作があると、該当のパラメータが `/etc/system` に明示的に設定されていなければ、その値が再計算されません。`/etc/system` にパラメータが指定されている場合は、変数の値が制約に違反していない限りその値が使用されます。違反している場合は、値がリセットされます。

## lotsfree

説明	システムページングを開始する最初のトリガーです。ページ数がこのしきい値に達すると、ページキャナが立ち上がり、再利用するメモリーページを探します。
データ型	符号なし long
デフォルト	物理メモリーの 1/64 か 512K バイト (大きい方)
範囲	最小値は、512K バイトか物理メモリーの 1/64 の大きい方です。この値は、 <code>getpagesize(3C)</code> から返されるページサイズに基づくページ数で表わされます。  最大値は物理メモリーページの数です。この最大値は、物理メモリーの 30% 以内であるべきです。システムは、「検査」の項目で記述されていなければ、この範囲を強制しません。
単位	ページ

動的か	はい。ただし、メモリーに関して DR 操作が行われると、動的な変更は失われます。
検査	lotsfree が物理メモリーの総量より大きい場合、値はデフォルトにリセットされます。
暗黙的制約	cachefree が lotsfree に等しいかそれより大きく、lotsfree が desfree より大きく、desfree が minfree より大きいという関係が常に維持されなければなりません。
どのような場合に変更するか	<p>ページ要求が急激に増えるような場合には、メモリアルゴリズムが要求に対応できないことがあります。この問題を回避する 1 つの方法は、メモリーの再利用を早くに開始することです。これは、ページングシステムにいくらか余裕を与えることとなります。</p> <p>経験則によると、このパラメータは、システムが 2 ～ 3 秒で割り当てる必要がある量の 2 倍にします。このパラメータの適正值は負荷によって異なります。DBMS サーバーはおそらくデフォルトの設定値で問題なく機能しますが、ファイルシステム入出力を頻繁に行うシステムでは、このパラメータを調整する必要があります。</p> <p>負荷が比較的安定し、メモリー総量が多いシステムでは、この値を下方に調整します。指定可能な最小値は 512K バイトで、この値は <code>getpagesize(3C)</code> から返されるページサイズに基づくページ数で表わされます。</p>
コミットレベル	変更の可能性あり
<b>desfree</b>	
説明	システムで常に空であることが望まれるメモリーの総量です。
データ型	符号なし整数

デフォルト	lotsfree / 2
範囲	<p>最小値は 256K バイトか物理メモリーの 1/128 の大きい方です。この値は <code>getpagesize(3C)</code> から返されるページサイズに基づくページ数で表わされます。</p> <p>最大値は物理メモリーのページ数です。この最大値は物理メモリーの 15% 以内であるべきです。システムは、「検査」の項目で記述されていなければ、この範囲を強制しません。</p>
単位	ページ
動的か	<p>はい。ただし、メモリーの追加や削除を伴う動的再構成が行われると、その時点でこの値は、<code>/etc/system</code> ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。</p>
検査	<p><code>desfree</code> が <code>lotsfree</code> より大きい場合、<code>desfree</code> は <code>lotsfree/2</code> に設定されます。メッセージは表示されません。</p>
暗黙的制約	<p><code>cachefree</code> が <code>lotsfree</code> に等しいかそれより大きく、<code>lotsfree</code> が <code>desfree</code> より大きく、<code>desfree</code> が <code>minfree</code> より大きいという関係が常に維持されなければなりません。</p>
副次的な影響	<p>この変数の値を増やすと、いくつかの副次的な影響が現われることがあります。この値がシステムの使用可能なメモリー量に近づくかそれを超えると、次のことが起ります。</p> <ul style="list-style-type: none"> <li>■ 使用可能なメモリーが <code>desfree</code> を越えない限り、非同期の入出力要求が処理されません。したがって、<code>desfree</code> の値を増やすと、増やす前なら処理されたであろう要求が拒否されることがあります。</li> <li>■ NFS バージョン 3 の非同期書き込みが、同期書き込みとして実行されます。</li> </ul>

- スワッパーが本来より早く立ち上がり、そのスワッパーの動作が、積極的な動作をする方向に傾きます。
- システムに前もって読み込む実行可能ページの数本来よりも少なくなることがあります。この副次的な影響の結果、アプリケーションの動作が本来よりも遅くなる可能性があります。

どのような場合に変更するか

負荷が比較的安定し、メモリー総量が多いシステムでは、この値を下方に調整します。指定可能な最小値は 256K バイトで、この値は `getpagesize(3C)` から返されるページサイズに基づくページ数で表わされます。

コミットレベル

変更の可能性あり

## minfree

説明

許容できる最小限のメモリーレベル。メモリーがこの値を下回ると、システムは、ページアウト操作を正常に完了するか、複数のプロセスをメモリーから完全にスワップアウトするために必要な割り当てを行い、他の割り当て要求を拒否するかブロックします。

データ型

符号なし整数

デフォルト

`desfree / 2`

範囲

最小値は 128K バイトか物理メモリーの 1/256 の大きい方です。この値は `getpagesize(3C)` から返されるページサイズに基づくページ数で表わされます。

最大値は物理メモリーのページ数です。この最大値は物理メモリーの 7.5% 以内であるべきです。システムは、「検査」の項目で記述されていなければ、この範囲を強制しません。

単位	ページ
動的か	はい。ただし、メモリーの追加や削除を伴う動的再構成が行われると、その時点でこの値は、 <code>/etc/system</code> ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検査	<code>minfree</code> が <code>desfree</code> より大きい場合、 <code>minfree</code> は <code>desfree/2</code> に設定されます。メッセージは表示されません。
暗黙的制約	<code>cachefree</code> が <code>lotsfree</code> に等しいかそれより大きく、 <code>lotsfree</code> が <code>desfree</code> より大きく、 <code>desfree</code> が <code>minfree</code> より大きいという関係が常に維持されなければなりません。
どのような場合に変更するか	一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を下方に調整します。指定可能な最小値は 128K バイトで、この値は <code>getpagesize(3C)</code> から返されるページサイズに基づくページ数で表わされます。
コミットレベル	変更の可能性あり

## throttlefree

説明	ブロック可能なメモリー割り当ての要求を、その要求を満足できるメモリーがある場合でも休眠状態に置くメモリーレベル。
データ型	符号なし整数
デフォルト	<code>minfree</code>
範囲	最小値は 128K バイトか物理メモリーの 1/256 の大きい方です。この値は <code>getpagesize(3C)</code> から返されるページサイズに基づくページ数で表わされます。

最大値は物理メモリーのページ数です。この最大値は物理メモリーの4%以内であるべきです。システムは、「検査」の項目で記述されていなければ、この範囲を強制しません。

単位	ページ
動的か	はい。ただし、メモリーの追加や削除を伴う動的再構成が行われると、その時点でこの値は、 <code>/etc/system</code> ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検査	<code>throttlefree</code> が <code>desfree</code> よりも大きい場合、 <code>throttlefree</code> は <code>minfree</code> に設定されません。メッセージは表示されません。
暗黙的制約	<code>cachefree</code> が <code>lotsfree</code> に等しいかそれより大きく、 <code>lotsfree</code> が <code>desfree</code> より大きく、 <code>desfree</code> が <code>minfree</code> より大きいという関係が常に維持されなければなりません。
どのような場合に変更するか	一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、この値を下方に調整します。指定可能な最小値は128Kバイトで、この値は <code>getpagesize(3C)</code> から返されるページサイズに基づくページ数で表わされます。
コミットレベル	変更の可能性あり

## pageout\_reserve

説明	ページアウトスレッドやスケジューラスレッドのために予約されるページ数。使用可能なメモリーがこの値を下回ると、ページアウトやスケジューラ以外のプロセスに対するブロックしない割り当ては拒否されます。ページアウトには専用の小さなメモリープールが必要です。ペー
----	--

ジアウトは、ページをバッキングストアに書き込む入出力に必要なデータ構造体をここから割り当てます。この変数は、メモリー不足が極めて深刻な場合でもページアウト操作を行えるように、Solaris 2.6 リリースから導入されました。

データ型	符号なし整数
デフォルト	throttlefree / 2
範囲	<p>最小値は 64K バイトか物理メモリーの 1/512 の大きい方です。この値は <code>getpagesize(3C)</code> から返されるページサイズに基づくページ数で表わされます。</p> <p>最大値は物理メモリーのページ数です。この最大値は物理メモリーの 2% 以内であるべきです。システムは、「検査」の項目で記述されていない場合は、この範囲を強制しません。</p>
単位	ページ
動的か	はい。ただし、メモリーの追加や削除を伴う動的再構成が行われると、その時点でこの値は、 <code>/etc/system</code> ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検査	<code>pageout_reserve</code> が <code>throttlefree / 2</code> より大きい場合、 <code>pageout_reserve</code> は <code>throttlefree / 2</code> に設定されます。メッセージは表示されません。
暗黙的制約	<code>cachefree</code> が <code>lotsfree</code> に等しいかそれより大きく、 <code>lotsfree</code> が <code>desfree</code> より大きく、 <code>desfree</code> が <code>minfree</code> より大きく、 <code>minfree</code> が <code>throttlefree</code> に等しいかそれより大きいという関係が常に維持されなければなりません。
どのような場合に変更するか	一般にはデフォルト値で十分です。負荷が比較的安定し、メモリー総量が多いシステムでは、

この値を下方に調整します。指定可能な最小値は 64K バイトで、この値は `getpagesize(3C)` から返されるページサイズに基づくページ数で表わされます。

コミットレベル

変更の可能性あり

## cachefree

説明

Solaris 8 リリースで、ファイルシステムページのキャッシュ方法が変更されました。これらの変更には、優先度ページング機能が含まれています。

---

注 - `cachefree` と `priority_paging` の両方の設定を `/etc/system` ファイルから削除してください。

---

キャッシュ方法の変更により、ファイルシステムの動作に伴って発生する仮想メモリーシステムへの負担がほとんど解消されます。統計によると、次のような新しい動作が見られます。

- ページ再利用の数が多くなります。これは、入出力が終了すると、ページが明示的に空リストに追加されるためです。
- 空きメモリーが増えます。これは、空きメモリーのカウンタにファイルキャッシュの大部分が含まれるためです。
- 走査率が大幅に減ります。

コミットレベル

廃止または互換性がなくなる可能性あり

変更履歴

詳細は、187ページの「`cachefree` (Solaris 2.6 リリースおよび Solaris 7 リリース)」を参照してください。

## priority\_paging

説明	<p>この変数に基づき <code>lotsfree</code> の 2 倍の値が <code>cachefree</code> に設定されます。</p> <p>Solaris 8 リリースで、ファイルシステムページのキャッシュ方法が変更されました。これらの変更には、優先度ページング機能が含まれます。</p> <hr/> <p>注 - <code>cachefree</code> と <code>priority_paging</code> の設定を <code>/etc/system</code> ファイルから削除してください。</p> <hr/>
コミットレベル	廃止または互換性がなくなる可能性あり
変更履歴	詳細は、186ページの「 <code>priority_paging</code> (Solaris 2.6 リリースおよび Solaris 7 リリース)」を参照してください。

## pages\_pp\_maximum

説明	ロックされていないことをシステムが要求するページ数を定義します。ページのロック要求によって使用可能なメモリーがこの値を下回る場合は、その要求は拒否されます。
データ型	符号なし long
デフォルト	200、 <code>tune_t_minarmem + 100</code> 、[ブート時に使用可能なメモリーの 10%]、のうちで最も大きいもの
範囲	デフォルト値から物理メモリーの 20% 以内。システムは、「検査」の項目で記述されていなければ、この範囲を強制しません。
単位	ページ
動的か	はい。ただし、メモリーの追加や削除を伴う動的再構成が行われると、その時点でこの値は、 <code>/etc/system</code> ファイルに指定された値か、計算された値にリセットされます。

検査	200、 <code>tune_t_minarmem + 100</code> 、[使用可能なメモリーの 10%]、 <code>/etc/system</code> からの値の中で最も大きいもの。 <code>/etc/system</code> からの値が増やされても、メッセージは表示されません。ブート時にのみ行われます。
どのような場合に変更するか	メモリーのロック要求や、 <code>SHARE_MMU</code> フラグを指定した共有メモリーセグメントへの接続が失敗したが、使用可能なメモリーが十分ありそうな場合。32G バイトシステムで、メモリーの 10% を空き状態に維持することは過剰である場合があります。  値が大きすぎると、メモリーロック要求が不必要に失敗する可能性があります。
コミットレベル	変更の可能性あり

## tune\_t\_minarmem

説明	デッドロックを避けるために維持すべき最小限の使用可能な常駐 (スワップされない) メモリー。この値は、オペレーティングシステムのコアによって使用されるメモリー部分を予約するために使用されます。この方法で予約されたページは、OS が使用可能なメモリーの最大量を判定するときにはカウントに入れられません。
データ型	符号付き整数
デフォルト	25
範囲	1 から物理メモリー
単位	ページ
動的か	いいえ
検査	なし。値が大きいと、物理メモリーが無駄になります。

どのような場合に変更するか 一般にはデフォルト値で十分です。システムがロックアップし、問題がメモリーが利用できないことに起因していることをデバッグ情報が示している場合は、この値を増やすことを検討してください。

コミットレベル 変更の可能性あり

## fastscan

説明 メモリーが最も逼迫しているときにシステムが1秒間に調べるページの最大数

データ型 符号付き整数

デフォルト 64M バイトか物理メモリーの1/2 (小さい方)

範囲 1 から物理メモリーの1/2

単位 ページ

動的か はい。ただし、メモリーの追加や削除を伴う動的再構成が行われると、その時点でこの値は、`/etc/system` ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。

検査 最大値は、64M バイトと物理メモリーの1/2のうちで小さい方です。

どのような場合に変更するか メモリー不足のときにメモリーの走査を強化したい場合。特に、システムでメモリーの要求が急激に多くなることがある場合や、多数のファイル入出力が行われることがある場合。

コミットレベル 変更の可能性あり

## slowscan

説明	メモリーを再利用するためにシステムが 1 秒間に調べるページの最小数
データ型	符号付き整数
デフォルト	物理メモリーの 1/20 (ページ数) か 100 (小さい方)
範囲	1 から fastscan / 2
単位	ページ
動的か	はい。ただし、メモリーの追加や削除を伴う動的再構成が行われると、その時点でこの値は、 <code>/etc/system</code> ファイルに指定された値か、新しい物理メモリーの値から計算された値にリセットされます。
検査	slowscan が fastscan / 2 より大きい場合、slowscan は fastscan / 2 に設定されません。メッセージは表示されません。
どのような場合に変更するか	メモリー不足のときにメモリーの走査を強化したい場合。特に、システムでメモリーの要求が急激に多くなるときがある場合。
コミットレベル	変更の可能性あり

## min\_percent\_cpu

説明	pageout で使用できる最小の CPU の割合 (%)。この変数は、ページスキャナで使用できる最大時間を判定するための開始点として使用されます。
データ型	符号付き整数
デフォルト	4
範囲	1 から 80

単位	%
動的か	はい
検査	なし
どのような場合に変更するか	複数の CPU と多くのメモリーを備えたシステム (このようなシステムではメモリーの要求が急激に多くなるときがある) でこの値を増やすと、ページャーがメモリーの検出に使用できる時間が増えます。
コミットレベル	変更の可能性あり

## handspreadpages

説明	Solaris 環境では、2 針クロックアルゴリズムを使用して、メモリーが少なくなったときにメモリー再利用の候補となるページを探します。最初の針はメモリーに使用されていないという印を付けていきます。次の針は、最初の針の少し後から、そのページに依然として使用されていないという印が付けられているかを調べます。使用されていなければ、そのページが再利用の対象になります。最初の針と次の針の間隔が <code>handspreadpages</code> です。
データ型	符号なし long
デフォルト	<code>fastscan</code>
範囲	1 からシステムの物理メモリーの最大ページ数
単位	ページ
動的か	はい。このパラメータを変更する場合、カーネル変数 <code>reset_hands</code> もゼロ以外の値に設定する必要があります。 <code>handspreadpages</code> の新しい値がいったん認識されると、 <code>reset_hands</code> はゼロに設定されます。

検査	物理メモリーの総量と <code>handsreadpages</code> の値のうちで小さい方に設定されます。
どのような場合に変更するか	再利用される前にページがメモリー上に存在するかもしれない時間を増やしたい場合。この値を増やすと2つの段階の間の時間が長くなるため、ページが再利用されるまでの時間が増えます。
コミットレベル	変更の可能性あり

## `pages_before_pager`

説明	入出力が終了したら、再利用に備えてページを保存する代わりに、それらのページをただちに解放することを示すシステムしきい値の一部です。このしきい値は <code>lotsfree + pages_before_pager</code> です。さらに、NFS 環境も、メモリーが不足するとこのしきい値を使用して非同期の活動を減らします。
データ型	符号付き整数
デフォルト	200
範囲	1 から物理メモリーの総量
単位	ページ
動的か	いいえ
検査	なし
どのような場合に変更するか	<p>入出力の大半が、一度だけ読み取られたり書き込まれたりするだけで、再び参照されることがないページに対するものである場合。この変数を大きなメモリーの値に設定すると、ページは空リストに追加され続けます。</p> <p>システムが突発的に深刻なメモリー不足に陥る可能性がある場合。この値を増やすと、メモリー不足に対する備えがそれだけ強化されます。</p>

コミットレベル

変更の可能性あり

## maxpgio

説明

ページングシステムがキューに入れることができるページ入出力要求の最大数。ページングシステムは、実際に使用する最大数を計算するために、この数字を 4 で割ります。これは、要求の数を制限する他に、プロセスのスワッピングを制御するためにも使用されます。

データ型

符号付き整数

デフォルト

40

範囲

1 から 1024

単位

入出力

動的か

いいえ

検査

なし

暗黙的制約

ページャからの入出力要求の最大数は、要求バッファのリストのサイズによって制限されます。現在のサイズは 256 です。

どのような場合に変更するか

システムが突発的に深刻なメモリー不足に陥る可能性がある場合。この値を増やすと、複数のスワップデバイスが構成されているか、スワップデバイスがストライプデバイスである場合、メモリー不足の解消が早くなります。

コミットレベル

変更の可能性あり

---

## スワッピング関連の変数

Solaris 環境のスワッピングは、swapfs 擬似ファイルシステムによって行われます。スワップデバイスの空間と物理メモリーを合わせたものが、匿名メモリーのバッキ

ングストアを維持するために利用可能な空間プールとして扱われます。システムは、バッキングストアとして最初にディスクデバイスから空間を割り当てようとし、その次に物理メモリーを使用します。swapfs がバッキングストアとしてシステムメモリーを使用しなければならない場合は、swapfs によるメモリーの使いすぎによってシステムがデッドロックに陥ることがないように制約が課せられます。

## swapfs\_reserve

説明	システム (UID = 0) プロセスによる使用のために予約するシステムメモリー量です。
データ型	符号なし long
デフォルト	4M バイトか物理メモリーの 1/16 (小さい方)
範囲	<p>最小値は 4M バイトか物理メモリーの 1/16 の小さい方です。この値は <code>getpagesize(3C)</code> から返されるページサイズに基づくページ数で表わされます。</p> <p>最大値は物理メモリーのページ数です。最大値は、物理メモリーの 10% 以内であるべきです。システムは、「検査」の項目で記述されていなければ、この範囲を強制しません。</p>
単位	ページ
動的か	いいえ
検査	なし
どのような場合に変更するか	変更は一般には必要ありません。ソフトウェアプロバイダからの推奨があったり、スワップ空間が取得できないためにシステムプロセスが終了してしまう場合だけ変更します。しかし、それより良い解決策は、物理メモリーかスワップデバイスをシステムに追加することです。
コミットレベル	変更の可能性あり

## swapfs\_minfree

説明	システムのその他の部分のために空き状態に保つことが望ましい物理メモリーの量。プロセスのスワップ空間としてメモリーを予約しようするときに、それによって使用可能なメモリーがこの値を下回るおそれがあるとシステムが判断する場合、この要求は拒否されます。この方法で予約されたページは、カーネルやユーザーレベルプロセスによってロックダウンされた割り当てに対してのみ使用できます。
データ型	符号なし long
デフォルト	2M バイトか物理メモリーの 1/8 (大きい方)
範囲	1 から物理メモリーのページ数
単位	ページ
動的か	いいえ
検査	なし
どのような場合に変更するか	システムに使用可能なメモリーがあるのにスワップ空間が得られないためにプロセスが失敗する場合
コミットレベル	変更の可能性あり

---

## 一般的なカーネル変数

### noexec\_user\_stack

説明	スタックを実行不能と印付けすることを可能にします。これは、バッファオーバーフロー攻撃を困難にする上で有効です。  64 ビットカーネルが動作している Solaris システムは、デフォルトで、すべての 64 ビットアプリ
----	--

リケーションのスタックを実行不能にします。  
64 ビットカーネルおよび 32 ビットカーネルが動作するシステムで 32 ビットアプリケーションのスタックを実行不能にするには、この変数の設定が必要です。

---

注 - この変数は Solaris 2.6 以降が動作しているすべてのシステムに存在しますが、sun4u、sun4m、sun4d アーキテクチャでのみ有効です。

---

データ型	符号付き整数
デフォルト	0 (無効)
範囲	0 (無効) か 1 (有効)
単位	切り替え (オン/オフ)
動的か	はい。ただし、すでに実行中のプロセスに対しては無効です。値が設定された後に起動されたプロセスに対してのみ有効です。
検査	なし
どのような場合に変更するか	アプリケーションが、mprotect(2) を使用してスタックを実行可能にすることなくスタックに実行可能コードを意図的に置いている場合を除き、この変数は常に有効にすべきです。
コミットレベル	変更の可能性あり
変更履歴	詳細は、188ページの「noexec_user_stack (Solaris 2.6 リリースおよび Solaris 7 リリース)」を参照してください。

---

## カーネルメモリアロケータ

Solaris カーネルメモリアロケータは、カーネル内の各エンティティに使用するメモリーの断片を配分します。アロケータは、そのクライアントが使用するさまざま

なサイズのキャッシュを作成します。一方、クライアントは、特定サイズの構造体の割り当てのためなど、クライアントが使用するキャッシュの作成をアロケータに要求できます。アロケータが管理する各キャッシュに関する統計は、`kstat -c kmem_cache` コマンドで表示できます。特殊化されたキャッシュを参照するには、`crash(1M)` コマンドで `kmastat` 演算子を使用します。

メモリーが壊されたために、システムがパニックになることがまれにあります。カーネルメモリーアロケータは、アロケータについての情報を収集するとともに、バッファに対してさまざまな整合性チェックを行うデバッグインタフェースをサポートします。整合性チェックは、エラーが実際に起こった場所の近くでエラーを検出するのに役立ちます。収集された情報は、サポート担当者がパニックの原因を特定する上で追加のデータを提供してくれます。

フラグを使用すると、システム操作で余分なオーバーヘッドと余分なメモリーの使用が発生します。したがって、フラグの使用は、メモリーの損傷が疑われるときだけに限るべきです。

## `kmem_flags`

### 説明

Solaris カーネルメモリーアロケータには、Solaris 環境の内部的開発サイクル中に頻繁に使用されたさまざまなデバッグやテストのオプションがあります。Solaris 2.5 より前のリリースでは、これらのオプションを、リリースされた Solaris バージョンで使用することはできませんでした。しかし、Solaris 2.5 リリースからは、これらのオプションのサブセットが利用でき、`kmem_flags` 変数で制御されます。`kmem_flags` 変数を設定するには、`kadb` をブートし、その後カーネルを開始する前にこの変数を設定しました。カーネルメモリーアロケータのインスタンス化と `/etc/system` の解析のタイミングの問題のため、これらのフラグは Solaris 8 より前のリリースでは `/etc/system` ファイルに設定できませんでした。

以下に、サポートされる 5 つのフラグの設定について説明します。

表 2-1 kmem\_flags の設定値

フラグ	設定	説明
AUDIT	0x1	アロケータは、自身の活動の最近の履歴が入ったログを維持します。ログされる項目の数は、CONTENTS も設定されているかどうかによって異なります。ログのサイズは固定であるため、空間が足りなくなると、古いレコードから順に再使用されます。
TEST	0x2	アロケータは解放されたメモリにパターンを書き込み、そのバッファを次に割り当てるときに、そのパターンが変更されていないことをチェックします。バッファの一部が変更されている場合は、そのバッファを前に割り当て、開放したエンティティがそのメモリを使用した可能性が強いことがわかります。上書きが検知されると、システムがパニックになります。
REDZONE	0x4	アロケータは要求されたバッファの終りに余分のメモリを割り当て、そのメモリに特殊なパターンを挿入します。そして、バッファが解放されたら、パターンをチェックして、データがバッファの終りより後ろに書き込まれていないか調べます。上書きが検知されると、カーネルがパニックになります。

表 2-1 kmem\_flags の設定値 続く

フラグ	設定	説明
CONTENTS	0x8	<p>アロケータは、バッファが解放されると、バッファの内容を 256 バイトまでログします。同時に AUDIT も設定する必要があります。</p> <p>これらのフラグの数値は、論理和をとる (OR) ことができ、Solaris 8 リリースでは /etc/system ファイルで設定し、それより前のリリースでは kadb をブートして、カーネルを開始する前にこれらのフラグを設定します。</p>
LITE	0x100	<p>バッファの割り当てや解放で最小限の妥当性検査を行います。このフラグが有効になっていると、アロケータは、レッドゾーンが書き込まれていないことや、解放されたバッファが再び解放されていないこと、解放されるバッファのサイズが割り当てられたものと同じであることをチェックします。このフラグは、Solaris 7 3/99 リリースから使用できるようになりました。このフラグは他のフラグと併用しないでください。</p>

データ型	符号付き整数
デフォルト	0 (無効)
範囲	0 (無効)、1 ~ 15、256 (0x100)
動的か	はい。実行時の変更は、新しいカーネルメモリーキャッシュだけに有効です。システムの初期設定後に新しいキャッシュを作成することはまれです。
検査	なし
どのような場合に変更するか	メモリーの損傷が疑われる場合

## 一般的なドライバ

### moddebug

説明	モジュールロードプロセスのさまざまなステップに関するメッセージの表示を制御する変数です。
データ型	符号付き整数
デフォルト	0 (メッセージを表示しない)
範囲	以下の値が有用です。

- 0x80000000 - [un] loading... メッセージを出力します。モジュールがロードされるたびに、次のようなメッセージがコンソールと `/var/adm/messages` ファイルに出力されます。

```
Nov 5 16:12:28 sys genunix: [ID 943528 kern.notice] load 'sched/TS_DPTBL'
id 9 loaded @ 0x10126438/0x10438dd8 size 132/2064
Nov 5 16:12:28 sys genunix: [ID 131579 kern.notice] installing TS_DPTBL,
module id 9.
```

- 0x40000000 - 詳細なエラーメッセージを出力します。モジュールがロードされるたびに、次のようなメッセージがコンソールと `/var/adm/messages` ファイルに出力されます。

```
Nov 5 16:16:50 sys krtld: [ID 284770 kern.notice] kobj_open: can't open
/platform/SUNW,Ultra-1/kernel/sched/TS_DPTBL
Nov 5 16:16:50 sys krtld: [ID 284770 kern.notice] kobj_open: can't open
/platform/sun4u/kernel/sched/TS_DPTBL
Nov 5 16:16:50 sys krtld: [ID 797908 kern.notice] kobj_open: '/kernel/
sch...
Nov 5 16:16:50 sys krtld: [ID 605504 kern.notice] descr = 0x2a
Nov 5 16:16:50 sys krtld: [ID 642728 kern.notice] kobj_read_file: size=34,
```

(続く)

```

Nov 5 16:16:50 sys krtld: [ID 217760 kern.notice] offset=0
Nov 5 16:16:50 sys krtld: [ID 136382 kern.notice] kobj_read: req 8192 bytes,
Nov 5 16:16:50 sys krtld: [ID 295989 kern.notice] got 4224
Nov 5 16:16:50 sys krtld: [ID 426732 kern.notice] read 1080 bytes
Nov 5 16:16:50 sys krtld: [ID 720464 kern.notice] copying 34 bytes
Nov 5 16:16:50 sys krtld: [ID 234587 kern.notice] count = 34
[33 lines elided]
Nov 5 16:16:50 sys genunix: [ID 943528 kern.notice] load 'sched/TS_DPTBL'
id 9 loaded @ 0x10126438/0x10438dd8 size 132/2064
Nov 5 16:16:50 sys genunix: [ID 131579 kern.notice] installing TS_DPTBL,
module id 9.
Nov 5 16:16:50 sys genunix: [ID 324367 kern.notice] init 'sched/TS_DPTBL'
id 9 loaded @ 0x10126438/0x10438dd8 size 132/2064

```

- 0x20000000 - より詳細なメッセージを出力します。「detailed error message」フラグがシステムのブート時に出力する以上の情報は出力されませんが、モジュールがアンロードされる時に、モジュールの解放に関する追加の情報が出力されます。

これらの値は足し合わせて指定できます。

動的か	はい
検査	なし
どのような場合に変更するか	期待通りにモジュールがロードされない場合や、モジュールのロード中にシステムがハングしている疑いがある場合。「print detailed messages」を設定すると、多数のメッセージがコンソールに書き込まれるため、システムのブートがかなり遅くなることに留意してください。
コミットレベル	変更の可能性あり

## 一般的な入出力

### maxphys

説明	物理入出力要求の最大サイズ。要求がこのサイズより大きいと、ドライバはこの要求を maxphys サイズの断片に分割します。個々のファイルシステムは独立して制限値を持つことが可能で、実際に独立した制限値を持ちます。
データ型	符号付き整数
デフォルト	126,976 (sun4m および sun4d)、131,072 (sun4u)、57,344 (Intel)。ワイド転送をサポートする sd ドライバは 1,048,576 を使用します。ssd ドライバはデフォルトで 1,048,576 を使用します。
範囲	マシン固有のページサイズから MAXINT
単位	バイト
動的か	はい。しかし、多くのファイルシステムでは、ファイルシステムがマウントされるときに、この値がマウントポイントごとのデータ構造体に設定されます。ドライバによっては、デバイスがドライバ固有のデータ構造体に設定されるときに、この値が設定されます。
検査	なし
どのような場合に変更するか	raw デバイスに対する入出力を大きな断片で行う場合。ただし、OLTP 操作を伴う DBMS では小さいサイズの入出力が頻繁に行われるため、maxphys を変更しても性能の向上は望めません。  UFS ファイルシステムに対して、64K バイトを超える大量のデータの読み取りや書き込みを一度に行う入出力を行う場合。ファイルシステム

は、連続性が増すように最適化すべきです。たとえば、シリンダグループのサイズを大きくし、シリンダグループ当たりの i ノードの数を減らす、などです。UFS では、転送する最大の入出力サイズは 1M バイトに制限されます。

コミットレベル

変更の可能性あり

## `rlim_fd_max`

説明

1 つのプロセスがオープンできるファイル記述子の「強い」限度。この制限を変更するには、スーパーユーザー特権が必要です。

データ型

符号付き整数

デフォルト

1024

範囲

1 から MAXINT

単位

ファイル記述子

動的か

いいえ

検査

なし

どのような場合に変更するか

1 プロセス当たりの最大オープンファイル数が十分でない場合。ただし、システムには他の制限もあるため、ファイル記述子の数を増やすことが必ずしも有用とは限らないことに留意してください。

- 標準入出力を使用する 32 ビットプログラムでは、256 ファイル記述子に制限されます。標準入出力を使用する 64 ビットプログラムでは、20 億の記述子まで使用できます。
- `select(3C)` では、デフォルトで、`fd_set` によって 1024 記述子に限定されます。ただし、Solaris 7 リリース以降では、32 ビットアプリケーションコードをより大きな `fd_set`

サイズ (65,536 以下) で再コンパイルできます。64 ビットアプリケーションの `fd_set` サイズは 65,536 で、変更することはできません。

システム全体に対してこれを変更する別の方法として `plimit(1)` コマンドがあります。`plimit` を使用して親となるプロセスの限度を変更すると、すべての子プロセスがその限度を継承します。これは `inetd` などのデーモンに便利です。

コミットレベル

変更の可能性あり

## `rlim_fd_cur`

説明

1つのプロセスがオープンできるファイル記述子数の「弱い」限度。プロセスは、自身のファイル記述子数限度を、`rlim_fd_max` で定義される「強い」限度の範囲内で調整できます。それには、`setrlimit()` 呼び出しを使用するか、プロセスを実行するシェルで `limit` コマンドを実行します。「強い」限度の範囲内で値を調整する場合には、スーパーユーザー特権は必要ありません。

データ型

符号付き整数

デフォルト

256

範囲

1 から `MAXINT`

単位

ファイル記述子

動的か

いいえ

検査

`rlim_fd_max` と比較して、`rlim_fd_cur` が `rlim_fd_max` より大きい場合、`rlim_fd_cur` は `rlim_fd_max` にリセットされます。

どのような場合に変更するか

1 プロセス当たりのデフォルトのオープンファイル数が十分でない場合。この値を増やす意味

は、プログラムで `setrlimit(2)` を使用して自身で利用できる最大ファイル記述子数を増やす必要がなくなる点にあります。

コミットレベル

変更の可能性あり

変更履歴

詳細は、189ページの「`rlim_fd_cur` (Solaris 7 リリースおよびそれより前のリリース)」を参照してください。

---

## 一般的なファイルシステム

`ncsize`

説明

ディレクトリ名ルックアップキャッシュ (DNLC) のエントリ数。このパラメータは、UFS や NFS が、解決されたパス名の要素をキャッシュするときに使用します。

Solaris 8 6/00 リリースから、DNLC には、ネガティブルックアップ情報もキャッシュされません。つまり、名前がキャッシュにない場合は、その名前がキャッシュされます。

データ型

符号付き整数

デフォルト

$4 \times (v.v\_proc + maxusers) + 320$

範囲

0 から MAXINT

単位

DNLC のエントリ

動的か

いいえ

検査

しない。値を増やすと、ファイルシステムのアンマウントに必要な時間が増えます。これは、アンマウントプロセスでそのファイルシステムのエントリをキャッシュから削除する必要があるためです。

どのような場合に変更するか

Solaris 8 6/00 より前のリリースでは、キャッシュが小さすぎるかどうかを判定するのは困難でした。しかし、`kstat -n ncstats` から返されるエントリ数を調べることにより、これを推定することが可能になりました。システムの負荷やファイルアクセスのパターンに較べこの値が大ききように思われる場合は、DNLC のサイズに原因があるかもしれません。

Solaris 8 6/00 リリース以降では、`kstat -n dnlcstats` を使用して、DNLC が小さすぎるために DNLC からエントリが削除されたことを知ることができます。`pick_heuristic` と `pick_last` の合計は、キャッシュが小さすぎるために再利用されたエントリ (そうでなければ有効であったはずのエントリ) の数を表します。

`ncsize` の値が大きすぎると、システムに直接的な影響があることに留意してください。システムは、`ncsize` の値に基づいて DNLC の一連のデータ構造体を割り当てるからです。32 ビットカーネルが動作しているシステムは `ncsize` に 36 バイトの構造体を、64 ビットカーネルが動作しているシステムは `ncsize` に 64 バイトの構造体をそれぞれ割り当てます。さらに、`ufs_inode` や `nfs:nfs_rnode` が明示的に設定されていない場合は、この値が UFS や NFS にも影響を与えます。

コミットレベル

変更の可能性あり

`rstchown`

説明

`chown(2)` システム呼び出しの POSIX 挙動が有効かどうかを示します。POSIX 挙動は次のとおりです。

- プロセスは、UID 0 で動作していない限りファイルの所有者を変更できない。

- プロセスは、UID 0 で動作していない限り、ファイルのグループ所有権を、現在メンバーになっていないグループに変更することはできない。

データ型	符号付き整数
デフォルト	1 (POSIX 挙動が使用されている)
範囲	0 = POSIX 挙動が有効ではない、1 = POSIX 挙動が使用されている
単位	切り替え (オン/オフ)
動的か	はい
検査	なし
どのような場合に変更するか	POSIX 挙動が適切でない場合。POSIX 挙動をオフにすると、さまざまなセキュリティホールの可能性が出てくる点に留意してください。さらに、ユーザーがファイルの所有権を別のユーザーに変更する可能性も生じます。その場合は、新たに所有者になったユーザーかシステム管理者の介入なしにはそのファイルの所有権を元に戻せません。
コミットレベル	廃止または互換性がなくなる可能性あり

## segkpsize

説明	利用できるページング可能なカーネルのメモリー量を指定します。このメモリーは主にカーネルスレッドのスタックに使用されます。この値を増やすと、スレッドの数を増やさないのであれば個々のスレッドでより大きなスタックが使用でき、あるいはより多くのスレッドを使用できるようになります。このパラメータは、64 ビットカーネルが動作しているシステムのみで設定できます。64 ビットカーネルが動作し
----	--

	ているシステムは、デフォルトで、24K バイトのスタックサイズを使用します。
データ型	符号なし long
デフォルト	64 ビットカーネルの場合は 2G バイト、32 ビットカーネルの場合は 512M バイト
範囲	64 ビットカーネルの場合は 512M バイトから 24G バイト、32 ビットカーネルの場合は 512M バイト
単位	M バイト
動的か	いいえ
検査	<p>値が最小値と最大値 (64 ビットシステムの場合は 512M バイトと 24G バイト) と比較され、この範囲にない場合、2G バイトにリセットされ、それを表すメッセージが表示されます。</p> <p>キャッシュの作成で実際に使用されるサイズは、制約チェックの後で <code>segkpsize</code> に指定されている値か、物理メモリーの 50% のうち、小さい方です。</p>
どのような場合に変更するか	システムで多数のプロセスをサポートする場合は、このステップが必要です (他のステップも必要)。物理メモリーが 1G バイト以上あると想定すると、デフォルトサイズの 2G バイトで、87,000 以上のカーネルスレッドに対し 24K バイトのスタックを作成できます。64 ビットカーネルのスタックサイズは、プロセスが 32 ビットプロセスでも 64 ビットプロセスでも同じです。これより大きな数が必要な場合は、物理メモリーが十分にあれば <code>segkpsize</code> を増やすことができます。
コミットレベル	変更の可能性あり
変更履歴	詳細は、189ページの「 <code>segkpsize</code> (Solaris 7 リリースおよびそれより前のリリース)」を参照してください。

## dnlc\_dir\_enable

説明	大きなディレクトリキャッシュを可能にします。
データ型	符号なし整数
デフォルト	1 (有効)
範囲	0 (無効) か 1 (有効)
動的か	はい。しかし、このチューニング可能パラメータを動的に変更してはいけません。これが無効になっていれば有効に、有効になっていれば無効にすることはできますが、このパラメータを有効にし、無効にし、再び有効にすると、ディレクトリキャッシュが最新の状態を表さないことがあります。
検査	なし
どのような場合に変更するか	ディレクトリキャッシュ機能に既知の問題はありませんが、何らかの問題が発生した場合は、dnlc_dir_enable を 0 に設定してキャッシュを無効にしてください。
コミットレベル	変更の可能性あり

## dnlc\_dir\_min\_size

説明	1つのディレクトリに対してキャッシュする前の最小のエントリ数
データ型	符号なし整数
デフォルト	40
範囲	0 から MAXUINT (無制限)
単位	
動的か	はい。いつでも変更できます。
検査	なし

どのような場合に変更するか	小さいディレクトリのキャッシュにおいてパフォーマンスに問題がある場合は、 <code>dnlc_dir_min_size</code> を増やします。個々のファイルシステムに、キャッシングディレクトリの独自の範囲限度があることもある点に留意してください。たとえば、UFSではディレクトリの最小は <code>ufs_min_dir_cache</code> バイトです(1 エントリあたり 16 バイトとして、およそ 1024 エントリ)。
コミットレベル	変更の可能性あり

## `dnlc_dir_max_size`

説明	1つのディレクトリに対してキャッシュする最大のエントリ数
データ型	符号なし整数
デフォルト	MAXUINT (無制限)
範囲	0 から MAXUINT
動的か	はい。いつでも変更できます。
検査	なし
どのような場合に変更するか	大きなディレクトリにおいてパフォーマンスに問題がある場合は、 <code>dnlc_dir_max_size</code> を減らします。
コミットレベル	変更の可能性あり

---

## UFS

### `bufhwm`

説明	入出力バッファのキャッシュに使用するメモリー量の最大値。バッファは、ファイルシステ
----	---

ムのメタデータ (スーパーブロック、iノード、間接ブロック、ディレクトリ) の書き込みに使用されます。バッファは、その量が `bufhwm` に達するまで必要に応じて割り当てられます。 `bufhwm` に達すると、バッファを再利用して要求を満たします。

歴史的経緯により、このパラメータには `ufs:` 接頭辞は必要ありません。

データ型	符号付き整数
デフォルト	物理メモリーの 2%
範囲	80K バイトから物理メモリーの 20%
単位	K バイト
動的か	いいえ。この値は、ハッシュバケットサイズの計算に使用された後、データ構造体に格納されます。このデータ構造体は、バッファの割り当てや解放に応じて、そのフィールドの値を調整します。動作しているシステムのロック手順に従わずにこの値を調整すると、正しくない動作を招くおそれがあります。
検査	<code>bufhwm</code> が 80K バイトより小さいか、 <code>bufhwm</code> が、物理メモリーの 20% もしくは現在のカーネルヒープ値の 2 倍のうち小さい方より大きい場合、 <code>bufhwm</code> は、物理メモリーの 20% か現在のカーネルヒープ値の 2 倍のうち小さい方にリセットされます。その場合は、次のメッセージがシステムコンソールと <code>/var/adm/messages</code> ファイルに出力されます。

```
"binit: bufhwm out of range (value attempted). Using N."
```

「value attempted」は、`/etc/system` で指定された値か `kadb -d` の使用による値を表します。「N」は、使用可能なシステムメモリーに基づいてシステムが計算した値です。

どのような場合に変更するか

バッファは必要が生じた時にのみ割り当てられるため、デフォルト設定によるオーバーヘッドは、考えられる最大のバッファ数を処理するために必要になる制御構造体が割り当てられることです。これらの構造体は、32 ビットカーネルでは想定されるバッファ当たり 52 バイト、64 ビットカーネルでは想定されるバッファ当たり 104 バイトを消費します。512M バイトの 64 ビットカーネルでは、この構造体は 104\*10144 バイト (約 1M バイト) を消費します。ヘッダーの割り当てでは、バッファサイズは 1K バイトであると想定します。ただし、ほとんどの場合、バッファサイズはこれより大きいのが普通です。

バッファプール内でまだ割り当てられていないメモリー量を知るには、カーネルデバッグを使用して、カーネルの `bfreelist` 構造体を調べます。この構造体の `bufsize` フィールドが、残っている可能性があるメモリーのバイト数です。`mdb` を使用して `buf` マクロでこのフィールドを表示するには、次のようにします。

```
# mdb -k
Loading modules: [ unix krtld genunix ip nfs ipc ]
> bfreelist$<buf
bfreelist:
[ 省略 ]
bfreelist + 0x78: bufsize    [ 省略 ]
                        75734016
```

6G バイトのメモリーを持つこのシステムの `bufhwm` は 122277 です。要求される実際のバッファサイズは一般に 1K バイトより大きいため、使用されるヘッダー構造体の数を直接知ることはできません。ただし、一部の領域は、このシステムに割り当てられた制御構造体からうまく回収されることがあります。

512M バイトシステム上でこの構造体が、10144K バイトのうち 4K バイトだけがまだ割り当てられ

ていないことを示したとします。また、`kstat -n biostats` で `biostats` の `kstat` を調べると、このシステムでは、`buffer_cache_hits` と `buffer_cache_lookups` の割合も適切であることが分かったとします。これらの情報は、このシステムのデフォルト設定であることを示します。

コミットレベル

変更の可能性あり

## ndquot

説明

UFS ファイルシステムに割り当てるべき割り当て (`quota`) 構造体の数。このパラメータは、1 つまたは複数の UFS ファイルで割り当てが有効になっているときだけ適用されます。歴史的経緯により、`ufs:` 接頭辞は必要ありません。

データ型

符号付き整数

デフォルト

$((\text{maxusers} \times 40) / 4) + \text{max\_nprocs}$

範囲

0 から MAXINT

単位

割り当て構造体

動的か

いいえ

検査

なし。値が大きすぎると、システムがハングします。

どのような場合に変更するか

デフォルトの割り当て構造体数では十分でない場合。このような状況は、コンソールやメッセージログに出力される次のメッセージから判別できます。

```
dquot table full
```

コミットレベル

変更の可能性あり

## ufs\_ninode

### 説明

メモリーに保持する *i* ノードの数。*i* ノードは、ファイルシステム単位ではなく、全体としてキャッシュされます (UFS の場合)。

この場合のキーとなる変数は `ufs_ninode` です。このパラメータを使用して、*i* ノードキャッシュの処理に関係する 2 つのキーとなる境界値が計算されます。高位境界値は  $\text{ufs\_ninode} / 2$ 、下位境界値は  $\text{ufs\_ninode} / 4$  で計算されます。

システムが *i* ノードの処理を終わると、次のどちらかが起こる可能性があります。

1. *i* ノードによって参照されるファイルがもはやシステムにないため、その *i* ノードが削除される。*i* ノードが削除されると、その空間は *i* ノードキャッシュに戻され、別の *i* ノード (ディスクから読み込まれるか、新規ファイル用に作成されるもの) 用に使用されます。
2. ファイルは存在するが、実行プロセスに参照されていない。*i* ノードは遊休キューに入れます。参照されていたページはメモリーに残ります。

*i* ノードを遊休化する場合、カーネルはこの遊休化処理を一定の時期まで先送りします。ファイルシステムがロギングファイルシステムの場合も、カーネルは *i* ノードの削除を先送りします。これを行うのは 2 つのカーネルスレッドです。それぞれのスレッドが一方のキューを処理します。

先送りされていた処理が終わると、システムはその *i* ノードを削除キューか遊休キューに入れます。それぞれのキューには、そのキューを処理できるスレッドがあります。*i* ノードがキューに入れられると、キューの占有率が下位境界値と比較され、占有率が下位境界値を超えている

と、そのキューに関連するスレッドが起こされます。起こされるとスレッドはキューを調べ、i ノードに結びつけられたページがあればディスクに書き出し、i ノードを解放します。スレッドは、起こされた時にキューにあった i ノードの 50% を削除すると停止します。

遊休スレッドの処理が負荷に追いつかない場合は、2 つめの機構が使用されます。システムは、vnode を見つける必要があると、ufs\_vget ルーチンを実行します。vget は「最初に」遊休キューの長さを調べ、長さが高位境界値を超えていると、遊休キューから 2 つの i ノード取り出し、遊休化します (ページをフラッシュし、i ノードを解放する)。vget は、自身が使用する i ノードを取得する「前に」これを行います。

システムは、コア内にページがない i ノードを遊休リストの先頭に置き、ページがある i ノードを遊休リストの終わりに置くことによって最適化を図ります。しかし、リストの順序に関し、それ以外の処理は行いません。i ノードは常に遊休キューの先頭から削除されます。

i ノード全体がキューから削除されるのは、同期 (sync)、アンマウント、または再マウントが行われるときだけです。

歴史的経緯により、このパラメータには ufs: 接頭辞は必要ありません。

データ型	符号付き整数
デフォルト	ncsize
範囲	0 から MAXINT
単位	i ノード
動的か	はい

検査	ufs_ninode が 0 以下の場合、この値は ncsiz に設定されます。
どのような場合に変更するか	デフォルトの i ノード数では足りない場合。kstat -n inode_cache で報告される maxsize reached フィールドが kstat の maxsize フィールドより大きい場合は、ufs_ninode の値が小さすぎる可能性があります。ただし、先に述べた i ノードの遊休化が過度に発生するのも問題です。  このような状況を知るには、kstat -n inode_cache を使用して inode_cache kstat を調べます。thread idles はバックグラウンドスレッドが遊休化した i ノード数を、vget idles は i ノードを使用する前の要求プロセスによる遊休数をそれぞれ表しています。
コミットレベル	変更の可能性あり

## ufs:ufs\_WRITES

説明	ufs_WRITES がゼロ以外の場合、ファイルに書き込むべき未処理のバイト数が調べられます。次に、ufs_HW を参照し、書き込みを行うべきか、未処理のバイト数が ufs_LW になるまで書き込みを延期すべきかが判定されます。未処理のバイト数のトータルはファイルごとに管理されるため、あるファイルの未処理のバイト数が限度を超えても、それが他のファイルに影響を与えることはありません。
データ型	符号付き整数
デフォルト	1 (有効)
範囲	0 (無効) か 1 (有効)
単位	切り替え (オン/オフ)

動的か	はい
検査	なし
どのような場合に変更するか	UFS の書き込みスロットル (抑制) を全体的にオフにしたい場合。十分な入出力能力がない場合は、このパラメータを無効にすると、ディスクに対するサービスキューが長くなるおそれがあります。
コミットレベル	変更の可能性あり

## ufs:ufs\_LW と ufs:ufs\_HW

説明	<p>ufs_HW は 1 つのファイルに対する未処理のバイト数の境界値です。未処理のバイト数がこの値を上回り、ufs_WRITES が設定されていると、書き込みは延期されます。書き込みの延期は、書き込みを行うスレッドを、条件変数で眠らせることで行われます。</p> <p>ufs_LW は 1 つのファイルに対する未処理のバイト数の境界値です。これを下回ると、他の処理が休眠状態となっている原因の条件変数が切り替えられます。書き込みが終了し、バイト数が ufs_LW を下回ると、条件変数が切り替わり、それによってその変数で待機しているすべてのスレッドが立ち上がり、それぞれの書き込みを行おうとします。</p>
データ型	符号付き整数
デフォルト	256 × 1024 (ufs_LW) と 384 × 1024 (ufs_HW)
範囲	0 から MAXINT
単位	バイト
動的か	はい
検査	なし

暗黙的制約	ufs_LW と ufs_HW は、ufs_WRITES がゼロでないときだけ意味があります。不要な問題を避けるために、ufs_HW と ufs_LW は共に変更する必要があります。たとえば、ufs_LW と ufs_HW が近すぎると、複数のスレッドが立ち上がっても、いずれも書き込みを実行できないことがあります。あるいは、ufs_LW と ufs_HW が離れすぎていると、複数のスレッドが必要以上に待たされることがあります。
どのような場合に変更するか	ファイルシステムがストライプ化ボリュームから構成されている場合は、これらの値の変更を検討します。使用可能な合計帯域幅が ufs_HW の現在の値を簡単に超える可能性があります。残念ながら、この値はファイルシステムごとに設定されるものではありません。  ufs_throttles が普通の値ではない場合。現在、ufs_throttles はカーネルデバッグでのみアクセスできます。
コミットレベル	変更の可能性あり

---

## TMPFS

### tmpfs:tmpfs\_maxkmem

説明	TMPFS が自身のデータ構造体 (tmpnode と ディレクトリエントリ) に使用できるカーネルメモリーの最大量
データ型	符号なし long
デフォルト	
範囲	1 ページのバイト数 (UltraSPARC™ システムの場合は 8192、その他のシステムの場合は 4096) から、TMPFS が最初に使用されたときに存在するカーネルメモリーの 25%

単位	バイト
動的か	はい
検査	なし
どのような場合に変更するか	次のメッセージがコンソールやメッセージファイルに出力される場合には、値を増やすします。 <pre>tmp_memalloc: tmpfs over memory limit</pre>
コミットレベル	変更の可能性あり

## tmpfs:tmpfs\_minfree

説明	TMPFS がシステムの他の部分のために保持するスワップ空間の最小量
データ型	符号付き long
デフォルト	256
範囲	0 からスワップ空間サイズの最大値
単位	バイト
動的か	はい
検査	なし
どのような場合に変更するか	TMPFS が大量に使用されるシステムで適度なスワップ空間を維持するために、この値を増やすことができます。次のメッセージがコンソールやシステムメッセージファイルに出力された場合は、使用量がこの限度に達したことを示しています。

---

## 仮想端末

仮想端末 (pty) は、Solaris において主に 2 つの目的で使用されます。

- telnet、rlogin、または rsh コマンドを使用したりリモートログインをサポートする。
- X ウィンドウシステムがコマンドインタプリタウィンドウを作成するときに使用するインタフェースを提供する。

デスクトップワークステーション用の仮想端末の数はデフォルトで十分なため、チューニングの対象はリモートログイン用の pty の数になります。

Solaris の以前のバージョンでは、pty の数を明示的にシステムに構成する手順を行う必要がありました。しかし、Solaris 8 リリースからは、新しい機構により、ほとんどの場合、このチューニングを行う必要はありません。pty 数のデフォルト値はシステムのメモリー量に基づいて計算されるようになったので、デフォルト値を増やしたり、減らしたりする必要があるのは、このデフォルト値を変更する場合だけです。

構成処理では、次の 3 つの関連する変数を使用されます。

- pt\_cnt - pty 数のデフォルトの最大値
- pt\_pctofmem - pty サポート構造体専用で使用できるカーネルメモリーの割合 (%)
- pt\_max\_pty - pty 数の「強い」制限の最大値

pt\_cnt のデフォルト値はゼロで、pt\_max\_pty が設定されていない限り、システムは pct\_pctofmem に指定されたメモリー量に基づいてログインを制限します。pt\_cnt がゼロでない場合は、この制限内で pty が割り当てられます。この制限に達すると、システムは pt\_max\_pty を参照します。pt\_max\_pty がゼロでなければ、pt\_cnt と比較され、pt\_cnt が pt\_max\_pty より小さければ、pty 割り当てが認められます。pt\_max\_pty がゼロの場合は、pt\_cnt が、pt\_pctofmem に基づいてサポートされる pty の数と比較されます。pt\_cnt がこの数より小さければ、pty 割り当てが認められます。pt\_pctofmem に基づいた制限値が有効となるのは、pt\_cnt と ptms\_ptymax のデフォルト値が両方ともゼロの場合だけであることに留意してください。

pty の「強い」制限値を、pt\_pctofmem から計算される最大値と異なるものにするには、/etc/system の pt\_cnt と ptms\_ptymax に希望する pty 数を設定します。この場合、ptms\_pctofmem の設定は関連しません。

システムメモリーの特定の割合を pty サポートのためだけに割り当て、明示的な限度の管理をオペレーティングシステムに任せる場合は、次のようにします。

- /etc/system の pt\_cnt と ptms\_ptymax を設定しない。
- /etc/system の pt\_pctofmem に希望する割合 (%) を設定する。たとえば、10% を割り当てると、pt\_pctofmem=10。

このメモリーは、pty のサポートに使用されるまで実際に割り当てられません。しかし、メモリーがいったん割り当てられると、解放されません。

## pt\_cnt

説明	利用できる /dev/pts エントリの数は、システムで使用可能な物理メモリー量によって決まる上限までの間で動的に決められます。pt_cnt は、システムがサポートできるログイン数の最小値を決める 3 つの変数のうちの 1 つです。システムがサポートできる /dev/pts デバイスのデフォルトの最大数は、ブート時に、指定されたシステムメモリーの割合 (次の pt_pctofmem を参照) に適合する pty 構造体の数を計算することによって決められます。pt_cnt がゼロの場合、システムはこの最大数まで割り当てます。pt_cnt がゼロでない場合は、システムは pt_cnt かデフォルトの最大数のうち大きい方で割り当てます。
データ型	符号なし整数
デフォルト	0
範囲	0 から maxpid
単位	ログイン / ウィンドウ
動的か	いいえ

検査	なし
どのような場合に変更するか	システムにリモートからログインできるユーザーの数を明示的にコントロールしたい場合
コミットレベル	変更の可能性あり
変更履歴	詳細は、190ページの「 <code>pt_cnt</code> (Solaris 7 リリースおよびそれより前のリリース)」を参照してください。

## pt\_pctofmem

説明	<code>/dev/pts</code> エントリをサポートするためにデータ構造体が消費できる物理メモリーの最大の割合 (%)。64 ビットカーネルのシステムでは <code>/dev/pts</code> エントリ当たり 176 バイト、32 ビットカーネルのシステムでは <code>/dev/pts</code> エントリ当たり 112 バイトを消費します。
データ型	符号なし整数
デフォルト	5
範囲	0 から 100
単位	%
動的か	いいえ
検査	なし
どのような場合に変更するか	システムにログインできるユーザーの数を制限するか増やしたい場合。ゼロを指定すると、リモートユーザーがシステムにログインすることはできません。
コミットレベル	変更の可能性あり

## pt\_max\_pty

説明	システムが提供する pty の最大数
データ型	符号なし整数
デフォルト	0 (システムが定義した最大数を使用する)
範囲	0 から MAXUINT
単位	ログイン / ウィンドウ
動的か	はい
検査	なし
暗黙的制約	pt_cnt 以上でなければなりません。ただし、割り当てられた pty の数が pt_cnt の値を超えるまで、この値はチェックされません。
どのような場合に変更するか	システムが、構成値に基づいてより多くのログインをサポートできる場合であっても、サポートするログイン数の絶対的な上限を設定したい場合。
コミットレベル	変更の可能性あり

---

## ストリーム

### nstrpush

説明	1 つのストリームに挿入できる (プッシュできる) モジュールの数
データ型	符号付き整数
デフォルト	9
範囲	9 から 16
単位	モジュール
動的か	はい

検査	なし
どのような場合に変更するか	ソフトウェアベンダーからの指定がある場合。 STREAM が許可されているプッシュカウントを超えても、メッセージは出されません。プッシュを試みたプログラムに EINVAL という値が返されます。
コミットレベル	変更の可能性あり

## strmsgsz

説明	1 回のシステム呼び出しで STREAM に渡してメッセージのデータ部分に入れることができるバイト数の最大値。このサイズを超える write(2) はすべて、複数のメッセージに分割されます。
データ型	符号付き整数
デフォルト	65,536
範囲	0 から 262,144
単位	バイト
動的か	はい
検査	なし
どのような場合に変更するか	putmsg(2) 呼び出しが ERANGE を返す場合
コミットレベル	変更の可能性あり

## strctlsz

説明	1 回のシステム呼び出しで STREAM に渡してメッセージの制御部分に入れることができるバイト数の最大値。
データ型	符号付き整数

デフォルト	1024
範囲	0 から MAXINT
単位	バイト
動的か	はい
検査	なし
どのような場合に変更するか	ソフトウェアベンダーの指定がある場合。putmsg(2) 呼び出しがこの限度を超えようとすると、ERANGE が返されます。
コミットレベル	変更の可能性あり

---

## System V メッセージキュー

System V メッセージキューは、カーネルが作成したキューを使用してメッセージを交換する、メッセージ転送インタフェースを提供します。Solaris 環境では、メッセージをキューに入れたりキューから取り出したりするためのインタフェースが提供されます。メッセージは、自身の型を持つことができます。キューに入れる場合、メッセージはキューの終わりに置かれます。キューから取り出す場合は、指定された型の最初のメッセージがキューから削除されます。型が指定されていない場合は、最初のメッセージが削除されます。

モジュールは最初の参照で動的にロードされます。サブシステムに提供されるパラメータはそのときに検査されます。/etc/system ファイルのエントリには msgsys: 接頭辞が含まれていなければなりません。

この機能は POSIX 1003.1b メッセージキューの機能とは異なります。

Solaris 8 リリースでは、この機能のパラメータが一部変更されました。msgsys:msginfo\_msgssz、msgsys:msginfo\_msgmap、msgsys:msginfo\_msgseg パラメータは廃止になりましたが、エラーメッセージを避けるためにそのまま残されています。これらのパラメータの値は無視されます。

一度に処理できるメッセージの最大数は完全に msgsys:msginfo\_msgtql で定義されるようになりました。この変数で指定された値に基づいたメッセージヘッダーの配列が割り当てられ、空リストとして初期設定されます。メッセージの送信が必要になると、この空リストが調べられ、使用できるヘッダーがあれば、メッセージ

データを処理するためにバッファがカーネルメモリから割り当てられます。次に、データがバッファにコピーされ、メッセージが宛先のキューに入れられます。メッセージが読み取られると、バッファは解放され、ヘッダーが空リストに置かれます。

Solaris の以前のバージョンでは、メッセージ数の制限

は、`msgsys:msginfo_msgtql` を設定するか、メッセージバッファプールに割り当てられるメモリセグメントの数とセグメントのサイズを制限することによって行われていました。モジュールは、最初にロードされる時に、メッセージの管理に必要な数のデータ構造体を割り当てます。これらの構造体に割り当てられた空間の合計は、使用可能なカーネルメモリの 25% を超えることはできません。合計がこの値を超えると、ロードは失敗し、次のメッセージが表示されます。

```
msgsys: can't load module, too much memory requested
```

Solaris の以前のバージョンとは異なり、メッセージバッファプールはセットアップの中には割り当てられませんし、25% のメモリ検査でも考慮されなくなりました。

## `msgsys:msginfo_msgmax`

説明	System V メッセージの最大サイズ
データ型	符号なし long
デフォルト	2048
範囲	0 から物理メモリの総量
単位	バイト
動的か	いいえ。msginfo 構造体の msgmax フィールドにロードされます。
検査	なし
どのような場合に変更するか	<code>msgsnd(2)</code> 呼び出しが <code>EINVAL</code> エラーを返す場合、またはソフトウェアベンダーが推奨する場合
コミットレベル	変更の可能性あり

## msgsys:msginfo\_msgmnb

説明	1つのメッセージキューに入れることができるバイト数の最大値
データ型	符号なし long
デフォルト	4096
範囲	0 から物理メモリーの総量
単位	バイト
動的か	いいえ。msginfo 構造体の msgmnb フィールドにロードされます。
検査	なし
どのような場合に変更するか	msgsnd(2) 呼び出しがブロックするか EAGAIN エラーを返す場合、またはソフトウェアベンダーが推奨する場合
コミットレベル	変更の可能性あり

## msgsys:msginfo\_msgmni

説明	作成することができるメッセージキューの最大数
データ型	符号付き整数
デフォルト	50
範囲	0 から MAXINT
動的か	いいえ。msginfo 構造体の msgmni フィールドにロードされます。
検査	なし
どのような場合に変更するか	msgget(2) 呼び出しが ENOSPC エラーを返す場合、またはソフトウェアベンダーが推奨する場合
コミットレベル	変更の可能性あり

## msgsys:msginfo\_msgtql

説明	作成できるメッセージの最大数。msgsnd(2) 呼び出しでこの限度を超える要求が行われた場合は、メッセージヘッダーが使用可能になるまで要求は延期されます。あるいは、その要求で IPC_NOWAIT フラグがオンに設定されていると、要求はエラー EAGAIN で失敗します。
データ型	符号付き整数
デフォルト	40
範囲	0 から MAXINT
動的か	いいえ。msginfo 構造体の msgtql フィールドにロードされます。
検査	なし
どのような場合に変更するか	msgsnd(2) 呼び出しがブロックするか EAGAIN エラーを返す場合、またはソフトウェアベンダーが推奨する場合
コミットレベル	変更の可能性あり

---

## System V セマフォ

System V セマフォは Solaris 環境で計数型セマフォを提供します。System V セマフォでは、セマフォの標準的な設定/解放操作の他に、必要に応じて増分や減分を行う値を持つことができます (たとえば、使用可能なリソースの数を表すなど)。セマフォ群に対して同時に操作を行う機能や、プロセスが終了すると、そのプロセスの最後の操作をシステムが取り消す機能も提供されます。

セマフォは組で作成されます。

モジュールは最初の参照で動的にロードされます。この時点で、サブシステムに渡されたパラメータが検査され、セマフォを含むすべてのデータ構造体を作成されます。したがって、実行時にパラメータの値を変更することはできません。値を増や

すと、データ破壊の可能性があるからです。/etc/system ファイルのエントリは `semsys:` 接頭辞を含んでいなければなりません。

この機能は、POSIX 1003.1b セマフォ機能とは異なります。

## `semsys:seminfo_semmni`

説明	セマフォ識別子の最大数
データ型	符号付き整数
デフォルト	10
範囲	1 から 65,535
動的か	いいえ
検査	<code>SEMA_INDEX_MAX</code> (現在は 65,535) と比較し、それより大きい場合は、 <code>SEMA_INDEX_MAX</code> の値にリセットされます。警告メッセージがコンソールかシステムメッセージファイル (またはその両方) に書き込まれます。
どのような場合に変更するか	デフォルトの組数では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。現在の設定値を越えるセットを作成しようとしても、エラーメッセージは表示されません。 <code>semget(2)</code> 呼び出しからアプリケーションに <code>ENOSPC</code> が返されます。
コミットレベル	変更の可能性あり

## `semsys:seminfo_semmns`

説明	システムにおける System V セマフォの最大数
データ型	符号付き整数
デフォルト	60
範囲	1 から MAXINT

動的か	いいえ
検査	セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュールが最初にロードされる時に使用可能なカーネルメモリーの 25% と比較されます。使用される容量がこのメモリーのしきい値より大きい場合、モジュールはロードを拒否し、セマフォ機能は利用できません。
どのような場合に変更するか	デフォルトのセマフォ数では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。設定された値を越える数のセマフォを作成しようとしても、エラーメッセージは表示されません。semget (2) 呼び出しからアプリケーションにリターンコード ENOSPC が返されます。
コミットレベル	変更の可能性あり

## semsys:seminfo\_semvmx

説明	セマフォに設定できる最大値
データ型	符号なし short
デフォルト	32,767
範囲	1 から 65,535
動的か	いいえ
検査	なし
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。この最大値を越えても、エラーメッセージは表示されません。semop (2) 呼び出しからアプリケーションにリターンコード ERANGE が返されます。

コミットレベル

変更の可能性あり

## semsys:seminfo\_semmsl

説明	1つのセマフォ識別子当たりの System V セマフォ数の最大値
データ型	符号付き整数
デフォルト	25
範囲	1 から MAXINT
動的か	いいえ
検査	セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュールが最初にロードされる時に使用可能なカーネルメモリーの 25% と比較されます。使用される容量がこのメモリーのしきい値より大きい場合、モジュールはロードを拒否し、セマフォ機能は利用できません。
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。設定された値を越える数のセマフォを組として作成しようとしても、エラーメッセージは表示されません。semget(2) 呼び出しからアプリケーションにリターンコード EINVAL が返されます。
コミットレベル	変更の可能性あり

## semsys:seminfo\_semopm

説明	1 回の semop(2) 呼び出しで実行できる System V セマフォ操作数の最大値。このパラメータは、semop(2) システム呼び出しで使用する sops 配列内の sembufs の数を規定します。
----	---

データ型	符号付き整数
デフォルト	10
範囲	1 から MAXINT
動的か	いいえ
検査	セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュールが最初にロードされる時に使用可能なカーネルメモリーの 25% と比較されます。使用される容量がこのメモリーのしきい値より大きい場合、モジュールはロードを拒否し、セマフォ機能は利用できません。
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。1 回の <code>semop(2)</code> 呼び出しで許可されている値を越える回数のセマフォ操作を実行しようとしても、エラーメッセージは表示されません。 <code>semop(2)</code> 呼び出しからアプリケーションにリターンコード <code>E2BIG</code> が返されます。
コミットレベル	変更の可能性あり

## `semsys:seminfo_semmnu`

説明	System V セマフォシステムによってサポートされる取り消し構造体の合計数
データ型	符号付き整数
デフォルト	30
範囲	1 から MAXINT
動的か	いいえ
検査	セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュール

が最初にロードされるときに使用可能なカーネルメモリーの 25% と比較されます。使用される容量がこのメモリーのしきい値より大きい場合、モジュールはロードを拒否し、セマフォ機能は利用できません。

どのような場合に変更するか

デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。設定された値を越える回数の取り消し操作を実行しようとしても、エラーメッセージは表示されません。システムが取り消し構造体を使い果たすと、`semop(2)` 呼び出しからアプリケーションに戻り値 `ENOSPC` が返されます。

コミットレベル

変更の可能性あり

前のリリースからの変更

詳細は、198ページの「`semsys:seminfo_semmnu` (Solaris 8 1/01 リリース)」を参照してください。

## `semsys:seminfo_semume`

説明

1つのプロセスで使用できる System V セマフォの取り消し構造体の最大数

データ型

符号付き整数

デフォルト

10

範囲

1 から MAXINT

動的か

いいえ

検査

セマフォとそれらに関するデータ構造体によって消費され得る空間の量が、モジュールが最初にロードされるときに使用可能なカーネルメモリーの 25% と比較されます。使用される容量がこのメモリーのしきい値より大きい場合、モジュールはロードを拒否し、セマフォ機能は利用できません。

どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。設定された値を越える回数の取り消し操作を実行しようとしても、エラーメッセージは表示されません。semop(2) 呼び出しからアプリケーションにリターンコード EINVAL が返されます。
コミットレベル	変更の可能性あり

## semsys:seminfo\_semaem

説明	1 つの取り消し構造体にセマフォの値として設定できる最大値
データ型	符号なし short
デフォルト	16,384
範囲	1 から 65,535
動的か	いいえ
検査	なし
どのような場合に変更するか	デフォルト値では不十分な場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。設定された値を越える回数の取り消し操作を実行しようとしても、エラーメッセージは表示されません。semop(2) 呼び出しからアプリケーションにリターンコード EINVAL が返されます。
コミットレベル	変更の可能性あり

## System V 共有メモリー

System V 共有メモリーでは、プロセスによるセグメントの作成が可能です。連携するプロセスがそのメモリーセグメントに接続し(セグメントに対するアクセス権が必要)、セグメントに含まれるデータにアクセスできます。この機能はロード可能モジュールとして実装されます。/etc/system ファイルのエントリは shmsys: 接頭辞を含んでいる必要があります。Solaris 7 リリースから、keyserv デーモンは System V 共有メモリーを使用するようになりました。

DBMS ベンダーは、パフォーマンスを高めるために、ISM (intimate shared memory) と呼ばれる特殊な共有メモリーを使用しています。共有メモリーセグメントを ISM セグメントにすると、そのセグメントのメモリーがロックされます。これにより、必要な入出力経路が短縮され、メモリーの使用効率が向上します。これは、セグメントを記述するカーネルリソースが、セグメントに ISM モードで接続するすべてのプロセスによって共有されるからです。

このモジュールは最初の参照で動的にロードされます。サブシステムに提供されるパラメータはこのときに検査されます。

この機能は POSIX 1003.1b 共有メモリー機能とは異なります。

### shmsys:shminfo\_shmmax

説明	作成できる System V 共有メモリーセグメントの最大サイズ。このパラメータは、システムが要求されたメモリーセグメントを作成するために必要な物理リソースが実際にあるか調べる前に検査される上限値です。
データ型	符号なし long
デフォルト	1,048,576
範囲	32 ビットシステムでは 0 から MAXINT、64 ビットシステムでは MAXINT64
単位	バイト

動的か	いいえ。shminfo 構造体の shmmax フィールドにロードされます。
検査	なし
どのような場合に変更するか	デフォルト値では足りない場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。ただし、共有メモリーセグメントのサイズを制限する必要がある場合以外は、このパラメータに利用できる最大値を設定しても副次的影響はありません。
コミットレベル	変更の可能性あり

## shmsys:shminfo\_shmmin

説明	作成できる System V 共有メモリーセグメントの最小サイズ
データ型	符号なし long
デフォルト	1
範囲	0 から物理メモリーの総量
単位	バイト
どのような場合に変更するか	できるだけ変更しないでください。この値が大きすぎると、powerd などのシステムプログラムが異常終了することがあります。shminfo_shmmin より小さいセグメントを作成するプログラムには、そのセグメントを作成しようとした時に EINVAL エラーが返され、通常、そのプログラムは終了します。
コミットレベル	変更の可能性あり
前のリリースからの変更	詳細は、197ページの「shmsys:shminfo_shmmin (Solaris 8 1/01 リリース)」を参照してください。

## shmsys:shminfo\_shmmni

説明	システム全体で作成できる共有メモリーセグメントの最大数
データ型	符号付き整数
デフォルト	100
範囲	0 から MAXINT
動的か	いいえ。shminfo 構造体の shmmni フィールドにロードされます。
検査	System V 共有メモリーに関するデータ構造体が最大数存在する場合に消費され得る空間量を、モジュールをロードしたときに使用可能なカーネルメモリーの 25% と比較します。消費されるメモリーが大きすぎると、モジュールをロードする試みは失敗します。
どのような場合に変更するか	システムの限度が小さすぎる場合。一般には、システムベンダーの推奨がある場合に変更します。
コミットレベル	変更の可能性あり

## shmsys:shminfo\_shmseg

説明	1 つのプロセスで作成できる共有メモリーセグメントの制限数
データ型	符号付き short
デフォルト	6
範囲	0 から 32,767
動的か	いいえ。shminfo 構造体の shmseg フィールドにロードされます。
検査	System V 共有メモリーに関するデータ構造体が最大数存在する場合に消費され得る空間量

を、モジュールをロードしたときに使用可能なカーネルメモリーの 25% と比較します。消費されるメモリーが大きすぎると、モジュールをロードする試みは失敗します。

どのような場合に変更するか	システムの限度が小さすぎる場合。一般には、ソフトウェアベンダーの推奨があった場合に変更します。
コミットレベル	変更の可能性あり

## segspt\_minfree

説明	ISM 共有メモリー用に割り当てることができないシステムメモリーのページ数
データ型	符号なし long
デフォルト	最初の ISM セグメントが作成される時に使用可能なシステムメモリーの 5%
範囲	0 から 32,767
単位	ページ
動的か	はい
検査	なし。値が小さすぎると、メモリーが ISM セグメントに消費される時に、システムがハングしたりパフォーマンスが大幅に低下することがあります。
どのような場合に変更するか	大量のメモリーがあるデータベースシステムで ISM を使用する場合、このパラメータを下げると効果があることがあります。ISM セグメントが使用されない場合には、このパラメータの効果はありません。大量のメモリーを備えたマシンでは、ほとんどの場合、最大値 128M バイト (0x4000) で十分です。
コミットレベル	変更の可能性あり

---

## スケジューリング

### rechoose\_interval

説明	プロセスが最後に実行していた CPU に対するすべての親和性を失ったものとみなされるまでの、クロック刻みの数。この期間が過ぎると、すべての CPU はスレッドスケジューリングの候補と見なされます。このパラメータは、タイムシェアリングクラスのスレッドに対してのみ意味を持ちます。リアルタイムスレッドは、最初の使用可能な CPU に対してスケジュールされます。
データ型	符号付き整数
デフォルト	3
範囲	0 から MAXINT
動的か	はい
検査	なし
どのような場合に変更するか	キャッシュが大きい場合、極めて重要なプロセスがシステムで動作している場合や、データアクセスパターン以外の原因により、一連のプロセスで過度のキャッシュミスが発生していると思われる場合。パラメータを変更する前に、Solaris 2.6 以降に利用可能になったプロセッサセット (psrset(1M)) 機能、またはプロセッサ結合 (pbind(1M)) を使用することを検討してください。
コミットレベル	変更の可能性あり

---

## タイマー

### hires\_tick

説明	この変数を有効にすると、Solaris 環境でシステムクロックレートとして、デフォルト値 100 の代わりに 1000 が使用されます。
データ型	符号付き整数
デフォルト	0
範囲	0 (無効)、1 (有効)
動的か	いいえ。新しいシステムタイミング変数はブート時に設定されます。ブート後は、このパラメータは参照されません。
検査	なし
どのような場合に変更するか	10 ミリ秒未満、1 ミリ秒以上の分解能を持つタイムアウトが必要な場合
コミットレベル	変更の可能性あり

### timer\_max

説明	使用可能な POSIX タイマーの数
データ型	符号付き整数
デフォルト	32
範囲	0 から MAXINT
動的か	いいえ。値を増やすと、システムクラッシュを起こす可能性があります。
検査	なし

どのような場合に変更するか	システムのデフォルトのタイマー数では不十分な場合。timer_create() 呼び出しの実行時に、EAGAIN エラーがアプリケーションに返されます。
コミットレベル	変更の可能性あり

## Sun4u 特有

### consistent\_coloring

#### 説明

Solaris 2.6 リリースから、UltraSPARC (sun4u) プラットフォームでさまざまなページ配置ポリシーを使用する機能が導入されました。ページ配置ポリシーは、L2 キャッシュの使用が最適化されるように物理ページアドレスを割り当てようとするものです。デフォルトアルゴリズムとしてどのアルゴリズムが選択されたとしても、特定のアプリケーション群にとって、そのアルゴリズムが別のアルゴリズムよりも適していない可能性があります。この変数は、システムのすべてのプロセスに適用される配置アルゴリズムを変更します。

メモリーは、L2 キャッシュのサイズに基づいて区画に分割されます。マップされていないページでページフォルトが最初に起こると、ページ配置コードは1つの区画から1つのページを割り当てます。選択されるページは、次の3つのアルゴリズムのどれが使用されているかによって異なります。

- ページ彩色 - ページが選択される区画は、仮想アドレスのさまざまなビットに基づいて決められます。Solaris 8 リリースでは、これがデフォルトのアルゴリズムです。このアルゴリズムを使用するには、consistent\_coloring をゼロに設定し

ます。このアルゴリズムでは、プロセス別の履歴はありません。

- 仮想アドレス = 物理アドレス - プログラム内の連続するページに、連続する区画からページを選択します。このアルゴリズムを使用するには、`consistent_coloring` に 1 を設定します。このアルゴリズムでは、プロセス別の履歴はありません。
- 区画飛び越し - プログラム内の連続するページに、通常、1つおきの区画からページを割り当てます。ただし、このアルゴリズムは、ときには2つ以上の区画を飛び越すこともあります。このアルゴリズムを使用するには、`consistent_coloring` に 2 を設定します。各プロセスは、無作為に選択された区画から開始し、割り当てられた最後の区画のプロセスごとの記録が保管されます。

動的か

はい

検査

なし。値が 2 より大きいと、「WARNING: AS\_2\_BIN: bad consistent coloring value」メッセージがいくつかコンソールに表示され、その後すぐにシステムがハングします。復旧には、電源を再投入する必要があります。

どのような場合に変更するか

システムの主な作業負荷が、長い時間動作するハイパフォーマンスコンピューティング (HPC) アプリケーションである場合。この値を変更すると、パフォーマンスが向上することがあります。ファイルサーバーやデータベースサーバー、それに多数のアクティブプロセスが動作するシステム (たとえばコンパイルやタイムシェアリングサーバーなど) では、この値を変更しても効果はありません。

コミットレベル

変更の可能性あり



## NFS チューニング可能パラメータ

---

この章では、NFS のチューニング可能パラメータについて説明します。カーネルのチューニング可能パラメータについては第 2 章を、TCP/IP のチューニング可能パラメータについては第 4 章をそれぞれ参照してください。

- 110ページの「NFS モジュールのパラメータ」
- 137ページの「nfssrv モジュールのパラメータ」
- 141ページの「rpcmod モジュールのパラメータ」

---

## NFS 環境のチューニング

これらのパラメータは、ブートプロセス中に読み込まれる `/etc/system` ファイルに定義します。各パラメータは、それが属するカーネルモジュールの名前と、自身を表わすパラメータ名で特定されます。詳細は、19ページの「Solaris システムのチューニング」を参照してください。

---

注・シンボル名や、それが存在するモジュール、デフォルト値は、リリースによって変わることがあります。変更を行ったり、前のリリースの値を適用したりする前に、使用する SunOS リリースのバージョンのマニュアルをチェックしてください。

---

---

## NFS モジュールのパラメータ

ここでは、NFS カーネルモジュールに関連するパラメータについて説明します。

### `nfs:nfs3_pathconf_disable_cache`

説明	NFS バージョン 3 でマウントされたファイルシステムの <code>pathconf(2)</code> 情報をキャッシングするかどうかを制御します。
データ型	整数 (32 ビット)
デフォルト	0 (キャッシングを有効にする)
範囲	0 (キャッシングを有効にする)、1 (キャッシングを無効にする)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	<code>pathconf</code> 情報はファイルごとにキャッシュされます。しかし、サーバーが特定ファイルの情報を動的に変更する可能性がある場合は、このパラメータを使用してキャッシングを無効にします。これは、クライアントがそのキャッシュエントリを検証する方法がないためです。
安定性レベル	発展中

### `nfs:nfs_allow_preepoch_time`

説明	タイムスタンプが正しくなかったり「負」であるファイルをクライアントから表示できるようにするかどうかを制御します。
----	--

従来、NFS クライアントも NFS サーバーも、返されるファイルの時間範囲を、これらの属性を使用してチェックすることはありませんでした。受信するタイムスタンプの値は符号なしの 32 ビット long であるため、すべての値が有効だったからです。

しかし、32 ビットの Solaris リリースが動作しているシステムでは、タイムスタンプの値が符号付きの 32 ビット long であるため、タイムスタンプが 1970 年 1 月 1 日より前の表示 (つまり「昔」の) になっていることがあります。

64 ビットの Solaris リリースが動作しているシステムでの問題は、これとは多少異なります。64 ビットの Solaris リリースでは、タイムスタンプ値が符号付きの 64 ビット long であるため、時刻フィールドがフルの 32 ビットの時刻を表しているのか、時刻フィールドが負の時刻、つまり、1970 年 1 月 1 日より前を表しているのかを判別できません。

32 ビットから 64 ビットに変換するとき時刻値に符号を付けるべきかどうかを決定することはできません。つまり、時刻値が本当に負数を表している場合は、時刻値に符号を付ける必要がありますし、本当にフルの 32 ビット時刻値を表している場合は、符号を付けるべきではありません。この問題は、フルの 32 ビット時刻値を無効にすることによって解決できます。

データ型	整数 (32 ビット)
デフォルト	0 (32 ビットのタイムスタンプを無効にする)
範囲	0 (32 ビットのタイムスタンプを無効にする) か、1 (32 ビットのタイムスタンプを有効にする)
単位	ブール値

動的か	はい
検査	なし
どのような場合に変更するか	「正常な」操作が行われていても、ファイルによっては、タイムスタンプ値がはるかに離れた将来や過去の日付に設定されることがあります。NFS でマウントされたファイルシステムを使用してこれらのファイルにアクセスすることが望ましい場合は、このパラメータを 1 にすれば、タイムスタンプ値をチェックなしで受け取ることができます。
安定性レベル	発展中

### nfs:nfs\_cots\_timeo

説明	トランスポートプロトコルとして TCP などの接続型トランスポートを使用している、NFS バージョン 2 でマウントされたファイルシステムの、デフォルトの RPC タイムアウトを制御します。
データ型	符号付き整数 (32 ビット)
デフォルト	600 (60 秒)
範囲	0 から $2^{31} - 1$
単位	10 分の 1 秒
動的か	はい。ただし、ファイルシステムの RPC タイムアウトは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからファイルシステムをアンマウントし、再びマウントします。
検査	なし
どのような場合に変更するか	TCP では、要求と応答が適切に転送されるよう、機能します。しかし、特に遅いネッ

トワークにおいて往復時間が非常に長くなると、NFS バージョン 2 のクライアントがタイムアウトになってしまう可能性があります。

クライアントが無用にタイムアウトしてしまうことを防ぐには、このパラメータを増やします。ただし、値の範囲が非常に広いため、この値を大きくしすぎると、実際には再転送が必要なのにその状況が長い間検出されないおそれがあります。

安定性レベル

発展中

## nfs:nfs3\_cots\_timeo

説明	トランスポートプロトコルとして TCP などの接続型トランスポートを使用している、NFS バージョン 3 でマウントされたファイルシステムの、デフォルトの RPC タイムアウトを制御します。
データ型	符号付き整数 (32 ビット)
デフォルト	600 (60 秒)
範囲	0 から $2^{31} - 1$
単位	10 分の 1 秒
動的か	はい。ただし、ファイルシステムの RPC タイムアウトは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからファイルシステムをアンマウントし、再びマウントします。
検査	なし
どのような場合に変更するか	TCP では、要求と応答が適切に転送されるよう機能します。しかし、特に遅いネットワークにおいて往復時間が非常に長くなると、NFS バージョン 3 のクライアントがタイムアウトになってしまう可能性があります。クライアント

が無用にタイムアウトしてしまうことを防ぐには、このパラメータを増やします。ただし、値の範囲が非常に広いので、この値を大きくしすぎると、実際には再転送が必要なのにその状況が長い間検出されないおそれがあります。

安定性レベル

発展中

## nfs:nfs\_do\_symlink\_cache

説明

NFS バージョン 2 でマウントされたファイルシステムに対してシンボリックリンクファイルの内容をキャッシュするかどうかを制御します。

データ型

整数 (32 ビット)

デフォルト

1 (キャッシングを有効にする)

範囲

0 (キャッシングを無効にする)か、1 (キャッシングを有効にする)

単位

ブール値

動的か

はい

検査

なし

どのような場合に変更するか

サーバーが、シンボリックリンクファイルの内容を変更してもそのファイルの更新タイムスタンプを更新しない場合や、タイムスタンプの精度が粗すぎる場合には、シンボリックリンクファイルの内容が変更されても、クライアントにはその変更が長い間見られないことがあります。その場合には、このパラメータを使用してシンボリックリンクの内容のキャッシングを無効にすれば、クライアントで動作しているアプリケーションで変更をただちに見ることができます。

安定性レベル

発展中

## nfs:nfs3\_do\_symlink\_cache

説明	NFS バージョン 3 でマウントされたファイルシステムに対してシンボリックリンクファイルの内容をキャッシュするかどうかを制御します。
データ型	整数 (32 ビット)
デフォルト	1 (キャッシングを有効にする)
範囲	0 (キャッシングを無効にする)か、1 (キャッシングを有効にする)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	サーバーが、シンボリックリンクファイルの内容を変更してもそのファイルの更新タイムスタンプを更新しない場合や、タイムスタンプの精度が粗すぎる場合には、シンボリックリンクファイルの内容が変更されても、クライアントにはその変更が長い間見られないことがあります。その場合には、このパラメータを使用してシンボリックリンクの内容のキャッシングを無効にすれば、クライアントで動作しているアプリケーションで変更をただちに見ることができます。
安定性レベル	発展中

## nfs:nfs\_dynamic

説明	UDP などのコネクションレストランスポートを使用する、NFS バージョン 2 でマウントされたファイルシステムに対し、「動的再転送」と呼ばれる機能を有効にするかどうかを制御します。この機能は、サーバーの応答時間を監視し、RPC タイムアウトと読み取り / 書き込みの
----	--

転送サイズを調整することによって、再転送の回数を減らそうとするものです。

データ型	整数 (32 ビット)
デフォルト	1 (有効)
範囲	0 (無効) か 1 (有効)
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	なし
どのような場合に変更するか	サーバーの応答やネットワークの負荷が急激に変動する状況では、動的再転送の機能により RPC タイムアウトの回数が不当に増えたり、読み取り/書き込みの転送サイズが不必要に小さくなったりすることがあります。この機能を無効にすると、スループットは向上するかもしれませんが、おそらくサーバーの応答やネットワークの負荷により急激な変動が目立つようになります。
安定性レベル	発展中

## nfs:nfs3\_dynamic

説明	UDP などのコネクションレストランスポートを使用する、NFS バージョン 3 でマウントされたファイルシステムに対し、「動的再転送」と呼ばれる機能を有効にするかどうかを制御します。この機能は、サーバーの応答時間を監視し、RPC タイムアウトと読み取り / 書き込みの転送サイズを調整することによって、再転送の回数を減らそうとするものです。
データ型	整数 (32 ビット)

デフォルト	0 (有効)
範囲	0 (無効) か 1 (有効)
単位	ブール値
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	なし
どのような場合に変更するか	サーバーの応答やネットワークの負荷が急激に変動する状況では、動的再転送の機能により RPC タイムアウトの回数が不当に増えたり、読み取り/書き込みの転送サイズが不必要に小さくなったりすることがあります。この機能を無効にすると、スループットは向上するかもしれませんが、おそらくサーバーの応答やネットワークの負荷により急激な変動が目立つようになります。
安定性レベル	発展中

## nfs:nfs\_lookup\_neg\_cache

説明	NFS バージョン 2 でマウントされたファイルシステムに対し、ネガティブ名前キャッシュを使用するかどうかを制御します。このネガティブ名前キャッシュを使用すると、ルックアップされたファイル名が存在しない場合には、そのファイル名が記録されます。このキャッシュは、存在しないことがすでにわかっているファイル名のルックアップ要求をネットワークを介して繰り返し行うのを避けるためにあります。
データ型	整数 (32 ビット)
デフォルト	1 (有効)

範囲	0 (無効) か 1 (有効)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	このキャッシュが正しく機能するためには、ネガティブエントリが使用前に厳密に検証されなければなりません。この整合性機構は、読み取り専用でマウントされたファイルシステムには若干緩和されています。つまり、サーバー上のファイルシステムが変更されないか、変更されてもゆっくりであり、そのような変更はクライアントにゆっくり伝達されても問題がないとみなされるためです。この場合は、この整合性機構が通常の属性キャッシュ機構になります。  読み取り専用でクライアントにマウントされているファイルシステムがサーバーで変更されたら、その変更をクライアントでただちに見る必要がある場合は、このパラメータを使用してネガティブキャッシュを無効にします。
安定性レベル	発展中

## nfs:nfs3\_lookup\_neg\_cache

説明	NFS バージョン 3 でマウントされたファイルシステムに対し、ネガティブ名前キャッシュを使用するかどうかを制御します。このネガティブ名前キャッシュを使用すると、ルックアップされたファイル名が存在しない場合には、そのファイル名が記録されます。このキャッシュは、存在しないことがすでにわかっているファイル名のルックアップ要求をネットワークを介して繰り返し行うのを避けるためにあります。
データ型	整数 (32 ビット)

デフォルト	1 (有効)
範囲	0 (無効) か 1 (有効)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	このキャッシュが正しく機能するためには、ネガティブエントリが使用前に厳密に検証されなければなりません。この整合性機構は、読み取り専用でマウントされたファイルシステムには若干緩和されています。つまり、サーバー上のファイルシステムが変更されないか、変更されてもゆっくりであり、そのような変更はクライアントにゆっくり伝達されても問題がないとみなされるためです。この場合は、この整合性機構が通常の属性キャッシュ機構になります。  読み取り専用でクライアントにマウントされているファイルシステムがサーバーで変更されたら、その変更をクライアントでただちに見る必要ある場合は、このパラメータを使用してネガティブキャッシュを無効にします。
安定性レベル	発展中

## nfs:nfs\_max\_threads

説明	NFS バージョン 2 クライアントの非同期入出力を行うカーネルスレッドの数を制御します。NFS は RPC に基づくものであり、RPC はもともと同期して動作する機能であるため、呼び出し側のスレッドと非同期に NFS 操作を行うには、個別の実行コンテキストが必要です。  非同期に実行できる操作には、先読み read、先読み readdir の readdir、putpage 要求と pageio 要求の書き込みがあります。
----	--

データ型	整数 (16 ビット)
デフォルト	8
範囲	0 から $2^{15} - 1$
単位	スレッド
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからファイルシステムをアンマウントし、再びマウントします。
検査	なし
どのような場合に変更するか	ある時点で存在する同時入出力操作の数を増やしたり、減らしたりする場合には、このパラメータを変更します。たとえば、帯域幅が非常に狭いネットワークでは、NFS クライアントによるネットワークの過負荷を防止するためにこの値を減らすことがあります。あるいは、ネットワークの帯域幅が非常に広く、クライアントとサーバーが両方とも十分なリソースを備えている場合は、ネットワークの帯域幅やクライアントとサーバーのリソースをより有効に使用するためにこの値を増やすことができます。
安定性レベル	変更の可能性あり

## nfs:nfs3\_max\_threads

説明	NFS バージョン 3 クライアントの非同期入出力を行うカーネルスレッドの数を制御します。NFS は RPC に基づくものであり、RPC はもともと同期して動作する機能であるため、呼び出し側のスレッドと非同期に NFS 操作を行うには、個別の実行コンテキストが必要です。
----	---

非同期に実行できる操作には、先読み read、先読み readdir の readdir、putpage 要求と pageio 要求の書き込み、およびコミットがあります。

データ型	整数 (16 ビット)
デフォルト	8
範囲	0 から $2^{15} - 1$
単位	スレッド
動的か	はい。ただし、このパラメータは、ファイルシステムごとにマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからファイルシステムをアンマウントし、再びマウントします。
検査	なし
どのような場合に変更するか	ある時点で存在する同時入出力操作の数を増やしたり、減らしたりする場合には、このパラメータを変更します。たとえば、帯域幅が非常に狭いネットワークでは、NFS クライアントによるネットワークの過負荷を防止するためにこの値を減らすかもしれません。あるいは、ネットワークの帯域幅が非常に広く、クライアントとサーバーが両方とも十分なリソースを備えている場合は、ネットワークの帯域幅やクライアントとサーバーのリソースをより有効に使用するためにこの値を増やすことができます。
安定性レベル	変更の可能性あり

## nfs:nfs\_nra

説明	ファイルの順次アクセスが見つかった時に NFS バージョン 2 クライアントがキューに入れる先読み操作の数を制御します。これらの先読み操作では、並行性が高まり、読み取りのスループットが
----	--

向上します。個々の先読み要求は、通常、ファイルデータの 8192 バイトに対するものです。

データ型	整数 (32 ビット)
デフォルト	4
範囲	0 から $2^{31} - 1$
単位	先読み要求
動的か	はい
検査	なし
どのような場合に変更するか	特定のファイルに対してある時点で存在する先読み要求の数を増やしたり、減らしたりする場合は、このパラメータを変更します。たとえば、ネットワークの帯域幅が非常に狭い場合やクライアントのメモリーが少ない場合は、NFS クライアントによるネットワークの過負荷やシステムメモリーの使いすぎを防止するために、この値を減らすことができます。あるいは、ネットワークの帯域幅が非常に広く、クライアントとサーバーが両方とも十分なリソースを備えている場合は、ネットワークの帯域幅やクライアントとサーバーのリソースをより有効に使用するためにこの値を増やすことができます。
安定性レベル	変更の可能性あり

## nfs:nfs3\_nra

説明	ファイルの順次アクセスが見つかった時に NFS バージョン 3 クライアントがキューに入れる先読み操作の数を制御します。これらの先読み操作では、並行性が高まり、読み取りのスループットが向上します。個々の先読み要求は、通常、ファイルデータの 32,768 バイトに対するものです。
データ型	整数 (32 ビット)

デフォルト	4
範囲	0 から $2^{31} - 1$
単位	先読み要求
動的か	はい
検査	なし
どのような場合に変更するか	特定のファイルに対してある時点で存在する先読み要求の数を増やしたり、減らしたりする場合には、このパラメータを変更します。たとえば、ネットワークの帯域幅が非常に狭い場合やクライアントのメモリーが少ない場合は、NFS クライアントによるネットワークの過負荷やシステムメモリーの使いすぎを防止するために、この値を減らすことができます。あるいは、ネットワークの帯域幅が非常に広く、クライアントとサーバーが両方とも十分なリソースを備えている場合は、ネットワークの帯域幅やクライアントとサーバーのリソースをより有効に使用するためにこの値を増やすことができます。
安定性レベル	変更の可能性あり

## nfs:nrnode

説明	<p>NFS クライアントの rnode キャッシュのサイズを制御します。</p> <p>NFS バージョン 2 クライアントでも、バージョン 3 クライアントでも使用される rnode キャッシュは、NFS クライアント上のファイルを記述する中心的なデータ構造体です。このデータ構造体には、サーバーのファイルを識別するファイルハンドルや、NFS クライアントがサーバーへのネットワーク呼び出しを避けるために使用するさまざまなキャッシュへのポインタが含まれています。個々の rnode は vnode と 1</p>
----	--

対 1 で対応しています。vnode には、ファイルデータがキャッシュされます。

NFS クライアントは、キャッシュされたデータやメタデータが破棄されないように、最小限の rnode を保持しようとします。rnode の再利用や解放が行われると、キャッシュされたデータやメタデータは破棄されなければなりません。

データ型	整数 (32 ビット)
デフォルト	このパラメータのデフォルト値は 0 です。これは、nrnode の値に ncsiz e パラメータの値が設定されるべきであることを示しています。実際、nrnode の値が正でないと、nrnode には ncsiz e が設定されます。
範囲	1 から $2^{31} - 1$
単位	rnode
動的か	いいえ。この値を変更するには、パラメータを /etc/system ファイルに追加するか、そのパラメータを変更し、システムを再起動する必要があります。
検査	rnode キャッシュが使用可能なメモリーの 25% を超えないような最大値をシステムは強制します。
どのような場合に変更するか	rnode の作成や破棄は動的に行われるため、システムは、システムのメモリーの要求や同時にアクセスされるファイルの数が増えるに従って、キャッシュのサイズを自動的に調整して、nrnode サイズキャッシュを決定する傾向があります。しかし、アクセスするファイルの組み合わせが前もって予測できる場合など、状況によっては、nrnode の値を設定する方がよいこともあります。たとえば、NFS クライアントが少数の非常に大きなファイルにアクセスしている場合は、nrnode の値に小さな数を設定す

れば、システムメモリーには、`rnode` の代りにファイルデータをキャッシュできます。あるいは、クライアントが多数の小さなファイルにアクセスしている場合は、`nrnode` に十分大きな値を設定すれば、ファイルメタデータの格納を最適化して、メタデータを求めるネットワーク呼び出しの回数を減らすことができます。

`nrnode` の値に 1 を設定すると、`rnode` キャッシュは事実上無効になります (ただし、この方法はお勧めできません)。この場合、クライアントは `rnode` を 1 つしかキャッシュしないため、それが頻繁に再使用されます。

安定性レベル

発展中

## `nfs:nfs_shrinkreaddir`

説明

以前の一部の NFS サーバーでは、NFS バージョン 2 の `REaddir` 要求で 1024 バイトより大きいディレクトリ情報を求めるものが正しく処理されないことがありました。これは、サーバーの実装にバグがあったためです。このパラメータは、NFS バージョン 2 クライアントでの対処方法を含んでいます。

このパラメータを有効にすると、クライアントは、1024 バイトより大きいディレクトリ情報を求める `REaddir` 要求を生成しません。このパラメータを無効にすると、送信されるサイズは、`getdents(2)` システム呼び出しが `NFS_MAXDATA` (8192 バイト) で渡されるサイズのうち小さい方に設定されます。

データ型

整数 (32 ビット)

デフォルト

0 (無効)

範囲

0 (無効) か 1 (有効)

単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	古い NFS バージョン 2 のみのサーバーが使用され、ディレクトリの読み取りで相互運用性に問題がある場合は、このパラメータの値を調べてください。このパラメータを有効にすると、ディレクトリを読み取るアプリケーションのパフォーマンスが多少低下することがあります。
安定性レベル	発展中

## nfs:nfs\_write\_error\_interval

説明	NFS クライアントで見られる書き込みエラー (ENOSPC と EDQUOT) のロギング間隔を制御します。このパラメータは、NFS バージョン 2 と 3 のクライアントに適用されます。
データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	5 秒
範囲	32 ビットプラットフォームでは 0 から $2^{31} - 1$ 、64 ビットプラットフォームでは 0 から $2^{63} - 1$
単位	秒
動的か	はい
検査	なし
どのような場合に変更するか	クライアントによってロギングされるメッセージ量に応じてこのパラメータの値を増減します。たとえば、サーバーのファイルシステムが満杯で頻繁に使用されているときに出力される

「out of space」メッセージを減らす場合は、このパラメータの値を増やします。

安定性レベル 発展中

## nfs:nfs\_write\_error\_to\_cons\_only

説明	NFS の書き込みエラーをシステムコンソールと syslog にログGINGするか、システムコンソールだけにログGINGするかを制御します。このパラメータは、NFS バージョン 2 と 3 のクライアントに適用されます。
データ型	整数 (32 ビット)
デフォルト	0 (システムコンソールと syslog)
範囲	0 (システムコンソールと syslog) か、1 (システムコンソール)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	syslogd(1M) デーモンによってログGINGされるメッセージを含むファイルシステムがいっぱいになるのを防ぐには、このパラメータの値を調べます。このパラメータを有効にすると、メッセージはシステムコンソールに出力されるだけで、syslog メッセージファイルにはコピーされません。
安定性レベル	発展中

## nfs:nfs\_disable\_rddir\_cache

説明	NFS バージョン 2 の READDIR 要求、および NFS バージョン 3 の READDIR と READDIRPLUS
----	--

要求からの応答を格納するキャッシュの使用を制御します。このキャッシュを使用すると、ディレクトリ情報を取得するためにサーバーを繰り返し呼び出すことがなくなります。

データ型	整数 (32 ビット)
デフォルト	0 (キャッシングを有効にする)
範囲	0 (キャッシングを有効にする) か、1 (キャッシングを無効にする)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	<p>ファイルやディレクトリがサーバーに作成されたりサーバーから削除されてもサーバーがディレクトリの変更時刻を更新しないために、相互運用性に問題がある場合は、このパラメータの値を調べます。ディレクトリにファイルを追加しても新しい名前が表示されなかったり、ディレクトリからファイルを削除しても古い名前が削除されない場合は、この問題があります。</p> <p>このパラメータは、NFS バージョン 2 や 3 でマウントされたファイルシステムのキャッシングに適用されます。このパラメータは NFS でマウントされたすべてのファイルシステムに適用されるため、キャッシングをファイルシステムごとに有効にしたり、無効にしたりすることはできません。</p>
安定性レベル	発展中

## nfs:nfs3\_bsize

説明	NFS バージョン 3 クライアントによって使用される論理ブロックサイズを制御します。このブ
----	--

	ロックサイズは、クライアントが入出力を行うときにサーバーに対して読み取りや書き込みを行うデータ量を表します。
データ型	符号なし整数 (32 ビット)
デフォルト	32,768 (32K バイト)
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい。ただし、ファイルシステムのブロックサイズは、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	なし。このパラメータの設定が小さすぎたり、大きすぎたりすると、システムの動作が異常になることがあります。このパラメータには、そのプラットフォームの PAGESIZE より小さい値を設定しないでください。さらに、このパラメータの値が大きすぎると、メモリー割り当てが許可されるのを待つのでシステムがハングすることがあります。
どのような場合に変更するか	データ転送サイズの最大値を変更したい場合は、このパラメータの値を調べてください。このパラメータは、 <code>nfs3_max_transfer_size</code> パラメータと連携して変更してください。転送サイズを増やしたい場合は両方のパラメータを増やしますが、転送サイズを減らしたい場合は、通常、このパラメータを減らすだけで十分です。
安定性レベル	変更の可能性あり

## nfs:nfs\_async\_clusters

説明	<p>NFS バージョン 2 クライアントによって生成される非同期要求の組み合わせを制御します。非同期要求には、read-ahead、putpage、pageio、readdir-ahead という 4 つのタイプがあります。クライアントは、これらのタイプをラウンドロビンに処理し、特定の操作タイプだけを優遇することがないようにします。</p> <p>しかし、書き込みの一括化(書き込みをまとめる)などの NFS バージョン 2 サーバーの機能の中には、既存の NFS バージョン 2 クライアントの特定の動作に依存するものがあります。特に、この機能では、クライアントが複数の WRITE 要求をほぼ同時に送信することに依存します。キューから要求を 1 度に 1 つずつ取り出したのでは、クライアントのパフォーマンスを高めるために設けられたこのサーバー機能が生かされません。</p> <p>したがって、タイプを変更する前に、送信される各タイプの要求数の制御にこのパラメータを使用します。</p>
データ型	符号なし整数 (32 ビット)
デフォルト	1
範囲	0 から $2^{31} - 1$
単位	非同期要求
動的か	はい。ただし、ファイルシステムに対するクラスタ設定は、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	なし。ただし、このパラメータに 0 を設定すると、キューに入れられている特定タイプの要求

がすべて処理されてから、次のタイプが処理されます。これによって、アルゴリズムの公平性の部分が実質的に無効にされます。

どのような場合に変更するか	あるタイプについて次のタイプへ移行する前に生成される非同期操作の数を増やす場合は、このパラメータを変更します。これによって、クライアントからの操作のクラスタに依存するサーバーの機能が生かされる場合があります。
安定性レベル	変更の可能性あり

## nfs:nfs3\_async\_clusters

説明	<p>NFS バージョン 3 クライアントによって生成される非同期要求の組み合わせを制御します。非同期要求には、read-ahead、putpage、pageio、readdir-ahead、commit という 5 つのタイプがあります。クライアントは、これらのタイプをラウンドロビンに処理し、特定の操作タイプだけを優遇することがないようにします。</p> <p>しかし、書き込みの一括化(書き込みをまとめる)などの NFS バージョン 3 サーバーの機能の中には、既存の NFS バージョン 3 クライアントの一定の動作に依存するものがあります。特に、この機能では、クライアントが複数の WRITE 要求をほぼ同時に送信することに依存します。キューから要求を 1 度に 1 つずつ取り出したのでは、クライアントのパフォーマンスを高めるために設けられたこのサーバー機能が生かされません。</p> <p>したがって、タイプを変更する前に、送信される各タイプの要求数の制御に、このパラメータを使用します。</p>
データ型	符号なし整数 (32 ビット)
デフォルト	1

範囲	0 から $2^{31} - 1$
単位	非同期要求
動的か	はい。ただし、ファイルシステムに対するクラスタ設定は、ファイルシステムのマウント時に設定されます。特定のファイルシステムに影響を与えるには、このパラメータを変更してからそのファイルシステムをアンマウントし、再びマウントします。
検査	なし。ただし、このパラメータに 0 を設定すると、キューに入れられている特定タイプの要求がすべて処理されてから、次のタイプが処理されます。これによって、アルゴリズムの公平性の部分が実質的に無効にされます。
どのような場合に変更するか	あるタイプについて次のタイプへ移行する前に生成される非同期操作の数を増やす場合は、このパラメータを変更します。これによって、クライアントからの操作のクラスタに依存するサーバーの機能が生かされる場合があります。
安定性レベル	変更の可能性あり

## nfs:nfs\_async\_timeout

説明	非同期の入出力要求を実行するスレッドが、終了するまでにどのくらい何もせず休眠するかを表す時間を制御する。実行する要求がないと各スレッドは休眠状態に入ります。このタイマーが切れる前に新しい要求が到着しないと、スレッドは休眠から起きて終了します。新しい要求が到着すると、スレッドは休眠から起きて要求を実行し、すべての要求を処理すると、次の要求が到着するかタイマーが切れるまで再び休眠に入ります。
データ型	整数 (32 ビット)

デフォルト	6000 (1分を 60 秒 * 100Hz として表す)
範囲	0 から $2^{31} - 1$
単位	Hz (一般にクロックは 100Hz で動作する)
動的か	はい
検査	なし。ただし、このパラメータに正以外の値を設定すると、スレッドが、自身が処理する要求がキューになくなるとすぐに終了します。
どのような場合に変更するか	<p>システムでのアプリケーションの動作を正確に把握し、非同期入出力要求の割合を予測できる場合は、次のどちらかの方法によってこのパラメータをチューニングすることで、パフォーマンスをある程度最適化することができます。</p> <ul style="list-style-type: none"> <li>■ スレッドの終了までの時間を短くして、カーネルリソースの解放を早くする。</li> <li>■ スレッドの終了までの時間を長くして、スレッドの作成や破棄にかかるオーバーヘッドを減らす。</li> </ul>
安定性レベル	発展中

## nfs:nacache

説明	<p>NFS クライアント上のファイルアクセスキャッシュにアクセスするハッシュキューの数をチューニングします。ファイルアクセスキャッシュは、ユーザーがアクセスしようとするファイルに関する、ユーザーの持つファイルアクセス権を格納します。キャッシュ自体は動的に割り当てられますが、それへのインデックスに使用されるハッシュキューは静的に割り当てられます。このアルゴリズムでは、アクティブファイルごとに1つのアクセスキャッシュエントリが、ハッシュバケットごとにこれらの4つのアクセスキャッ</p>
----	--

シュエントリがあるものとみなします。したがって、このパラメータの値には、デフォルトで `nrnode` パラメータの値が設定されます。

データ型	整数 (32 ビット)
デフォルト	このパラメータのデフォルトの設定は 0 です。これは、 <code>nacache</code> の値に <code>nrnode</code> パラメータの値が設定されるべきであることを示しています。
範囲	1 から $2^{31} - 1$
単位	アクセスキャッシュエントリ
動的か	いいえ。この値は、 <code>/etc/system</code> ファイルにパラメータを追加するか、パラメータを変更し、その後システムをリブートすることによってのみ変更できます。
検査	なし。ただし、このパラメータに負の値を設定すると、システムは、おそらく、非常に多くのハッシュキューをシステムに割り当て、その間にハングします。
どのような場合に変更するか	1 つのファイルごとに 1 つのアクセスキャッシュエントリがあるという基本的な前提が損われるおそれがある場合は、このパラメータの値を検討します。複数のユーザーが同じファイルにほぼ同時にアクセスするタイムシェアリングモードのシステムでは、このような状況になることがあります。このような場合には、予想されるアクセスキャッシュのサイズを増やすことが、キャッシュへのハッシュアクセスの効率性を保つ上で役立つことがあります。
安定性レベル	発展中

## nfs:nfs3\_jukebox\_delay

説明	NFS バージョン 3 のクライアントが、前に出した要求に対して NFS3ERR_JUKEBOX エラーを受信してから、新しい要求を送信するまでにどのくらいの時間待つべきかを制御します。NFS3ERR_JUKEBOX エラーは、通常、何らかの理由でファイルが一時的に使用できないときにサーバーから返されます。このような状況は、通常、階層型記憶装置、CD やテープといったジュークボックスに関連しています。
データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	1000 (10 秒を 10 秒 * 100Hz で表す)
範囲	32 ビットプラットフォームでは 0 から $2^{31} - 1$ 64 ビットプラットフォームでは 0 から $2^{63} - 1$
単位	Hz (一般にクロックは 100Hz で動作する)
動的か	はい
検査	なし
どのような場合に変更するか	このパラメータの値を調べ、必要ならサーバーが示す動作に合わせて値を調整します。再送信を繰り返すことによるネットワークオーバーヘッドを減らすためにファイルを使用できる遅延を長くする場合は、この値を増やします。ファイルが使用可能になったことを検知する遅延を減らしたい場合は、この値を減らします。
安定性レベル	発展中

## nfs:nfs3\_max\_transfer\_size

説明	NFS バージョン 3 の READ、WRITE、READDIR、または READDIRPLUS 要求のデータ部分の最大サイズを制御します。このパラメータは、サーバーが返す要求の最大サイズとクライアントが生成する要求の最大サイズの両方を制御します。
データ型	整数 (32 ビット)
デフォルト	32,768 (32K バイト)
範囲	0 から $2^{31} - 1$
単位	バイト
動的か	はい
検査	<p>なし。ただし、サーバー側の最大転送サイズに 0 を設定すると、クライアントはおそらく、誤作動するか、単にサーバーに要求を送信しないかのどちらかになる可能性があります。</p> <p>また、UDP トランスポート経由の NFS を使用する場合にも、転送サイズの最大値に制限があります。UDP は、データグラム当たり 64K バイトの「強い」制限があります。この 64DK バイトには、要求のデータ部分の他に RPC ヘッダーやその他の NFS 情報を含まなければなりません。この制限値が大きすぎると、UDP エラーのためにクライアントとサーバーの通信に問題が発生することがあります。</p>
どのような場合に変更するか	ネットワーク上を転送するデータのサイズをチューニングする場合は、このパラメータを変更します。このパラメータを変更する場合は、通常、これに合わせて <code>nfs3_bsize</code> パラメータを変更する必要があります。たとえば、ネットワーク上のデフォルト転送サイズを 8K バイトに減らす場合は、 <code>nfs3_max_transfer_size</code> と <code>nfs3_bsize</code> パラメータの値を両方とも 8192 に変更して、それぞれで 8K バイトの読み取りま

たは書き込みが行われ、複数の操作が行われることを防ぐことができます。あるいは、32K バイトを超える値に転送サイズ増やす場合は、これに合わせて `nfs3_bsize` も変更する必要があります。変更しないと、ネットワーク上の要求サイズは変わりません。

安定性レベル

変更の可能性あり

---

## nfssrv モジュールのパラメータ

この節では、`nfssrv` モジュールの NFS パラメータについて説明します。

### `nfssrv:nfs_portmon`

説明

NFS サーバーがクライアント側の整合性を保つためにある種のセキュリティチェックを行うかどうかを制御します。要求を送信したソースポートが「予約ポート」だったかどうかをチェックできます。「予約ポート」とは、番号が 1024 より小さいものを指します。BSD ベースのシステムでは、これらのポートは、`root` で実行するプロセスのために予約されています。このチェックでは、ユーザーが独自の RPC ベースのアプリケーションを作成して、NFS クライアントが使用するアクセスチェックを破ることを防止できます。

データ型

整数 (32 ビット)

デフォルト

0 (チェックを無効にする)

範囲

0 (チェックを無効にする)、1 (チェックを有効にする)

単位

ブール値

動的か

はい

検査

なし

どのような場合に変更するか

悪意のあるユーザーが、普通ならアクセス権のない NFS サーバーを使用してファイルにアクセスするのを防ごうとする場合、このパラメータを使用します。ただし、「予約ポート」は広範にサポートされている概念ではありません。したがって、このチェックにおけるセキュリティの側面は非常に弱いものです。さらに、すべての NFS クライアントの実装で、トランスポートの終端が、予約範囲にあるポート番号にバインドされているわけではありません。したがって、このチェックを有効にすると、相互運用性に問題が生じることがあります。

安定性レベル

発展中

## nfssrv:rfs\_write\_async

説明

NFS バージョン 2 サーバーが WRITE 要求をどのように処理するかを制御します。NFS バージョン 2 プロトコルでは、WRITE 要求に関連するすべての変更済みデータとメタデータが安定したストレージに格納されていないと、サーバーはクライアントに応答できません。NFS バージョン 2 の WRITE 要求は、データは 8192 バイトに制限されます。したがって、各 WRITE 要求によって、複数の小さい書き込みがストレージサブシステムに対して行われることがあります。これは、パフォーマンス低下の原因になります。

NFS バージョン 2 の WRITE を高速化する 1 つの方法は、クライアントの動作を利用することです。クライアントは、複数の WRITE 要求をバッチで (一括して) 送信する傾向があります。サーバーでは、この動作を利用して複数の WRITE 要求を 1 つの要求にクラスタ化し、ファイルシステムに出すことができます。こうして、要求の数を少なくし、要求のサイズを大きくして、

データをストレージサブシステムに書き込むことができます。これにより、WRITE 要求のスループットが大幅に向上することがあります。

データ型	整数 (32 ビット)
デフォルト	1 (クラスタ化を有効にする)
範囲	0 (クラスタ化を有効にする)、1 (クラスタ化を無効にする)
単位	ブール値
動的か	はい
検査	なし
どのような場合に変更するか	特に PC クライアントなど、非常に小さい NFS クライアントのなかには、複数の WRITE 要求をバッチ化しないものがあります。この場合には、クライアント側から要求される動作が存在しないため、NFS バージョン 2 サーバーで要求をクラスタ化しても、オーバーヘッドが増えるだけで、パフォーマンスがかえって低下することがあります。
安定性レベル	発展中

## nfssrv:nfsauth\_ch\_cache\_max

説明	NFS 認証サーバーに接続するクライアントハンドルのキャッシュサイズを制御します。サーバーは、NFS クライアントを認証して、クライアントが使用しようとしているファイルハンドルへのアクセスが許可できるかどうかを判定します。
データ型	整数 (32 ビット)
デフォルト	16
範囲	0 から $2^{31} - 1$

単位	クライアントハンドル
動的か	はい
検査	なし
どのような場合に変更するか	このキャッシュは動的でないため、すべてのクライアントハンドルが使用されていると、クライアントハンドルの割り当て要求は失敗します。これは NFS サーバーが、この要求を認証できなかったという理由で落とすことを意味します。しかし、ほとんどの場合、NFS クライアントはタイムアウトになり、要求を再転送するため、これは問題になりません。しかし、クライアントにソフトマウントされたファイルシステムの場合、クライアントは、タイムアウトになっても要求を再試行せず、アプリケーションにエラーを返すことがあります。サーバーのキャッシュを負荷に対応できるだけの大きさにしておけば、このような状況を防止することができます。
安定性レベル	変更の可能性あり

## nfssrv:exi\_cache\_time

説明	システムのメモリー圧迫によって消去される前に NFS 認証キャッシュにエントリをどのくらいの間保持するかを制御します。
データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	3600 秒 (1 時間)
範囲	32 ビットプラットフォームでは 0 から $2^{31} - 1$ 、64 ビットプラットフォームでは 0 から $2^{63} - 1$
単位	秒
動的か	はい

検査	なし
どのような場合に変更するか	NFS 認証キャッシュのサイズは、エントリがキャッシュから消去されるまでの最小限の時間を変更することによって調整することができます。キャッシュのサイズは、大きくなりすぎないように制御すべきです。そうすれば、この時間のプロセスによって開放されなかったシステムリソースを使用できます。
安定性レベル	発展中

---

## rpcmod モジュールのパラメータ

この節では、rpcmod モジュールの NFS パラメータについて説明します。

### rpcmod:clnt\_max\_conns

説明	NFS クライアントが各 NFS サーバーと通信するときに使用する TCP 接続の数を制御します。カーネル RPC は、1 つの接続で RPC を多重化して構築しますが、必要なら複数の接続で使用することもできます。
データ型	整数 (32 ビット)
デフォルト	1
範囲	1 から $2^{31} - 1$
単位	接続
動的か	はい
検査	なし
どのような場合に変更するか	一般には、1 つの接続だけでネットワーク帯域幅全体を使いこなすことができます。しかし、ネットワークが提供する帯域幅を TCP が 1 つのスト

リームだけで利用できない場合は、複数の接続を使えば、クライアントとサーバー間のスループットが向上することがあります。

接続数の増加には犠牲が伴います。接続数を増加すると、個々の接続を管理するために、より多くのカーネルリソースの使用が必要になります。

安定性レベル 発展中

## rpcmod:clnt\_idle\_timeout

説明 クライアントとサーバー間の接続がクライアント側でどのくらいの間遊休であれば、接続をクローズするかを制御します。

データ型 long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)

デフォルト 300,000 ミリ秒 (5 分)

範囲 32 ビットプラットフォームでは 0 から  $2^{31} - 1$ 、64 ビットプラットフォームでは 0 から  $2^{63} - 1$

単位 ミリ秒

動的か はい

検査 なし

どのような場合に変更するか クライアント側でどのくらいの間遊休であれば接続をクローズするかを変更したい場合は、このパラメータを使用します。システムリソースが浪費されるのを防ぐために、遊休接続をクローズする時間を短縮したい場合などです。

安定性レベル 発展中

## rpcmod:svc\_idle\_timeout

説明	クライアントとサーバー間の接続がサーバー側でどのくらいの間遊休であれば接続をクローズするかを制御します。
データ型	long 整数 (32 ビットプラットフォームでは 32 ビット、64 ビットプラットフォームでは 64 ビット)
デフォルト	360,000 ミリ秒 (6 分)
範囲	32 ビットプラットフォームでは 0 から $2^{31} - 1$ 、64 ビットプラットフォームでは 0 から $2^{63} - 1$
単位	ミリ秒
動的か	はい
検査	なし
どのような場合に変更するか	サーバー側でどのくらいの間遊休であれば接続をクローズするかを変更したい場合は、このパラメータを使用します。必要な場合は、システムリソースの浪費を防ぐために遊休接続をクローズする時間を短縮します。
安定性レベル	発展中

## rpcmod:svc\_default\_stksize

説明	カーネル RPC のサービススレッドに対するカーネルスタックのサイズを設定します。
データ型	整数 (32 ビット)
デフォルト	デフォルト値 0 は、スタックサイズにシステムのデフォルト値を設定することを表します。
範囲	0 から $2^{31} - 1$
単位	バイト

動的か	スタックサイズはスレッドの作成時に設定されます。したがって、このパラメータの変更は、既存のスレッドには適用されず、新しく割り当てられるすべてのスレッドに適用されます。
検査	なし
どのような場合に変更するか	呼び出し深度が非常に深いために、スタックがオーバーフローし、レッドゾーン障害が発生するおそれがある場合。トランスポートに対する呼び出し深度が比較的深く、ローカルファイルシステムに対する呼び出しの深さが深いという組み合わせは、NFS サービススレッドのスタックがオーバーフローを起こすことがあります。  このパラメータには、プラットフォームのハードウェア pagesize の倍数を設定する必要があります。
安定性レベル	発展中

## rpcmod:svc\_default\_max\_same\_xprt

説明	各トランスポート終端の要求を最大でいくつ処理したら、次のトランスポート終端に進むかを制御します。カーネル RPC では、サービススレッドのプールとトランスポート終端のプールが使用されます。個々のサービススレッドは、どのトランスポート終端からの要求でも処理できます。ただし、パフォーマンス上の理由により、次のトランスポート終端に進む前に各トランスポート終端の複数の要求が処理されます。このアプローチにより、不足を避け、パフォーマンス上の利点を得ることができます。
データ型	整数 (32 ビット)
デフォルト	8
範囲	0 から $2^{31} - 1$

単位	要求
動的か	はい。ただし、トランスポート終端を切り替える前に要求を最大でいくつ処理するかは、トランスポート終端がカーネル RPC サブシステムに構成されるときに設定されます。このパラメータへの変更は、新しいトランスポート終端だけに適用されます。つまり、既存のトランスポート終端には無効です。
検査	なし
どのような場合に変更するか	サービスが、NFS バージョン 2 の WRITE 要求を高速化するクラスタ化などのクライアントの動作を利用できるようにこの値をチューニングすることができます。このパラメータの値を増やすことにより、サーバー側でクライアントの動作の利点をよりよく利用できる可能性があります。
安定性レベル	発展中

## rpcmod:maxdupreqs

説明	コネクションレストランスポートにおける RPC レベルの再転送を検出する重複要求キャッシュのサイズを制御します。このキャッシュは、クライアントネットワークアドレス、RPC の手順番号、プログラム番号、バージョン番号、および、トランザクション ID でインデックス化されます。このキャッシュにより、非べき等であるかもしれない再転送要求の処理が防止されます。
データ型	整数 (32 ビット)
デフォルト	1024
範囲	1 から $2^{31} - 1$
単位	要求

動的か	<p>キャッシュのサイズは動的に決められますが、キャッシュへの高速アクセスを可能にするハッシュキューのサイズは静的に決められます。キャッシュのサイズを著しく大きくすると、キャッシュ内のエントリの検索に長い時間がかかることがあります。</p> <p>このパラメータに 0 を設定しないでください。0 を設定すると、NFS サーバーが非べき等の要求を処理できません。</p>
検査	なし
どのような場合に変更するか	<p>NFS クライアントで不正な障害エラーが見られる場合は、このパラメータの値を調べます。たとえば、ディレクトリの作成が失敗したのに、実際にはディレクトリが作成されている場合は、再転送された MKDIR 要求をサーバーが検出しなかった可能性があります。</p> <p>キャッシュのサイズは、サーバーの負荷に見合ったものでなければなりません。キャッシュには非べき等の要求が格納されるため、キャッシュでは、要求全体の一部だけしか管理する必要がありません。キャッシュは、クライアント側の再転送を検出できるだけの間、情報を保持していなければなりません。一般に、コネクションレスタランスポートのクライアントのタイムアウトは比較的短く、1 秒から 20 秒くらいです。</p>
安定性レベル	変更の可能性あり

## rpcmod:cotsmaxdupreqs

説明	<p>コネクション型のトランスポートにおける RPC レベルの再転送を検出する重複要求キャッシュのサイズを制御します。このキャッシュは、クライアントネットワークアドレス、RPC の手順番号、プログラム番号、バージョン番号、および、トランザクション ID でインデックス化され</p>
----	---

ています。このキャッシュにより、非べき等であるかもしれない再転送要求の処理が防止されます (呼び出し回数に依存する)。

データ型	整数 (32 ビット)
デフォルト	1024
範囲	1 から $2^{31} - 1$
単位	要求
動的か	はい
検査	<p>キャッシュのサイズは動的に決められますが、キャッシュへの高速アクセスを可能にするハッシュキューのサイズは静的に決められます。キャッシュのサイズを著しく大きくすると、キャッシュ内のエントリの検索に長い時間がかかることがあります。</p> <p>このパラメータに 0 を設定しないでください。0 を設定すると、NFS サーバーは非べき等の要求を処理できません。</p>
どのような場合に変更するか	<p>NFS クライアントで不正な障害エラーが見られる場合は、このパラメータの値を調べます。たとえば、ディレクトリの作成が失敗したのに、実際にはディレクトリが作成されている場合は、再転送された MKDIR 要求をサーバーが検出しなかった可能性があります。</p> <p>キャッシュのサイズは、サーバーの負荷に見合ったものでなければなりません。キャッシュには非べき等の要求が格納されるため、キャッシュでは、要求全体の一部だけしか管理する必要がありません。キャッシュは、クライアント側の再転送を検出できるだけの間、情報を保持していなければなりません。一般に、コネクション型のトランスポートのクライアントのタイムアウトは非常に長く、1 分くらいです。し</p>

たがって、エント리는、キャッシュに比較的長く留まる必要があります。

安定性レベル

変更の可能性あり

## TCP/IP チューニング可能パラメータ

---

この章では、TCP/IP のチューニング可能パラメータについて説明します。カーネルのチューニング可能パラメータについては第 2 章を、NFS のチューニング可能パラメータについては第 3 章 をそれぞれ参照してください。

- 151ページの「IP チューニング可能パラメータ」
- 156ページの「TCP チューニング可能パラメータ」
- 173ページの「UDP チューニング可能パラメータ」
- 175ページの「ルート別のメトリック」

---

## TCP/IP パラメータのチューニングの概要

この章で述べるすべてのチューニング可能パラメータは、次の 2 つのパラメータを除き `ndd` コマンドで設定できます。次の 2 つのパラメータは、`/etc/system` ファイルでのみ設定できます。

- 167ページの「`tcp_conn_hash_size`」
- 168ページの「`ipc_tcp_conn_hash_size`」

`ndd` コマンドで TCP/IP パラメータを設定する場合は、次の構文を使用します。

```
# ndd -set driver parameter
```

たとえば、次の `ndd` コマンドでは IP 転送を無効にします。

```
# ndd -set /dev/ip ip_forwarding 0
```

詳細は、`ndd(1M)` のマニュアルページを参照してください。

システムのリブート後も TCP/IP パラメータの設定を有効にするには、適切な `ndd` コマンドをシステム起動スクリプトに指定します。`ndd` コマンドを含むシステム起動スクリプトの作成には、次のガイドラインを使用してください。

- スクリプトを `/etc/init.d` ディレクトリに作成し、それへのリンクを `/etc/rc2.d`、`/etc/rc1.d`、`/etc/rcS.d` ディレクトリにそれぞれ作成します。
- このスクリプトは、既存の `S69inet` スクリプトと `S72inetsvc` スクリプトの間で実行する必要があります。
- スクリプトに `S70` か `S71` 接頭辞を付けます。同じ接頭辞を持つスクリプトはある種の方法で順に実行されますので、同じ接頭辞を持つスクリプトが複数あっても問題はありません。
- 実行制御スクリプトの名前付けについての詳細は、`/etc/init.d` ディレクトリの `README` ファイルを参照してください。

起動スクリプトの作成方法についての詳細は、『*Solaris* のシステム管理 (第 1 巻)』の「実行制御スクリプト」を参照してください。

## TCP/IP パラメータの検査

この節で記述する TCP/IP パラメータのうち、前述の `/etc/system` ファイルでのみ設定できる 2 つのパラメータの検査については、167ページの

「`tcp_conn_hash_size`」と 168ページの「`ipc_tcp_conn_hash_size`」の「検査」の項を参照してください。その他のパラメータについては、各パラメータについての説明中の「範囲」の項で述べる範囲に入っているかどうかを検査されます。

## Internet Request for Comments (RFC)

インターネットのプロトコルと標準の仕様は、RFC 文書に記述されています。RFC のコピーは、`sri-nic.arpa` マシンから匿名 `ftp` を使用して入手できます。このサイトの `rfc-index.txt` ファイル中で RFC のトピックを探してください。

## IP チューニング可能パラメータ

この節では、IP チューニング可能パラメータの一部を説明します。

### `ip_icmp_err_interval` と `ip_icmp_err_burst`

説明	IP で IPv4 または IPv6 ICMP エラーメッセージを生成する頻度を制御します。IP は、 <code>ip_icmp_err_interval</code> の間に最大で <code>ip_icmp_err_burst</code> の IPv4 または IPv6 ICMP エラーメッセージを生成します。このパラメータは、サービス拒否攻撃から IP を守るためのものです。 <code>ip_icmp_err_interval</code> に 0 を設定すると、IP で IPv4 または IPv6 ICMP エラーメッセージは生成されません。
デフォルト	<code>ip_icmp_err_interval</code> は 100 ミリ秒、 <code>ip_icmp_err_burst</code> は 10
範囲	<code>ip_icmp_err_interval</code> は 0 から 99,999 ミリ秒、 <code>ip_icmp_err_burst</code> は 1 から 99,999
動的か	はい
どのような場合に変更するか	診断の目的でエラーメッセージの生成頻度を増やしたい場合は、このパラメータ値を変更します。
コミットレベル	変更の可能性あり

### `ip_forwarding` と `ip6_forwarding`

説明	IP がインタフェース間で IPv4 または IPv6 転送を行うかどうかを制御します。 <code>xxx:ip_forwarding</code> も参照してください。
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)

動的か	はい
どのような場合に変更するか	IP 転送が必要な場合は、これを有効にします。
コミットレベル	変更の可能性あり

### `xxx:ip_forwarding`

説明	特定の <code>xxx</code> インタフェースに対し IPv4 転送を有効にします。このパラメータの正確な表記は <code>interface-name:ip_forwarding</code> です。たとえば、2つのインタフェース <code>hme0</code> と <code>hme1</code> がある場合、それぞれのパラメータ名は <code>hme0:ip_forwarding</code> と <code>hme1:ip_forwarding</code> です。
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に変更するか	IPv4 転送が必要な場合は、このパラメータを使用してインタフェースごとに転送を有効にします。
コミットレベル	変更の可能性あり

### `ip_respond_to_echo_broadcast` と `ip6_respond_to_echo_multicast`

説明	IPv4 や IPv6 で、ブロードキャスト ICMPv4 エコー要求またはマルチキャスト ICMPv6 エコー要求に応答するかどうかを制御します。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい

どのような場合に変更するか	セキュリティ上の理由でこの動作を行いたくない場合は、無効にします。
コミットレベル	変更の可能性あり

## ip\_send\_redirects と ip6\_send\_redirects

説明	IPv4 または IPv6 で ICMPv4 または ICMPv6 リダイレクトメッセージを送信するかどうかを制御します。151ページの「ip_forwarding と ip6_forwarding」も参照してください。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に変更するか	セキュリティ上の理由でこの動作を行いたくない場合は、無効にします。
コミットレベル	変更の可能性あり

## ip\_forward\_src\_routed と ip6\_forward\_src\_routed

説明	IPv4 または IPv6 で、パケットをソース IPv4 ルーティングオプションを指定してまたは IPv6 ルーティングヘッダーを指定して転送するかどうかを制御します。151ページの「ip_forwarding と ip6_forwarding」も参照してください。
デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい

どのような場合に変更するか	セキュリティ上の理由でこの動作を行いたくない場合は、無効にします。
コミットレベル	変更の可能性あり

## ip\_addr\_per\_if

説明	実インタフェースに対応する論理インタフェースの最大数
デフォルト	256
範囲	1 から 8192
動的か	はい
どのような場合に変更するか	この値は変更しないでください。ただし、論理インタフェースの数を増やす必要がある場合は、この値を増加させます。変更によって IP のパフォーマンスに悪影響が生ずることがあることに留意してください。
コミットレベル	変更の可能性あり

## ip\_strict\_dst\_multihoming と ip6\_strict\_dst\_multihoming

説明	非転送インタフェースに到着したパケットを、そのインタフェースに明示的に構成されていない IP アドレス向けとして受け入れるかどうかを制御します。ip_forwarding が有効になっているか、該当するインタフェースに対し xxx:ip_forwarding が有効になっていると、このパラメータは無視されます (そのパケットが実際に転送されるため)。  RFC 1122 3.3.2.4 を参照してください。
デフォルト	0 (緩やかなマルチホーミング)

範囲	0 = オフ (緩やかなマルチホーミング)、1 = オン (厳密なマルチホーミング)
動的か	はい
どのような場合に変更するか	厳密なネットワークドメイン (たとえばファイアウォールや VPN ノードなど) を通過するインタフェースがマシンにある場合は、この変数に 1 を設定します。
コミットレベル	変更の可能性あり

## 特別な注意を要する IP チューニング可能パラメータ

各パラメータで記述する特別な状況である場合以外は、次のパラメータを変更することは推奨されません。

### `ip_ire_pathmtu_interval`

説明	IP がパス最大転送単位 (PMTU) 検出情報をフラッシュしてから PMTU を再び検出開始するまでの間隔をミリ秒単位で指定します。  PMTU の検出については、RFC 1191 を参照してください。
デフォルト	10 分
範囲	5 秒から 277 時間
動的か	はい
どのような場合に変更するか	この値は変更しないでください。
コミットレベル	変更の可能性あり

### `ip_icmp_return_data_bytes` と `ip6_icmp_return_data_bytes`

説明	IPv4 や IPv6 は、ICMPv4 や ICMPv6 のエラーメッセージを送信するときに、エラーメッセージの原因になったパケットの IP ヘッダーを含め
----	---

ます。このパラメータでは、パケットのうち IPv4 や IPv6 のヘッダーを除いてあと何バイトを ICMPv4 や ICMPv6 のエラーメッセージに含めるかを制御します。

デフォルト	64 バイト
範囲	8 から 65,536 バイト
動的か	はい
どのような場合に変更するか	この値は変更しないでください。ただし、ICMP エラーメッセージに含む情報を増やすと、ネットワークの問題を診断する上で役立つことがあるため、この機能が必要な場合はこの値を増やします。
コミットレベル	変更の可能性あり

---

## TCP チューニング可能パラメータ

### `tcp_deferred_ack_interval`

説明	TCP 遅延肯定応答 (ACK) タイマーのタイムアウト値をミリ秒単位で指定します。  RFC 1122, 4.2.3.2 を参照してください。
デフォルト	100 ミリ秒
範囲	1 ミリ秒から 1 分
動的か	はい
どのような場合に変更するか	このパラメータには、500 ミリ秒を超える値を設定しないでください。  ネットワークリンクが遅く (57.6 Kbps 未満)、最大セグメントサイズ (MSS) が 512 バイトを超え、この間隔が複数の TCP セグメントを受信

するには短すぎるような状況では、この値を増やします。

コミットレベル 変更の可能性あり

## tcp\_deferred\_acks\_max

説明 TCP セグメントを最大でいくつ (個々の接続ごとに最大セグメントサイズ MSS の単位で) 受信したら肯定応答 (ACK) を生成するかを指定します。このパラメータに 0 か 1 を設定すると、すべてのセグメントが 1 MSS の長さであるとみなされ、遅延 ACK はないことになります。

直接的に接続されていないリモート宛先の場合は、このパラメータの値が何であれ、最大値は 2 に固定されます。実際の数値は、接続ごとに動的に計算されます。この値はデフォルトの最大値です。

デフォルト 8

範囲 0 から 16

動的か はい

どのような場合に変更するか この値は変更しないでください。ただし、遅延 ACK の影響でネットワークトラフィックが著しく混雑するような状況では、この値を減らします。ただし、2 より小さくしないでください。

コミットレベル 変更の可能性あり

## tcp\_wscale\_always

説明 1 が設定されていると、TCP は、オプションの値が 0 であっても、常にウィンドウスケールオプションを指定して SYN セグメントを送信します。TCP は、ウィンドウスケールオプションの

指定された SYN セグメントを受信すると、このパラメータに 0 が設定されていても、ウィンドウスケールオプションを指定して SYN セグメントに返信し、またオプションの値は受信ウィンドウサイズに従って設定されることに留意してください。

ウィンドウスケールオプションについては、RFC 1323 を参照してください。

デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に変更するか	高速ネットワーク構成でウィンドウスケールオプションを使用したい場合は、このパラメータを有効にします。
コミットレベル	変更の可能性あり

## tcp\_tstamp\_always

説明  
1 が設定されていると、TCP は常に、タイムスタンプオプションを指定して SYN セグメントを送信します。TCP は、タイムスタンプオプションの指定された SYN セグメントを受信すると、このパラメータに 0 が設定されていても、タイムスタンプオプションを指定して SYN セグメントに返信することに留意してください。

デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に変更するか	要約すると、往復時間 (RTT) や TCP シーケンス番号ラップアラウンドを正確に測定したい場合に、これを有効にします。

このオプションを有効にする理由についての詳細は、RFC 1323 を参照してください。

コミットレベル 変更の可能性あり

## tcp\_xmit\_hiwat

説明 デフォルトの送信ウィンドウサイズをバイト数で指定します。ルートごとに異なる値を設定する方法については、後で述べるルート別のメトリックの説明を参照してください。160ページの「tcp\_max\_buf」も参照してください。

デフォルト 16,384 バイト

範囲 4096 から 1,073,741,824

動的か はい

どのような場合に変更するか これはデフォルト値です。アプリケーションでは、setsockopt(3SOCKET) SO\_SNDBUF を使用して、送信バッファを接続ごとに変更できます。

コミットレベル 変更の可能性あり

## tcp\_recv\_hiwat

説明 デフォルトの受信ウィンドウサイズをバイト数で指定します。ルートごとに異なる値を設定する方法については、後で述べるルート別のメトリックの説明を参照してください。172ページの「tcp\_recv\_hiwat\_minmss」と160ページの「tcp\_max\_buf」も参照してください。

デフォルト 24,576

範囲 2048 から 1,073,741,824

動的か はい

どのような場合に変更するか      これはデフォルト値です。アプリケーションでは、`setsockopt(3SOCKET) SO_RCVBUF` を使用して、受信バッファを接続ごとに変更できます。

コミットレベル      変更の可能性あり

## tcp\_max\_buf

説明      バッファサイズの最大値をバイト数で指定します。これは、アプリケーションが `setsockopt(3SOCKET)` を使用して設定する送信バッファや受信バッファのサイズの最大値を制御します。

デフォルト      1,048,576

範囲      8192 から 1,073,741,824

動的か      はい

どのような場合に変更するか      高速ネットワーク環境で TCP 接続を行う場合は、ネットワークリンクの速度に合わせて値を増やします。

コミットレベル      変更の可能性あり

## tcp\_cwnd\_max

説明      TCP 輻輳ウィンドウ (cwnd) のサイズの最大値をバイト数で指定します。

TCP 輻輳ウィンドウについては、RFC 1122 と RFC 2581 を参照してください。

デフォルト      1,048,576

範囲      128 から 1,073,741,824

動的か      はい

どのような場合に変更するか	これは、TCP <code>cwnd</code> をどこまで拡張できるかを示す最大値です。アプリケーションが <code>setsockopt (3SOCKET)</code> を使用してウィンドウサイズに <code>tcp_cwnd_max</code> より大きい値を設定しようとしても、使用される実際のウィンドウが <code>tcp_cwnd_max</code> を超えることはありません。したがって、一般には、 <code>tcp_max_buf</code> は <code>tcp_cwnd_max</code> より大きくなければなりません。
コミットレベル	変更の可能性あり

## `tcp_slow_start_initial`

説明	輻輳ウィンドウ ( <code>cwnd</code> ) の初期サイズの最大値を TCP 接続の MSS 単位で指定します。  輻輳ウィンドウの初期サイズがどのように計算されるかについては、RFC 2414 を参照してください。
デフォルト	4
範囲	1 から 4
動的か	はい
どのような場合に変更するか	この値は変更しないでください。  特殊な状況下で <code>cwnd</code> の初期サイズがネットワークの輻輳を招く場合は、この値を減らします。
コミットレベル	変更の可能性あり

## `tcp_slow_start_after_idle`

説明	輻輳ウィンドウが 1 再送タイムアウト (RTO) の間遊休にされた (セグメントを全く受信しなかった) 後の、輻輳ウィンドウのサイズを TCP 接続の MSS 単位で指定します。
----	--

この計算については、RFC 2414 を参照してください。

デフォルト	4
範囲	1 から 16,384
動的か	はい
どのような場合に変更するか	詳細は、161ページの 「tcp_slow_start_initial」を参照してください。
コミットレベル	変更の可能性あり

## tcp\_sack\_permitted

説明	2 が設定されていると、TCP は常に、選択的肯定応答 (SACK) 許可オプションを指定して SYN セグメントを送信します。TCP は、SACK 許可オプションを指定した SYN セグメントを受信し、かつこのパラメータに 1 が設定されていると、SACK 許可オプションを指定して応答します。このパラメータに 0 が設定されていると、着信セグメントに SACK 許可オプションが指定されているかどうかにかかわらず、TCP は SACK 許可オプションを送信しません。  SACK オプションについては、RFC 2018 を参照してください。
デフォルト	2 (アクティブに有効にする)
範囲	0 (無効)、1 (パッシブに有効にする)、2 (アクティブに有効にする)
動的か	はい
どのような場合に変更するか	SACK 処理を行うと TCP 再送のパフォーマンスが向上するため、アクティブに有効にすべきです。SACK オプションをアクティブに有効にす

ると相手方が混乱するおそれがあるような場合は、1を設定します。それによって SACK 処理は、着信接続で SACK 処理が許可されているときだけ行われます。

コミットレベル 変更の可能性あり

## tcp\_rev\_src\_routes

説明 0 が設定されていると、TCP は、セキュリティ上の理由により、着信接続に対し IP ソースルーティングオプションを逆方向に使用しません。1 が設定されている場合は、TCP は通常どおりソースルーティングを逆方向に使用します。

デフォルト 0 (無効)

範囲 0 (無効)、1 (有効)

動的か はい

どのような場合に変更するか 診断のために IP ソースルーティングが必要な場合は、有効にします。

コミットレベル 変更の可能性あり

## tcp\_time\_wait\_interval

説明 TCP 接続を TIME-WAIT 状態に保つ時間をミリ秒で指定します。

詳細は、RFC 1122, 4.2.2.13 を参照してください。

デフォルト 4 分

範囲 1 秒から 10 分

動的か はい

どのような場合に変更するか	負荷の高い Web サーバーでは、あまりに多くの TCP 接続が TIME-WAIT 状態になっているために、メモリーが過度に使用されていることがあります。このような場合には、パフォーマンスの向上のために、この値を減らすことができます。ただし、この値は 60 秒より小さくしないでください。  詳細は、RFC 1122, 4.2.2.13 を参照してください。
コミットレベル	変更の可能性あり

## tcp\_conn\_req\_max\_q

説明	accept(3SOCKET) によって受け付けられるのを待っている保留状態の TCP 接続を 1 つの TCP リスナー当たり最大でいくつ持てるかのデフォルト数を指定します。165 ページの「tcp_conn_req_max_q0」も参照してください。
デフォルト	128
範囲	1 から 4,294,967,296
動的か	はい
どのような場合に変更するか	いくつかの接続要求を受信することがある Web サーバーのようなアプリケーションでは、着信頻度に応じてこのデフォルト値を増やすことができます。  このパラメータに著しく大きい値を設定しないでください。保留状態の TCP 接続はメモリーを過度に使用することがあります。さらに、保留状態の TCP 接続の数が多すぎて接続要求を適時に処理できるほどアプリケーションが速くない場合は、新しい着信要求が拒否されることがあります。

`tcp_conn_req_max_q`を増やしても、アプリケーションでそれだけの数の保留状態のTCP接続を持てるとは限りません。アプリケーションでは、`listen(3SOCKET)`を使用して、保留状態のTCP接続の最大数をソケットごとに変更できます。このパラメータは、アプリケーションが`listen()`を使用して設定できる最大値を表しています。つまり、このパラメータに非常に大きな値を設定しても、あるソケットに対する実際の最大数は、`listen()`に指定された値によっては`tcp_conn_req_max_q`よりもはるかに少ないことがあります。

コミットレベル

変更の可能性あり

## `tcp_conn_req_max_q0`

説明

不完全な(3段階ハンドシェークがまだ終わっていない)保留状態のTCP接続を1つのTCPリスナー当たりいくつ持てるかのデフォルトの最大数を指定します。

TCPの3段階ハンドシェークについての詳細は、RFC 793を参照してください。さらに、164ページの「`tcp_conn_req_max_q`」も参照してください。

デフォルト

1024

範囲

0 から 4,294,967,296

動的か

はい

どのような場合に変更するか

極めて多くの接続要求を受信することがあるWebサーバーのようなアプリケーションでは、着信頻度に応じてこのデフォルト値を増やすことができます。

tcp\_conn\_req\_max\_q0 と、各ソケットについて保留状態にある接続の最大数との関係は、次のとおりです。

接続要求を受信すると、TCP はまず、受け付けられるのを待っている保留状態の TCP 接続 (3 段階ハンドシェイクが終わっている) の数 (N) が、そのリスナーに対する最大数を超えていないかをチェックします。接続数が超えていると、その要求は拒否されます。超えていなければ、TCP は、不完全な保留状態の TCP 接続の数が、N と tcp\_conn\_req\_max\_q0 の合計を超えていないかをチェックします。超えていなければ、その要求は受け付けられます。超えていると、最も古い不完全な保留状態の TCP 要求がドロップされます。

コミットレベル

変更の可能性あり

## tcp\_conn\_req\_min

説明

受け付けられるのを待っている保留状態の TCP 接続の最大数 (1 つのリスナーについての) の、デフォルトの最小値。これは、1 つのアプリケーションが使用できる listen(3SOCKET) の最も低い最大値です。

デフォルト

1

範囲

1 から 1024

動的か

はい

どのような場合に変更するか

このパラメータにより、listen(3SOCKET) を使用するアプリケーションが保留状態の TCP 接続の最大数を過度に小さく設定するのを防ぐことができます。この値を着信接続要求の頻度に応じて増やしてください。

コミットレベル

変更の可能性あり

## /etc/system ファイルに設定する TCP パラメータ

これらのパラメータは、/etc/system ファイルでしか設定できません。ファイルを変更してからシステムをリブートする必要があります。

次のエントリは tcp\_conn\_hash\_size の設定例です。

```
set tcp:tcp_conn_hash_size=1024
```

### tcp\_conn\_hash\_size

説明	すべての TCP 接続に対する、TCP モジュールのハッシュテーブルサイズを制御します。
データ型	符号付き整数
デフォルト	512
範囲	512 から 1,073,741,824
暗黙的制約	この値には 2 のべき乗を指定すべきです。
動的か	いいえ。このパラメータはブート時にのみ変更できます。
検査	パラメータの値が 2 のべき乗でないと、一番近い 2 のべき乗に丸められます。
どのような場合に変更するか	何万に上る TCP 接続が継続的にあるシステムでは、この値をそれに応じて増やします。TCP は、デフォルト値で 2 ~ 3 千のアクティブ接続までは十分にサポートできます。ハッシュテーブルのサイズを増やすと、それだけ多くのメモリーが必要になりますので、適切な値を指定するようにしてメモリーが不必要に使用されるのを防いでください。
コミットレベル	変更の可能性あり

## ipc\_tcp\_conn\_hash\_size

説明	すべてのアクティブ (ESTABLISHED 状態にある) TCP 接続に対する、IP モジュールのハッシュテーブルサイズを制御します。
データ型	符号なし整数
デフォルト	512
範囲	512 から 2,147,483,648
暗黙的制約	この値は 2 のべき乗を指定すべきです。
動的か	いいえ。このパラメータはブート時のみ変更することができます。
検査	パラメータの値が 2 のべき乗でないと、一番近い 2 のべき乗に丸められます。
どのような場合に変更するか	何万に上るアクティブ TCP 接続が継続的にあるシステムでは、この値をそれに応じて増やします。TCP は、デフォルト値で 2 ~ 3 千までのアクティブ接続を十分にサポートできます。ハッシュテーブルのサイズを増やすと、それだけ多くのメモリーが必要になりますので、適切な値を指定するようにしてメモリーが不必要に使用されるのを防いでください。
コミットレベル	変更の可能性あり

## 特別な注意を要する TCP パラメータ

各パラメータで記述する特別な状況である場合以外は、次のパラメータを変更することは推奨されません。

## tcp\_ip\_abort\_interval

説明	TCP 接続に対するデフォルトのトータルの再送タイムアウト値をミリ秒で指定します。1 つの TCP 接続について、TCP が
----	--

tcp\_ip\_abort\_interval の間再転送を行ってもこの間に相手側のエンドポイントから肯定応答を全く受け取らないと、TCP はこの接続をクローズします。

TCP 再送タイムアウト (RTO) の計算については、RFC 1122, 4.2.3 を参照してください。170 ページの「tcp\_rexmit\_interval\_max」も参照してください。

デフォルト	8 分
範囲	500 ミリ秒から 1193 時間
動的か	はい
どのような場合に変更するか	この値は変更しないでください。例外については、170ページの「tcp_rexmit_interval_max」を参照してください。
コミットレベル	変更の可能性あり

### tcp\_rexmit\_interval\_initial

説明	1 つの TCP 接続に対するデフォルトの初期再送タイムアウト値 (RTO) をミリ秒で指定します。ルートごとに異なる値を設定する方法については、後で述べるルート別メトリックの説明を参照してください。
デフォルト	3 秒
範囲	1 ミリ秒から 20 秒
動的か	はい
どのような場合に変更するか	この値は変更しないでください。この値を小さくすると、不要な再転送が行われるおそれがあります。
コミットレベル	変更の可能性あり

## tcp\_rexmit\_interval\_max

説明	デフォルトの最大再送タイムアウト値 (RTO) をミリ秒で指定します。すべての TCP 接続に対して、計算された RTO がこの値を超えることはできません。168ページの「tcp_ip_abort_interval」も参照してください。
デフォルト	60 秒
範囲	1 ミリ秒から 20 秒
動的か	はい
どのような場合に変更するか	通常のネットワーク環境では、この値を変更しないでください。  1 つの接続に対する往復時間 (RTT) が数 10 秒になるような特別な状況では、この値を増やすことができます。この値を変更する場合は、これに合わせて tcp_ip_abort_interval パラメータも変更すべきです。tcp_ip_abort_interval には、tcp_rexmit_interval_max の 4 倍以上の値を指定します。
コミットレベル	変更の可能性あり

## tcp\_rexmit\_interval\_min

説明	デフォルトの最小再送タイムアウト値 (RTO) をミリ秒で指定します。すべての TCP 接続に対して、計算された RTO がこの値を下回ることにはできません。170ページの「tcp_rexmit_interval_max」も参照してください。
デフォルト	400 ミリ秒
範囲	1 ミリ秒から 20 秒
動的か	はい

どのような場合に変更するか	通常のネットワーク環境では、この値を変更しないでください。  TCP の RTO 計算は、RTT の最も大きい変動に対処できなければなりません。1 つの接続に対する往復時間 (RTT) が数 10 秒になるような特別な状況では、この値を増やすことができます。この値を変更する場合は、これに合わせて <code>tcp_rexmit_interval_max</code> パラメータも変更すべきです。 <code>tcp_rexmit_interval_max</code> には、 <code>tcp_rexmit_interval_min</code> の 8 倍以上の値を指定します。
コミットレベル	変更の可能性あり

### `tcp_rexmit_interval_extra`

説明	計算された再送タイムアウト値 (RTO) に加える定数をミリ秒で指定します。
デフォルト	0 ミリ秒
範囲	0 から 2 時間
動的か	はい
どのような場合に変更するか	この値は変更しないでください。  計算された RTO が接続に対して適切でないような状況では、不要な再転送を避けるためにこの値を変更することができます。
コミットレベル	変更の可能性あり

### `tcp_tstamp_if_wscale`

説明	このパラメータに 1 が設定され、かつ、ある接続に対するウィンドウスケールオプションが有効になっていると、TCP は、その接続の <code>timestamp</code> オプションも有効にします。
----	---

デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に変更するか	この値は変更しないでください。一般に、TCP を高速ネットワークで使用する場合は、シーケンス番号のラップアラウンドに対するプロテクションが必要なため、 <code>timestamp</code> オプションの使用が必要になることがあります。
コミットレベル	変更の可能性あり

### `tcp_recv_hiwat_minmss`

説明	デフォルトの最小受信ウィンドウサイズを制御します。最小値は、 <code>tcp_recv_hiwat_minmss</code> に、接続の最大セグメントサイズ (MSS) を掛けたものです。
デフォルト	4
範囲	1 から 65,536
動的か	はい
どのような場合に変更するか	この値は変更しないでください。この値を変更する必要がある場合は、4 より小さい値にしないでください。
コミットレベル	変更の可能性あり

### `tcp_compression_enabled`

説明	1 が設定されていると、メモリーを節約するために、TIME-WAIT 状態にある TCP 接続のプロトコル制御ブロックが圧縮されます。0 が設定されていると、圧縮は行われません。163ページの「 <code>tcp_time_wait_interval</code> 」も参照してください。
----	---

デフォルト	1 (有効)
範囲	0 (無効)、1 (有効)
動的か	はい
どのような場合に変更するか	圧縮機能はオフにしないでください。
コミットレベル	変更の可能性あり

---

## UDP チューニング可能パラメータ

この節では、UDP チューニング可能パラメータの一部を説明します。

### udp\_xmit\_hiwat

説明	デフォルトの最大 UDP ソケットデータグラムサイズをバイト数で指定します。詳細は、174 ページの「udp_max_buf」を参照してください。
デフォルト	8192 バイト
範囲	4096 から 65,536
動的か	はい
どのような場合に変更するか	このサイズは、アプリケーションで <code>setsockopt(3SOCKET) SO_SNDBUF</code> を使用してソケットごとに変更できることに留意してください。一般には、デフォルト値を変更する必要はありません。
コミットレベル	変更の可能性あり

## udp\_recv\_hiwat

説明	デフォルトの最大 UDP ソケット受信バッファサイズをバイト数で指定します。詳細は、174 ページの「udp_max_buf」を参照してください。
デフォルト	8192 バイト
範囲	4096 から 65,536
動的か	はい
どのような場合に変更するか	このサイズは、アプリケーションで <code>setsockopt(3SOCKET) SO_RCVBUF</code> を使用してソケットごとに変更できることに留意してください。一般には、デフォルト値を変更する必要はありません。
コミットレベル	変更の可能性あり

## 特別な注意を要する UDP パラメータ

各パラメータで記述する特別な状況である場合以外は、次のパラメータを変更することは推奨されません。

## udp\_max\_buf

説明	UDP ソケットに対する送信バッファと受信バッファのサイズの最大値をバイト数で指定します。
デフォルト	262,144 バイト
範囲	65,536 から 1,073,741,824
動的か	はい
どのような場合に変更するか	この値は変更しないでください。このパラメータの値が大きすぎると、UDP ソケットアプリケーションがメモリーを過度に使用するおそれがあります。

## ルート別のメトリック

Solaris 8 リリースでは、ルート別のメトリック (測定基準) を使用して、一定のプロパティを IPv4 や IPv6 のルーティングテーブルエントリに関連付けることができます。

たとえば、システムに、fast ethernet と gigabit ethernet という 2 つの異なるネットワークインタフェースがあるとします。tcp\_recv\_hiwat のシステムデフォルト値は 24,576 バイトです。このデフォルト値は fast ethernet インタフェースには十分ですが、gigabit ethernet インタフェースには不十分な可能性があります。

tcp\_recv\_hiwat のシステムデフォルト値を増やす代わりに、gigabit ethernet インタフェースのルーティングエントリに別のデフォルトの TCP 受信ウィンドウサイズを関連付けることができます。これにより、そのルートを通るすべての TCP 接続には、大きくした受信ウィンドウサイズが使用されます。

IPv4 の場合、ルーティングテーブルは次のように表示されます (netstat -rn)。

192.123.123.0	192.123.123.4	U	1	4	hme0
192.123.124.0	192.123.124.4	U	1	4	ge0
default	192.123.123.1	UG	1	8	

次のコマンドを実行します。

```
# route change -net 192.123.124.0 -recvpipe x
```

このコマンドの結果、ge0 リンクの 192.123.124.0 ネットワークへのすべての接続には、デフォルトの受信ウィンドウサイズ 24567 の代わりに、受信バッファサイズ *x* が使用されます。

宛先が a.b.c.d ネットワークにあり、そのネットワーク固有のルーティングエントリがない場合は、そのネットワークに接頭辞ルートを追加し、メトリックを変更できます。

たとえば、次のようにします。

```
# route add -net a.b.c.d 192.123.123.1 -netmask w.x.y.z
# route change -net a.b.c.d -recvpipes y
```

接頭辞ルートのゲートウェイがデフォルトのルーターであることに留意してください。そのネットワークへのすべての接続は受信バッファサイズ *y* を使用します。複数のインタフェースがある場合は、`-ifp` 引数を使用して、使用するインタフェースを指定します。それによって、特定の宛先に対してどのインタフェースを使用するかを制御できます。メトリックを検査するには、`route(1M) get` コマンドを使用します。

## システム機能のパラメータ

---

この章では、さまざまなシステム機能のデフォルト値を設定するパラメータについて説明します。ここには、ほとんどのパラメータが含まれています。

- 178ページの「cron」
- 178ページの「devfsadm」
- 178ページの「dhcpagent」
- 178ページの「fs」
- 179ページの「inetinit」
- 179ページの「init」
- 179ページの「kbd」
- 179ページの「login」
- 179ページの「nfslogd」
- 180ページの「passwd」
- 180ページの「power」
- 180ページの「su」
- 180ページの「sys-suspend」
- 180ページの「tar」
- 181ページの「utmpd」

---

## システムのデフォルトのパラメータ

さまざまなシステム機能の動作は、その機能が起動時に読み込む一連の値によって制御されます。各機能について1つのファイルに保存された値は、`/etc/default` ディレクトリに置かれます。ただし、すべてのシステム機能がこのディレクトリにファイルを持っているわけではありません。

### cron

詳細は、`cron(1M)` のマニュアルページの「Setting cron Defaults」の節を参照してください。

### devfsadm

現在、このファイルは使用されていません。

### dhcpcagent

クライアントでは、`dhcpcagent` デーモンで提供される DHCP を使用します。`ifconfig` は、DHCP からネットワーク構成を受信するように構成されたインタフェースを識別する際に、このクライアントデーモンを起動してそのインタフェースを管理します。

詳細は、`dhcpcagent(1M)` のマニュアルページの「`/etc/default/dhcpcagent`」の節を参照してください。

### fs

ファイルシステム管理コマンドには、汎用的な部分とファイルシステム固有の部分があります。ファイルシステムのタイプが `-F` オプションで明示的に指定されていない場合は、デフォルトが使用されます。その値はこのファイルに指定されています。詳細は、`default_fs(4)` のマニュアルページを参照してください。

## inetinit

/etc/rc2.d/S69inet スクリプトが、TCP によって使用されるシーケンス番号を制御するために使用します。

## init

詳細は、init(1M) のマニュアルページの「/etc/default/init」の節を参照してください。

このファイルで参照される CMASK 変数の説明は、マニュアルページには記載されていません。CMASK は init が使用する umask で、すべてのプロセスが init プロセスから継承します。設定されていないと、init は、カーネルから取得したデフォルトの umask を使用します。init プロセスは、CMASK の設定に関係なく、常に、ファイルを作成する前に umask として 022 を適用しようとします。このファイルのすべての値は、シングルユーザーのブート要求に応じて init が呼び出すシェルの環境に置かれます。さらに、init プロセスは、自身が /etc/inittab ファイルから起動または再起動するすべてのコマンドにこれらの値を渡します。

## kbd

詳細は、kbd(1) のマニュアルページの「Extended Description」の節を参照してください。

## login

詳細は、login(1) のマニュアルページの「FILES」節の「/etc/default/login」を参照してください。

## nfslogd

詳細は、nfslogd(1M) のマニュアルページの「Description」の節を参照してください。

## passwd

詳細は、passwd(1) のマニュアルページの「FILES」の節の「/etc/default/passwd」を参照してください。

## power

詳細は、pmconfig(1M) のマニュアルページの「FILES」の節の「/etc/default/power」を参照してください。

## su

詳細は、su(1M) のマニュアルページの「FILES」の節の「/etc/default/su」を参照してください。

## sys-suspend

詳細は、sys-suspend(1M) のマニュアルページの「FILES」の節の「/etc/default/sys-suspend」を参照してください。

## tar

tar(1) のマニュアルページで -f 関数修飾子の説明を参照してください。

TAPE 環境変数がなく、いずれかの引数が数字で、かつ -f が指定されていない場合、archiveN 文字列と一致する数値が /etc/default/tar ファイルで検索されます。このファイルの archiveN 文字列の値は、出力デバイス、ブロック化因数、サイズとして使用されます。

たとえば、次のコマンドを実行すると、

```
% tar -c 2 /tmp/*
```

出力は、/etc/default/tar ファイルで archive2 と指定されているデバイスに書き込まれます。

## utmpd

utmpd デーモンは、`/var/adm/utmpx` (および、Solaris の以前のバージョンでは `/var/adm/utmp` も) を監視し、`pututxline(3C)` で `root` 以外のプロセスに挿入された `utmp` エントリがプロセスの終了時に消去されることを確認します。

`/etc/default/utmpd` の次の 2 つのエントリがサポートされています。

- `SCAN_PERIOD` - utmpd が、監視するプロセスがまだ生きているかどうかの `/proc` のチェックとチェックの合間に休眠する秒数。デフォルト値は 300 です。
- `MAX_FDS` - utmpd が監視しようとするプロセスの最大数。デフォルト値は 4096 です。通常、この値を変更する必要はありません。



## チューニング可能パラメータの変更履歴

---

この付録には、特定のパラメータの変更履歴を記載します。また、元の機能がすでに削除されているパラメータもリストされています。

- 183ページの「プロセスのサイズに関するチューニング可能パラメータ」
- 186ページの「ページング関連のチューニング可能パラメータ」
- 188ページの「一般的なカーネル変数」
- 189ページの「一般的な入出力」
- 190ページの「仮想端末」
- 191ページの「Sun4u 特有」
- 192ページの「機能が削除されたパラメータ」

---

### カーネルパラメータ

#### プロセスのサイズに関するチューニング可能パラメータ

##### maxusers (Solaris 7 リリース)

説明	maxusers パラメータから max_nprocs と maxuprc が計算されます。
データ型	符号付き整数

デフォルト	M バイト単位のメモリー量と 1024 の小さい方
範囲	1 から 2048
	注 - /etc/system には、1024 より大きい値を指定する必要があります。2048 より大きい値が提供されると、計算では 2048 の値を使用しますが、その後の処理では、指定された値に設定されます。
単位	ユーザー
動的か	いいえ。従属変数の計算が終わった後に maxusers が再び参照されることはありません。
検査	なし
どのような場合に変更するか	システムが計算したデフォルトのユーザープロセス数では足りない場合。プロセス数が足りないと、次のメッセージがシステムコンソールやメッセージファイルに出力されます。
	out of processes
コミットレベル	変更の可能性あり

### max\_nprocs (Solaris 8 より前のリリース)

説明	<p>システム上に作成できるプロセスの最大数。これには、システムプロセスとユーザープロセスが含まれます。Solaris 8 より前のリリースでは、この値は計算によって算出され、maxuprc の設定に使用されていました。</p> <p>さらに、この値は、他のシステムデータ構造体のサイズを決める際にも使用されます。Solaris 8 より前のリリースでは、/etc/system に値が指定されていると、計算された値の代わりにこの値が使用されます。この変数が使用される他のデータ構造体は次のとおりです。</p>
----	---

- ディレクトリ名ルックアップキャッシュのサイズを決めるとき (ncsize が指定されていない場合)
- UFS のディスク割り当て構造体を割り当てるとき (ndquot が指定されていない場合)
- 構成された system V セマフォによって使用されるメモリの総量がシステム限度をこえていないか確認するとき
- sun4d、sun4m、Intel プラットフォーム向けのハードウェアアドレス変換のリソースを構成するとき

データ型	符号付き整数
デフォルト	10 + (16 × maxusers)
範囲	266 から pidmax の値
動的か	いいえ。初期のパラメータ計算の後に、max_nprocs は v 構造体の v_proc 要素に代入されます。動作しているシステムの v.v_proc を変更すると、ほぼ確実に、システムがクラッシュするか、エラー表示のないままデータが損傷を受けます。
検査	maxpid と比較し、それより大きい場合は、maxpid に設定されます。sun4d と Intel プラットフォームでは、さらにプラットフォーム固有の値と比較されます。max_nprocs は、max_nprocs、maxpid、プラットフォーム値のうち最も小さい値に設定されます。両プラットフォームとも、プラットフォーム値として 65,534 を使用します。
どのような場合に変更するか	Solaris 8 リリースから、この値を変更することで 1 つのシステムで 30,000 を超えるプロセスをサポートできるようになりました。このパラメータの変更は、一つのシステムで 30,000 を

越えるプロセスを可能にするために必要となる手順の一部です。

コミットレベル

変更の可能性あり

## ページング関連のチューニング可能パラメータ

Solaris 2.6 カーネルパッチの一部のリビジョン (SPARC プラットフォームの 105181-10 および Intel プラットフォームの 105182-09) と Solaris 7 リリースで、新しいパラメータ `priority_paging` (優先ページング) が導入されました。ページアウトスレッド活動の新しい開始点 (`cachefree`) も使用されます。使用可能なメモリーが `cachefree` と `lotsfree` の間にある間は、実行可能部分 (テキスト、スタック、またはデータ) からのページである場合は、優先ページングはそのページをスキップするようページ検査のアルゴリズムを変更します。メモリーが `lotsfree` を下回ると、すべてのページが同等にみなされます。この機能はデフォルトで無効になっています。この機能を有効にするには、`cachefree` に `lotsfree` より大きい値を設定するか、`priority_paging` 変数にゼロ以外の値を設定します (`cachefree` に `lotsfree` の 2 倍の値が設定されます)。

### `priority_paging` (Solaris 2.6 リリースおよび Solaris 7 リリース)

説明	優先ページング機能を有効にします。設定すると、 <code>cachefree</code> に <code>lotsfree</code> の 2 倍の値が設定され、優先ページングが有効になります。
データ型	符号付き整数
デフォルト	0
範囲	0 ( <code>cachefree</code> が別途設定されていない場合、優先ページングを無効にする) か 1 (有効にする)
単位	切り替え (オン/オフ)
動的か	いいえ。 <code>cachefree</code> の値はブート時にのみ設定します。実行時に有効にするには、システムの動作中に <code>adb</code> で <code>cachefree</code> を設定します。
検査	なし

どのような場合に変更するか システムのメモリーが逼迫している場合、およびファイルの内容が将来必要であるような入出力を極めて多く行う場合を除き、常に有効にしておくべきです。

コミットレベル 廃止または互換性がなくなる可能性あり

## cachefree (Solaris 2.6 リリースおよび Solaris 7 リリース)

説明 cachefree に lotsfree より大きい値を設定すると、優先ページング機能が有効になります。この変数は、少なくともパッチ 105181 のリビジョン 10 がインストールされた Solaris 2.6 リリース、および Solaris 7 リリースが動作するシステムで使用できます。この機能は、デフォルトでは無効になっています (cachefree と lotsfree が等しい)。

データ型 符号なし long

デフォルト priority\_paging が設定されていない場合は、lotsfree の値です。設定されている場合は、cachefree は lotsfree の 2 倍です。

範囲 lotsfree からシステムの物理メモリー量

単位 ページ

動的か はい

検査 lotsfree より小さいと、lotsfree の値にリセットされます。

どのような場合に変更するか システムのメモリーが逼迫している場合、およびファイルの内容が将来必要であるような入出力を極めて多く行う場合を除き、常に有効にしておくべきです。

コミットレベル 廃止または互換性がなくなる可能性あり

## 一般的なカーネル変数

### noexec\_user\_stack (Solaris 2.6 リリースおよび Solaris 7 リリース)

説明	<p>Solaris 2.6 リリースで導入されたもので、スタックを実行不能と印付けすることができます。これにより、バッファオーバーフロー攻撃が前よりも困難になります。</p> <p>Solaris 2.6 リリースでは、この値はスレッド化されたアプリケーションには適用されません。すべての 64 ビットの Solaris アプリケーションでは、この変数の設定に関係なく、すべてのスタックが実行不能にされます。</p> <hr/> <p>注 - この変数は Solaris 2.6 以降が動作しているすべてのシステムに存在しますが、sun4u、sun4m、sun4d アーキテクチャでのみ有効です。</p> <hr/>
データ型	符号付き整数
デフォルト	0 (無効)
範囲	0 (無効)、1 (有効)
単位	切り替え (オン/オフ)
動的か	はい。ただし、すでに実行中のプロセスに対しては無効です。値が設定された後に起動されたプロセスに対してのみ有効です。
検査	なし
どのような場合に変更するか	アプリケーションが、mprotect(2) を使用してスタックを実行可能にすることなくスタックに実行可能コードを意図的に置いている場合を除き、この変数は常に有効にすべきです。
コミットレベル	変更の可能性あり

## 一般的な入出力

### `rlim_fd_cur` (Solaris 7 リリースおよびそれより前のリリース)

説明	1つのプロセスがオープンできるファイル記述子の「弱い」限度。プロセスは、自身のファイル記述子限度数を、 <code>rlim_fd_max</code> で定義される「強い」限度の範囲内で調整できます。それには、 <code>setrlimit()</code> 呼び出しを使用するか、プロセスを実行するシェルで <code>limit</code> コマンドを実行します。「強い」限度の範囲内で値を調整する場合には、スーパーユーザー特権は必要ありません。
データ型	符号付き整数
デフォルト	64
範囲	1 から MAXINT
単位	ファイル記述子
動的か	いいえ。 <code>rlimits</code> 構造体にロードされます。
検査	<code>rlim_fd_max</code> と比較され、 <code>rlim_fd_cur</code> の方が大きい場合は、 <code>rlim_fd_cur</code> は <code>rlim_fd_max</code> にリセットされます。
どのような場合に変更するか	1 プロセス当たりのデフォルトのオープンファイル数が十分でない場合。この値を増やす意味は、プログラムで <code>setrlimit(2)</code> を使用して自身で利用できる最大ファイル記述子数を増やす必要がなくなるという点にあります。
コミットレベル	変更の可能性あり

### `segkpsize` (Solaris 7 リリースおよびそれより前のリリース)

説明	利用できるページング可能なカーネルのメモリー量を指定します。このメモリーは主にカー
----	---

ネルスレッドのスタックに使用されます。この数を増やすと、スレッドの数を増やさないのであれば個々のスレッドでより大きなスタックが使用でき、あるいはより多くのスレッドを使用できるようになります。このパラメータは 64 ビットカーネルでのみ設定できます。64 ビットカーネルは、デフォルトでスタックサイズに 24K バイトを使用します。

このパラメータは、パッチ 106541-04 が適用された Solaris 7 リリースか、Solaris 7 5/99 リリース、および Solaris 8 リリースで使用できます。

データ型	符号なし long
デフォルト	64 ビットカーネルでは 2G バイト、32 ビットカーネルでは 512M バイト
範囲	64 ビットカーネルでは 512M バイトから 24G バイト、32 ビットカーネルでは 512M バイト
単位	M バイト
動的か	いいえ
検査	なし
どのような場合に変更するか	スレッドを増やしたい場合
コミットレベル	変更の可能性あり

## 仮想端末

### pt\_cnt (Solaris 7 リリースおよびそれより前のリリース)

説明	再構成ブートで作成する /dev/pts (telnet や rlogin でネットワークログインに使用される仮想端末デバイス) エントリの数。このパラメータにより、ネットワークを介して同時にログインできるユーザー数が、pt_cnt の値に制限されま
----	---

す。追加のデバイスノードを作成するためには、`/etc/system` ファイルを変更した後で再構成ブート (`boot -r`) を行う必要があります。

データ型	符号付き整数
デフォルト	48
範囲	0 から <code>maxpid</code>
単位	ログイン / ウィンドウ
動的か	いいえ
検査	なし。値が過度に大きすぎると、システムがハングします。
どのような場合に変更するか	必要な数のユーザーがシステムにログインできない場合
コミットレベル	変更の可能性あり

## Sun4u 特有

### `enable_grp_ism` (Solaris 2.6 リリース)

説明	<code>SHARE_MMU</code> フラグを設定して接続された System V Shared Memory の共有メモリー Translation Setaside Buffer (TSB) 機能を有効にします。このパラメータは、パッチ 105181-05 以降がインストールされた Solaris 2.6 リリースで利用できます。Solaris 7 リリースからはこのパラメータ名は削除されましたが、システムはデフォルトでこの機能を実装します。
データ型	符号付き整数
デフォルト	0
範囲	0 (無効)、1 (有効)

動的か	いいえ
検査	なし
どのような場合に変更するか	SHARE_MMU フラグを設定して接続された System V Shared Memory を使用する場合に有効にする
コミットレベル	変更の可能性あり

---

## 機能が削除されたパラメータ

この節では、機能はすでに削除されているが、互換性の理由でまだ残されているパラメータについて記述します。これらのパラメータを設定しても無視されます。

### ページング関連のチューニング可能パラメータ

`tune_t_gpgslo`

説明 廃止。変数は互換性の理由で残されています。

`tune_t_minasmem`

説明 廃止。変数は互換性の理由で残されています。

## System V メッセージパラメータ

`msgsys:msginfo_msgssz`

説明 システムがメッセージバッファ用の空間の制御に使用する断片のサイズを指定します。Solaris 8 リリースから廃止されています。

データ型 符号付き整数

デフォルト 40

範囲	0 から MAXINT
動的か	いいえ。msginfo 構造体の msgtql フィールドにロードされます。
検査	メッセージやキューのために作成されるデータ構造体が最大数存在する場合に消費される空間と、モジュールがロードされるときの使用可能なカーネルメモリーの 25% を比較します。使用される空間があまりに大きいと、メッセージキューモジュールはロードを拒否し、この機能は利用できません。この計算には、メッセージに消費される空間も含まれます。この状態は、モジュールが最初にロードされるときだけ起こります。
どのような場合に変更するか	デフォルト値では十分でない場合。一般には、ソフトウェアベンダーからの推奨がある場合に変更します。
コミットレベル	廃止または互換性がなくなる可能性あり

#### msgsys:msginfo\_msgmap

説明	システムがサポートするメッセージの数。 Solaris 8 リリースから廃止されています。
データ型	符号付き整数
デフォルト	100
範囲	0 から MAXINT
動的か	いいえ
検査	メッセージやキューのために作成されるデータ構造体が最大数存在する場合に消費される空間と、モジュールがロードされるときの使用可能なカーネルメモリーの 25% を比較します。使用される空間があまりに大きいと、メッセージキューモジュールはロードを拒否し、この機能は利用できません。この計算には、メッセージに消費さ

れる空間も含まれます。この状態は、モジュールが最初にロードされる時だけ起こります。

どのような場合に変更するか デフォルト値では十分でない場合。一般には、ソフトウェアベンダーから推奨がある場合に変更します。

コミットレベル 廃止または互換性がなくなる可能性あり

## msgsys:msginfo\_msgseg

説明 使用可能なメッセージメモリーのプールとしてシステムが使用する msginfo\_msgssz セグメントの数。メッセージに使用できる合計のメモリーは msginfo\_msgseg \* msginfo\_msgssz です。Solaris 8 リリースで廃止されています。

データ型 符号付き short

デフォルト 1024

範囲 0 から 32,767

動的か いいえ

検査 メッセージやキューのために作成されるデータ構造体が最大数存在する場合に消費される空間と、モジュールがロードされる時の使用可能なカーネルメモリーの 25% を比較します。使用される空間があまりに大きいと、メッセージキューモジュールはロードを拒否し、この機能は利用できません。この計算には、メッセージに消費される空間も含まれます。この状態は、モジュールが最初にロードされる時だけ起こります。

どのような場合に変更するか デフォルト値では十分でない場合。一般には、ソフトウェアベンダーからの推奨がある場合に変更します。

コミットレベル 廃止または互換性がなくなる可能性あり

## System V セマフォのパラメータ

`semsys:seminfo_semmap`

廃止。変数は互換性の理由でカーネルに残されていますが、使用されません。

`semsys:seminfo_semusz`

廃止。値を入力しても無視されます。

## NFS モジュールのパラメータ

`nfs:nfs_32_time_ok`

Solaris 8 リリースで廃止されています。

`nfs:nfs_acl_cache`

Solaris 2.6 リリースで廃止されています。



## このマニュアルの改訂履歴

---

この付録では、このマニュアルの改訂履歴について説明しています。

---

### 現行バージョン - Solaris 8 1/01 リリース

このマニュアルの現行バージョンは、Solaris 8 1/01 リリースに適用されます。

---

### 前のリリースからの変更

#### shmsys:shminfo\_shmmin (Solaris 8 1/01 リリース)

次の箇所が変更されています。

どのような場合に変更するか	現在、変更する理由は見当たりません。 を、次の内容へ変更：
どのような場合に変更するか	変更は推奨されません。この値が大きすぎると、powerd などのシステムプログラムが異常を起こすことがあります。プログラムで shminfo_shmmin の値より小さいセクションの作成を試みると、そのセグメントを作成しようとしている時に EINVAL エラーが返され、通常、終了します。

詳細は、101ページの「shmsys:shminfo\_shmmin」を参照してください。

## `semsys:seminfo_semmnu` (**Solaris 8 1/01** リリース)

このパラメータが追加されました。

詳細は、97ページの「semsys:seminfo\_semmnu」を参照してください。

# 索引

---

## A

autoup 34

## B

bufhwm 76

## C

cachefree 51, 187  
consistent\_coloring 106  
CRON 178

## D

desfree 45  
dhcpageant 178  
dnlc\_dir\_enable 74  
dnlc\_dir\_max\_size 75  
dnlc\_dir\_min\_size 74  
doiflush 36  
dopageflush 35

## E

enable\_grp\_ism 191

## F

fastscan 54  
fs 178  
fsflush 32

## H

handspreadpages 56  
hires\_tick 105

## I

inetinit 179  
init 179  
ip6\_forwarding 151  
ip6\_forward\_src\_routed 153  
ip6\_icmp\_return\_data\_bytes 156  
ip6\_respond\_to\_echo\_multicast 152  
ip6\_send\_redirects 153  
ip6\_strict\_dst\_multihoming 154  
ipc\_tcp\_conn\_hash\_size 168  
ip\_addr\_per\_if 154  
ip\_forwarding 151  
ip\_forward\_src\_routed 153  
ip\_icmp\_err\_burst 151  
ip\_icmp\_err\_interval 151  
ip\_icmp\_return\_data\_bytes 156  
ip\_ire\_pathmtu\_interval 155  
ip\_respond\_to\_echo\_broadcast 152  
ip\_send\_redirects 153  
ip\_strict\_dst\_multihoming 154

## K

kbd 179  
kmem\_flags 62



segspt\_minfree 103  
semsys:seminfo\_semaem 99  
semsys:seminfo\_semmap 195  
semsys:seminfo\_semmni 94  
semsys:seminfo\_semmns 94  
semsys:seminfo\_semmnu 97  
semsys:seminfo\_semmsl 96  
semsys:seminfo\_semopm 97  
semsys:seminfo\_semume 98  
semsys:seminfo\_semusz 195  
semsys:seminfo\_semvmx 95  
shmsys:shminfo\_shmmax 100  
shmsys:shminfo\_shmmin 101  
shmsys:shminfo\_shmmni 102  
shmsys:shminfo\_shmseg 102  
slowscan 55  
strmsgsz 89  
su 180  
swapfs\_minfree 60  
swapfs\_reserve 59  
sys-suspend 180

## T

tar 180  
tcp\_compression\_enabled 173  
tcp\_conn\_hash\_size 167  
tcp\_conn\_req\_max\_q 164  
tcp\_conn\_req\_max\_q0 165  
tcp\_conn\_req\_min 166  
tcp\_cwnd\_max 160  
tcp\_deferred\_acks\_max 157  
tcp\_deferred\_ack\_interval 156  
tcp\_ip\_abort\_interval 169  
tcp\_max\_buf 160  
tcp\_rcv\_hiwat 159  
tcp\_rcv\_hiwat\_minmss 172

tcp\_rev\_src\_routes 163  
tcp\_rexmit\_interval\_extra 171  
tcp\_rexmit\_interval\_initial 169  
tcp\_rexmit\_interval\_max 170  
tcp\_rexmit\_interval\_min 170  
tcp\_sack\_permitted 162  
tcp\_slow\_start\_after\_idle 162  
tcp\_slow\_start\_initial 161  
tcp\_time\_wait\_interval 163  
tcp\_tstamp\_always 158  
tcp\_tstamp\_if\_wscales 172  
tcp\_wscales\_always 158  
tcp\_xmit\_hiwat 159  
throttlefree 48  
timer\_max 105  
tmpfs\_maxkmem 83  
tmpfs\_minfree 84  
tune\_t\_fsflushr 33  
tune\_t\_gpgslo 192  
tune\_t\_minarmem 53  
tune\_t\_minasmem 192

## U

udp\_max\_buf 174  
udp\_rcv\_hiwat 174  
udp\_xmit\_hiwat 173  
ufs:ufs\_HW 82  
ufs:ufs\_LW 82  
ufs:ufs\_WRITES 81  
ufs\_ninode 79  
utmpd 181

## X

xxx:ip\_forwarding 152