



Hewlett Packard
Enterprise

HP-UX 11i 構成可能 カーネル・パラメータ (HP-UX 11.0 用)

《 2000 年時点 》

カーネル動作パラメータには、特定のシステム・ニーズに合わせて構成できるものがあります。これにより、性能を向上したり、リソースをより効果的に割り当てることができます。各パラメータの理想的な値は、多くの場合システムの特定のハードウェア構成、システムで実行されるアプリケーションの組み合わせ、システム・ユーザーの信用度など、システム毎に幅広く変化する要因により決まります。

これらのパラメータには、あらかじめ合理的なデフォルト値が設定されていますが、実際に使用するユーザーのニーズに応じてより適するように変更することもできます。それぞれの構成可能カーネル・パラメータの詳細は、下記の一覧を参照してください。

なお、個々のパラメータは通常、特定のサブシステムに関連付けられています。パラメータは、独立して機能するものもありますが、相互関係を持っていたり、相互作用を行うものもあります。下記のオンライン・ヘルプ・トピックの一覧は、サブシステム別にグループ化しています。

警告

カーネル・パラメータに不適切な値を設定したり、複数のパラメータの値を互いに不適切な組み合わせで設定すると、データの損失、システム・パニック、その他、動作上の問題が発生する可能性があります。パラメータの不適切な設定によって引き起こされた問題は、後から原因を究明するのが困難になりがちなので、十分に注意してください。

- 構成可能カーネル・パラメータの値を変更するときには、まず最初に、その変更がどのような影響を与える可能性があるかを十分に検討してください。
- 各システム・パラメータは、必ず許容範囲内の値に設定してください (SAM では、許容範囲外の値は拒否されます)。
- 多くのパラメータは互いに影響を及ぼし合います。このようなパラメータの値は、相互のバランスを保って選択する必要があります。

記載内容に食い違いがある場合

技術の進歩によりシステムのサイズと能力はますます拡張されています。リリース間や、32 ビット・プロセッサと 64 ビット・プロセッサ間で、特定のパラメータの最大値やデフォルト値が変更されることも少なくありません。この Web ページに関連付けられているドキュメントには、作成の時点において正確と考えられる情報が記載されています。しかし、システムの変更により、ドキュメントの記載内容と実際のシステム仕様間に食い違いが生じる可能性があります。

デフォルト値や制限値に関して食い違いがあることが判明した場合は、システム上の `/etc/conf/master.d` に格納されているファイルを参照し、マシンに対応する実際の値を確認してください。

— 目次 —

アカウントティング・サブシステム

アカウントティングの中断と再開の概要

アカウントティングに関する構成可能パラメータ

acctsuspend

アカウントティングを再開します。

acctresume

アカウントティングを再開します。

チュートリアル: アカウンティングのしきい値の指定**非同期 I/O サブシステム**非同期 I/O に関するパラメータの概要aio_listio_max

lio_list()コールに指定する AIO 処理の最大許容数。

aio_max_ops

同時に待ち行列に格納する AIO 処理の最大許容数。

aio_phymem_pct

lio_list()コールに指定する AIO 処理の最大許容数。

aio_prio_delta_max

スローダウン・要素の最大許容値。aiocb の aio_reqprio フィールドに指定する優先順位の最大下げ幅。

ダンプに関するパラメータカーネル・パニック・ダンプに関するパラメータの概要alwaydump

ダンプに含めるカーネル・メモリ・ページのビット・マスク。

dontdump

ダンプから除外するカーネル・メモリ・ページのビット・マスク

initmodmax

システム・クラッシュ・ダンプによって保存されるカーネル・モジュールの最大許容数。

modstrmax

カーネル・モジュール savecrash テーブルの最大許容サイズ。

FibreChannel サブシステムFibreChannel に関するカーネル・パラメータの概要num_tachyon_adapters

システム内にインストールされている Tachyon(タキオン)アダプタの数。

max_fcp_reqs

Tachyon FCP アダプタ上で同時に処理する FCP 要求の最大許容数。

大容量記憶装置サブシステムファイル・システムに関するカーネル・パラメータの概要ファイル・システムに関するカーネル・パラメータの一覧

- **ファイル・システム・バッファ・キャッシュに関する構成可能パラメータ**

bufpages

静的バッファ・キャッシュのページ数。

dbc_min_pct

動的バッファ・キャッシュの最小サイズ。

dbc_max_pct

動的バッファ・キャッシュの最大サイズ。

nbuf

静的バッファ・ヘッダの数。

- **ディスクの先読みに関する構成可能パラメータ**

hfs_ra_per_disk

HFS ファイル・システムにおけるディスク・ドライブあたりの先読みデータ量を、KB 単位で設定します。

hfs_max_ra_blocks

単一の HFS ファイル・システムについて、カーネルが未処理のまま先読みできる最大ブロック数を設定します。

hfs_revra_per_disk

HFS ファイル・システムにおけるディスク・ドライブあたりの逆方向の先読みデータ量を、KB 単位で設定します。

hfs_max_revra_blocks

単一の HFS ファイル・システムについて、カーネルが未処理のまま逆方向に先読みできる最大ブロック数を設定します。

vxfs_ra_per_disk

VxFS ファイル・システムにおけるディスク・ドライブあたりの先読みデータ量を、KB 単位で設定します。

vxfs_max_ra_kbytes

単一の VxFS ファイル・システムについて、カーネルが未処理のまま先読みできる最大データ量を、KB 単位で設定します。

- **SCSI デバイスに関する構成可能パラメータ**

scsi_maxphys

SCSI I/O サブシステムにおける最大レコード・サイズを、バイト単位で設定します。

scsi_max_qdepth

SCSI デバイスについて、待ち行列に入れることのできる SCSI コマンドの最大数を設定します。

st_fail_overruns

データ・オーバーランが発生した SCSI テープ読み込みはエラーとなります。

st_large_rec

SCSI テープにおける大容量レコードのサポートを可能にします。

- **ファイルのオープンおよびロックに関する構成可能パラメータ**

maxfiles

オープンするファイルの数の論理制限値。

maxfiles_lim

オープンするファイルの数の物理制限値。

nfile

システム全体においてオープンするファイルの数の制限値。

nflocks

システム全体においてロックするファイルの数の制限値。

ninode

メモリ内で存在可能な、HFS ファイルシステムのオープン i ノードの最大数。

- **非同期書き込みに関する構成可能パラメータ**

fs_async

ディスクへの非同期書き込みを可能/不可能にします。

- **VxFS (ジャーナル) ファイル・システムに関する構成可能パラメータ**

vx_ncsize

VxFS のディレクトリ・パス名キャッシュ用に予約するメモリ・スペース。

論理ボリューム・マネージャ(LVM)

LVM の動作の概要

Logical Volume Manager に関する構成可能パラメータ

maxvgs

システム上のボリューム・グループの最大許容数。

no_lvm_disks

システム上にボリューム・グループが存在しないことを示します (ワークステーションのみ)。

メモリ・スワップ・サブシステム

メモリ・ページングに関するパラメータの概要

メモリ・ページングに関する構成可能パラメータ

allocate_fs_swapmap

スワップ・データ構造の固定割り当てまたは動的割り当て。

maxswapchunks

システム上で構成可能なスワップ・スペースの最大サイズ。

nswapdev

利用可能なスワップ・デバイスの数。

nswapfs

スワップに利用できるファイル・システムの数。

swapmem_on

擬似スワップ予約を使用可能/使用不可能にします。

swchunk

クライアント・スワップ・チャンクのサイズ。

- **可変ページ・サイズに関するパラメータ**

vps_ceiling

システムによって選択されるページの最大許容サイズ (KB 単位)。

vps_chatr_ceiling

chatr によって選択されるページの最大許容サイズ (KB 単位)。

vps_pagesize

ユーザー・ページのデフォルト・サイズ (KB 単位)。

プロセス管理サブシステム

プロセス管理に関するパラメータの概要

プロセス管理に関する構成可能パラメータ

maxdsiz

プロセス・データ・セグメントの最大許容サイズ。

maxssiz

プロセス記憶セグメントの最大許容サイズ。

max_thread_proc

1 つのプロセスが作成するスレッドの最大許容数。

maxtsiz

プロセス・テキスト・セグメントの最大許容サイズ。

maxuprc

1 ユーザーあたりのプロセスの最大許容数。

nkthread

システム上に同時に存在するカーネル・スレッドの最大許容数。

nproc

システム全体のプロセスの最大許容数。

timeslice

競合プロセス間でのタイム・スライスの割り当て。

キャラクタ・モード I/O ストリームに関するパラメータ

キャラクタ・モード I/O ストリームに関するパラメータの概要

ストリームに関する構成可能カーネル・パラメータ

NSTRBLKSCHED

出荷前の設定専用。値を変更しないでください。

NSTREVENT

システム上に同時に存在する未処理のストリーム bufcall の最大許容数。

NSTRPUSH

システム上の 1 つのストリーム内に同時に存在するストリーム・モジュールの最大許容数。

NSTRSCHED

システム上で同時に実行されるストリーム・スケジューラ・デーモンの最大許容数。

STRCTL SZ

システム上の任意のストリーム・メッセージの制御部分に格納される制御バイトの最大許容数。

STRMSG SZ

システム上の任意のストリーム・メッセージのデータ部分に格納されるバイトの最大許容数。

nstrpty

システム上に同時に存在するストリーム・ベースの PTY の最大許容数。これは、システム全体に適用される値です。

streampipes

すべてのパイプを強制的にストリーム・ベースにします。

System V IPC 共有メモリ・サブシステム

System V のプロセス間通信機構メッセージ待ち行列の処理の概要メッセージ用メモリ・スペースの割り当てIPC メッセージに関する構成可能パラメータmesg

IPC メッセージを使用可能/使用不可能にします (ワークステーションのみ)。

msgmap

メッセージの空きスペース・マップのサイズ。

msgmax

メッセージの最大サイズ。

msgmnb

メッセージ待ち行列に格納されるバイト数の最大値。

msgmni

システム上に存在するメッセージ・キューの最大許容数。

msgseq

メッセージ待ち行列内のセグメント数。

msgssz

メッセージ・セグメントのサイズ。

msgtql

システム上に存在するメッセージの最大許容数。

セマフォに関するパラメータの概要IPC セマフォに関する構成可能パラメータsema

セマフォを使用可能/使用不可能にします (ワークステーションのみ)。

semaem

セマフォの値を変更するときの差分の最大許容値。

semmap

セマフォ・リソース・マップのサイズ。

semmni

システム全体の IPC セマフォの最大許容数。

semmns

システム全体において、システム・ユーザーが利用できるセマフォの総数。

semmnu

1 つのセマフォに対するアンドウの最大許容数。

semume

1 プロセスあたりのセマフォ・アンドウの最大許容数。

semvmx

セマフォの最大許容値。

共有メモリの処理の概要

IPC 共有メモリに関する構成可能パラメータ

shmem

共有メモリを使用可能/使用不可能にします (ワークステーションのみ)。

shmmax

共有メモリ・セグメントの最大許容サイズ。

shmmni

システム上のセグメントの最大許容数。

shmseg

1 プロセスあたりのセグメントの最大許容数。

VME I/O サブシステムに関するパラメータ

VME I/O サブシステムに関するパラメータ(オプション製品)

vmebpn_public_pages

VME に必要なカーネル I/O スペース・ページの数

vmebpn_sockets

ソケット・ドメイン AF_VME_LINK をアクティブにするかどうかを指定します (ブール値をとります)。

vmebpn_tcp_ip

DLPI PPA の最大許容数。

vmebpn_tcp_ip_mtu

PPA 転送単位の最大許容サイズ (KB 単位)。

vmebpn_total_jobs

同時にオープンする VME ポートの最大許容数。

vme_io_estimate

VME に必要な 4 KB のカーネル I/O スペース・ページの数。

その他の各種パラメータ

その他の各種パラメータの概要

その他の各種構成可能パラメータ

clireservedmem

クラスタの相互接続用に予約するシステム・メモリ (バイト単位)。

create_fastlinks

高速シンボリック・リンクを作成します。

default_disk_ir

ディスク I/O 時にすぐに制御を戻すかどうかを指定します。

dst

夏時間を使用可能/使用不可能にします。

eqmemsize

等価マップ・メモリ・プールのサイズ。

executable_stack

スタック領域でのプログラムの実行を許可/禁止します。

ksi_alloc_max

システム全体において割り当てられる待ち行列シグナルの最大許容数。

ksi_send_max

プロセスが1つまたは複数の受信プロセスに送信して保留にできる待ち行列シグナルの最大数。

max_async_ports

システム全体において、プロセスが非同期ディスク I/O ドライバにアクセスするためにオープンするポートの最大許容数。

maxusers

システムに同時にログインするユーザーの最大想定数。

nccallout

タイムアウトの最大許容数。

ncdnode

オープンする CD-ROM FS ノードの最大許容数。

nclist

tty/pty I/O に使用できる cblock の最大数。

ndilbuffers

デバイス I/O ライブラリ用に同時にオープンされるファイルの最大許容数。

npty

システム全体に存在する pty の最大許容数。

nstrtel

telnet セッション・デバイス・ファイルの数。

nsysmap、nsysmap64

カーネル動的メモリ仮想アドレス空間のリソース・マップのエントリ数。

o_sync_is_o_dsync

open()およびfcntl()システム・コールにおいて O_SYNC を O_DSYNC に変換するかどうかを指定します。

pfail_enabled

電源異常の復旧を可能にします。

public_shlibs

共有ライブラリ上でパブリック保護 ID を使用できるようにするかどうかを指定します。

rtsched_numpri

システム上で利用できるリアルタイム割り込みの優先順位の最大数。

scroll_lines

ITE スクロール・バッファのサイズ。

sendfile_max

Web サーバー用の特殊パラメータ。

スピンロック・パラメータ (技術情報は非公開)。

timezone

世界協定時刻に対するタイムゾーン・オフセット。

unlockable_mem

システム用に予約するメモリのサイズ。

技術情報が公開されていないパラメータ

構成可能カーネル・パラメータには、すでに使用されなくなったもの、独立したソフトウェア・システムに関連付けられているもの、十分に理解せずに変更すると重大なシステム障害を引き起こす可能性があるもの、SAM ではサポートされていないものなども含まれています。これらのパラメータは、技術情報が非公開になっています。

[ドキュメントに記載されていないカーネル・パラメータ](#)

アカウントティングの中断と再開の概要

プロセス・アカウントティング・サブシステムは、システム管理者によって起動され、システム・プロセスとユーザー・プロセスの情報を格納するためのログ・ファイルの保守を行います。状況によってはこのログ・ファイルは容量が増大するため、ログ・ファイルが格納されているファイル・システム上の、他のファイル用の使用可能なスペースを圧迫する場合があります。

プロセス・アカウントティングが実行中の場合で、アカウントティング・ファイルが格納されているファイル・システムの使用可能なスペースがしきい値を下回った場合、システムはプロセス・アカウントティングを中断します。このしきい値は、システム・ブート時のカーネル・パラメータ `acctsuspend` の値と、ファイル・システム作成時に定義されているファイル・システム・パラメータ `minfree` の値の符号付き合計値として定義します。プロセス・アカウントティングが中断された場合は、以下のメッセージが出力されます。

```
Accounting suspended
```

このとき、アカウントティングを再開するのに十分なスペースが使用可能になるまで、アカウントティングは中断されます。再開するためのしきい値は、カーネル・パラメータ `acctresume` の値と、ファイル・システム・パラメータ `minfree` の値の符号付き合計値として定義します。アカウントティングの中断と再開との矛盾を防止するため、`acctresume` の値は必ず、`acctsuspend` の値より大きく設定することが必要です。

十分なファイル・システム・スペースが使用可能になり、アカウントティングが再開されると、以下のメッセージが出力されます。

```
Accounting resumed
```

値の範囲: アカウントティング中断のしきい値

アカウントティングを中断するしきい値は、0~100%までの、使用可能なファイル・システムの容量を示す任意のパーセンテージに設定できます。ただし、`acctsuspend` の値と `minfree` の合計が0または負の値となる場合は、中断しきい値が0%に設定されるため、アカウントティングを中断せずに、予約されている最小空きスペース全体をアカウントティング・ファイルで上書きできません。

`acctsuspend` と `acctresume` の値の指定方法については、[しきい値の指定](#) を参照してください。

値の範囲: アカウントティング再開のしきい値

アカウントティングを再開するしきい値についても、0~100%までの、使用可能なファイル・システムの容量を示す任意のパーセンテージに設定できます。システム・ブート時の `acctresume` と `minfree` との符号付き合計値が 100%またはそれ以上になった場合、使用可能なファイル・システムが 100%使用可能になるまで、アカウントティングは再開できません。これは、ファイル・システムが完全に空になるか(このような状況が発生する可能性は低いですが)、システムを再ブートするか、またはキーボードから `startup` コマンドを実行するまで、アカウントティングは中断されたままであることを意味します(HP-UX Reference の `acctsh(1M)` を参照)。より実用的な観点で言えば、ファイル・システム上の通常の最大空きスペースより再開しきい値を大きく設定すると、プロセス・アカウントティングが再開されなくなります。

付加情報

[アカウントティングに関する構成可能パラメータの一覧](#)

[しきい値の指定](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

アカウントティングに関する構成可能パラメータ

アカウントティングに関して構成可能なカーネル・パラメータには以下の 2 種類があります。

acctsuspend

アカウントティング・ログ・ファイルが格納されているファイル・システム上の空きスペースが、`acctsuspend` とファイル・システム全体の使用可能サイズに対する `minfree` のパーセンテージの合計値を下回った時点でアカウントティングを中断します。

acctresume

アカウントティング・ログ・ファイルが格納されているファイル・システム上の空きスペースが、`acctresume` とファイル・システム全体の使用可能サイズに対する `minfree` のパーセンテージの合計値を上回った時点でアカウントティングを再開します。

付加情報

[アカウントティングに関する構成可能パラメータの一覧](#)

[しきい値の指定](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

acctsuspend

`acctsuspend` は、アカウントティングの実行中に使用可能になっている必要のあるファイル・システムの空きスペースの最小値を指定します。

指定可能な値:

最小値	100
最大値	100
デフォルト値	2

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

このパラメータは、システム上でプロセス・アカウントングが使用されている場合に限り有効です。

acctsuspend は、他のプロセスが必要とするファイル・システムの空きスペースがアカウントング・ファイルに使用されるのを防止します。このパラメータは、プロセス・アカウントングの実行中に他の用途向けに確保しておく必要のある使用可能なファイル・システム・スペースの最小値を指定します。使用可能なスペースがこの値を下回った場合、アカウントングは中断され、十分なスペースが再び使用可能になった時点でアカウントングが再開されます(再開に必要なスペースは [acctresume](#) で定義します)。

アカウントングの中断/再開方法の概要については、[アカウントングの中断と再開の概要](#)を参照してください。

acctsuspend に値を指定する

アカウントングを中断するしきい値は、ファイル・システム内の空きスペース 0 から使用可能な領域すべてを示す値まで任意の値に設定できます。ただし、acctsuspend は [minfree](#) の値と関連付けて指定します。値の設定方法については、[しきい値の指定](#)を参照してください。

関連するパラメータ

acctsuspend と acctresume は互いに関連するパラメータです。アカウントングの中断と再開との矛盾を防止するため、acctsuspend の符号付き整数値は必ず、acctresume の符号付き整数値より大きくなければなりません。

付加情報

[アカウントングの中断と再開の概要](#)

[アカウントングに関する構成可能パラメータの一覧](#)

[しきい値の指定](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

acctresume

acctresume は、アカウントングを再開する前に使用可能になっている必要のあるファイル・システムの空きスペースを指定します。

指定可能な値:

最小値	100
最大値	101
デフォルト値	4

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

このパラメータは、システム上でプロセス・アカウントングが使用されている場合に限り有効です。

`acctresume` は、空きスペース不足が原因で中断されていたプロセス・アカウントングをシステムが再開する前に、使用可能になっている必要のあるファイル・システムの空きスペースの最小値を指定します。アカウントングが中断されるしきい値は、`acctsuspend` で定義します。

アカウントングの中断/再開方法の概要については、[アカウントングの中断と再開の概要](#)を参照してください。

`acctresume` に値を指定する

アカウントングを再開するしきい値は、ファイル・システム内の空きスペース 0 から使用可能な領域すべてを示す値まで任意の値に設定できます。ただし、`acctresume` は `minfree` の値と関連して指定されており、`acctsuspend` よりも大きい値であることが必要です (下記の「関連するパラメータ」を参照)。値の設定方法については、[しきい値の指定](#)を参照してください。プロセス・アカウントングは、(`acctresume` と `minfree` の和で指定された)十分なスペースが使用可能になった時点で再開されます。合計が 100%、または通常の最大空きスペースを超える場合は、アカウントングを再開できません。

関連するパラメータ

`acctsuspend` と `acctresume` は互いに関連するパラメータです。アカウントングの中断と再開との矛盾を防止するため、`acctresume` の符号付き整数値は必ず、`acctsuspend` の符号付き整数値より大きくなければなりません。

付加情報

[アカウントング・パラメータの概要](#)

[アカウントングに関する構成可能パラメータの一覧](#)

[しきい値の指定](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

チュートリアル: アカウントングのしきい値の指定

`acctsuspend` および `acctresume` は整数値で、それぞれアカウントング・ログ・ファイルが置かれるファイル・システムの全使用可能領域の割合をパーセンテージで指定します。このパーセンテージを現在の `minfree` の値に加えることで、プロセス・ア

カウンティングが稼働し続けるために、全ファイル・システム空間の何パーセントが利用可能でなければならないかが決まります。

しきい値は、acctsuspend または acctresume に指定した値を minfree の現在値に加えることで決定されます。いずれかのパラメータの値が 0 より小さいと、その値は空き領域の必要量を減らし、minfree で予約され保護されている最小の空き領域を、アカウントング・ファイルが使用できるようになります。acctsuspend と minfree の和が 0 または負の値である場合には、アカウントングはファイル・システム全体が満杯になるまで続き、acctresume の値は無意味になります。

しきい値の計算

acctsuspend はアカウントング中断のしきい値に対応し、acctresume は、アカウントング再開に対応するしきい値に対応します。これらのパラメータの値は、対応するしきい値と、アカウントング・ログ・ファイルが格納されているファイル・システムに設定されている minfree の現在値の間の差を表します。

- acctsuspend および acctresume に 0 または正の値を指定すると、それらの値は minfree の現在の値に加算されます。その計算結果が、中断または再開のしきい値 (ファイル・システム全体の使用可能サイズに対するパーセンテージ) として適用されます。この場合、アカウントングに空きスペース領域が使用されることはありません。
- acctsuspend および acctresume に負の値を指定すると、minfree の現在の値に加算されます (指定した値は負であるため、結果としては minfree よりも小さくなります)。アカウントングの中断や再開は、それぞれ予約された最小の空きスペースの領域内で起こります。
- アカウントング中断のしきい値に 0 または負の値を指定すると、ファイル・システムがオーバーフローするまで、アカウントングが継続して実行されます。
- 再開のしきい値に、100%以上あるいはどの時点であれ期待されるファイル・システムの空き領域よりも大きな値を与えると、システムが再ブートされるか、アカウントングが startup コマンド (HP-UX Reference の acctsh(1M)を参照) を使って手動で再起動されない限り、アカウントングが再開しなくなります。

例:

minfree = 10%

acctsuspend = 5%

acctresume = 8%

ファイル・システム全体の利用可能領域が 15% より小さいとアカウントングを中断し、利用可能領域が 18% に達するとアカウントングを再開します。

minfree = 5%

acctsuspend = -3% (minus 3%)

acctresume = 0%

ファイル・システム全体の利用可能領域が 2% 未満になるとアカウントングを中断し、利用可能領域が minfree に達するとアカウントングを再開します。

minfree = 5%

acctsuspend = 0%

acctresume = 3%

ファイル・システムの全体の利用可能領域が minfree まで減少するとアカウントリングを中断し、利用可能領域がファイル・システム全体のサイズの 3%に minfree を加えたところに達するとアカウントリングを再開します。

minfree = 5%

acctsuspend = -5% (minus 5%)

acctresume = 101%

ファイル・システムから空きスペースが完全になくなったときにアカウントリングを中断します。アカウントリングの再開は行われません (空きスペースがファイル・システム全体のサイズの 106%に達したときにアカウントリングを再開するように指定していますが、これは現実には成立しない条件です)。

minfree がアカウントリングのしきい値に及ぼす影響

acctsuspend および acctresume では、アカウントリング・ファイルが格納されているファイル・システムに対して設定されている minfree パラメータの値に基づいて、アカウントリングのしきい値を指定します。minfree は、ファイル・システムの作成時に指定されるパラメータです。このパラメータの値によって、ファイル・システムのうち、通常のユーザーが使用できない予約スペースのパーセンテージが決定されます。このパラメータには、-100%~100%の値を指定できます。

HP-UX ファイル・システムの場合、minfree のデフォルト値は 10%です。minfree の現在の値を確認するには、df コマンドを -t オプション付きで実行します (HP-UX Reference の df(1M)を参照してください)。この値は、ファイル・システム・データ構造 (fs(4)を参照) の fs_minfree の現在値によって決定されます。

例えば、マウントされているファイル・システム/users に設定されている minfree の現在値を確認するには、df コマンドを次のように実行します。

```
df -t /users
```

minfree がアカウントリングのしきい値に及ぼす影響

[アカウントリング・パラメータの概要](#)

[アカウントリング・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

構成可能パラメータの値の指定方法

構成可能カーネル・パラメータを設定するには、整数値を直接指定するか、または有効な整数値を返す数式を指定する必要があります。

どのパラメータの値も整数値として指定できます。また、大半のパラメータは、数式を使って設定することもできます。ただし、常に数式を使って設定するのは、ごく一部のパラメータだけです。

値の入力

整数値または数式を入力して構成可能パラメータを設定するには、以下の手順に従ってください。

1. Kernel Configuration: Configurable Parameters ウィンドウの上端にある Actions プルダウン・メニューから Modify Configurable Parameter を選択します。
2. Modify Configurable Parameter ダイアログ・ボックスが表示されたら、Specify New Formula/Value を選択し、Formula/Value のラベルが付いたボックスをクリックします。
3. 使用する値または数式を入力します。このとき、[Backspace]キーで現在の値の文字を削除し、新しい値の文字を入力します。
4. 新しい値を確定してダイアログ・ボックスを閉じるには、OK ボタンを使用します。元の値を変更せずにダイアログ・ボックスを閉じるには、Cancel を使用します。

整数値

整数値を指定するには、Modify Configurable Parameter ウィンドウの Formula/Value:ボックスに正確な値を入力します。

整数値を返す数式

数式を使ってパラメータ値を指定する場合は、整数式でなければなりません。これは、式の中のすべての要素は、プログラミング言語 C で定義されている整数でなければならないということです。

構成可能パラメータの名前は通常、小文字で示しますが、ほかの構成可能パラメータの値を数式の中で参照するときは、パラメータの名前の先頭またはすべてを大文字にします。例えば、npty の値を C 言語の #define 文で参照するときは、Npty と記述します。例えば、カーネル記述ファイル内で nfile を設定するときには、次のような数式を指定できます。

```
nfile ((16*(Nproc+16+MaxUsers)/10)+32+2*Npty)
```

ここで、Nproc は nproc の定義値、MaxUsers は maxusers の定義値、Npty は npty の定義値をそれぞれ表しています。この大文字/小文字の規則に該当しない例外もありますが、ほとんどの場合はこの規則が適用されます。

カーネル記述ファイル内では、ホワイトスペース (スペースおよびタブ) を使用することはできません。値ボックスに入力した数式にホワイトスペースが含まれている場合、不要であれば SAM により削除されます。

非同期 I/O に関するパラメータの概要

非同期 I/O 処理の管理に使用するカーネル・パラメータを以下に示します。最初の 4 つのパラメータは POSIX 非同期 I/O 処理に関連するパラメータで、最後のパラメータはプロセスと非同期ディスク I/O ドライバ間のポートのオープンに関連するパラメータです。

[aio_listio_max](#)

1 回の `listio()` コール中に許される POSIX 非同期 I/O 処理数

`aio_max_ops`

システム全体で同時に許される POSIX 非同期 I/O の最大操作数

`aio_physmem_pct`

POSIX 非同期 I/O 処理用にロック可能なシステム・メモリの最大容量

`aio_prio_delta_max`

POSIX 非同期 I/O 制御ブロック(`aiocb`)で許容される優先順位からの最大オフセット

`max_async_ports`

プロセスがオープンできる非同期ディスク I/O ドライバの最大ポート数

付加情報

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

`aio_listio_max`

`aio_listio_max` は、`listio()` コールで指定できる POSIX 非同期 I/O 処理数の最大値を指定します。

指定可能な値:

最小値	2
最大値	0x10000
デフォルト値	256

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは、1 回の `listio()` コールで大量の POSIX 非同期 I/O 処理が要求された場合に使用可能なシステム・リソースの最大値を設定します。この値は、システムのプログラミング要件を満たすのに十分な値に設定することが必要です。同時に、プロセスの誤動作による過剰な非同期 I/O 処理からのシステムの保護を考慮することも必要です。

このパラメータには、[`aio_max_ops`](#) を超える値を指定してはいけません。

POSIX 非同期 I/O の詳細については、HP-UX Reference の [`aio\(5\)`](#) を参照してください。

付加情報

[非同期 I/O に関するパラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

aio_max_ops

aio_max_ops は、同時に待ち行列に入れることができる POSIX 非同期 I/O 処理のシステム全体における最大数を指定します。

指定可能な値:

最小値	1
最大値	0x100000
デフォルト値	2048

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは、同時に大量の POSIX 非同期 I/O 処理がシステム上で待ち行列に入れられた場合に消費されるシステム・リソースに上限を設定します。このパラメータでは、大量の非同期 I/O 処理とそれに必要なメモリによってシステムに悪影響を及ぼす競合プロセスの能力を制限します。

待ち行列に入った非同期処理はそれぞれ、内部制御構造に対するシステム・メモリの割り当てを必要とします。このため、上記パラメータによる非同期処理数の制限が必要となります。システム全体における制限以外に、プロセスごとの制限もあります。プロセスごとの制限値は、getrlimit() や setrlimit() の呼び出し時に使用する引き数 RLIMIT_AIO_OPS により制御されます。

[aio_listio_max](#) は、あるプロセスからの 1 回の listio() コールに含めることができる処理数を制限します。aio_max_ops には、システム全体のバランスを崩さずに、listio() コールを同時またはほぼ同時に発行するすべてのプロセスからの妥当な量の要求を満たすのに十分な値を指定する必要があります。

POSIX 非同期 I/O の詳細については、HP-UX Reference の aio(5) を参照してください。

付加情報

[非同期 I/O に関するパラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

aio_physmem_pct

`aio_physmem_pct` は、POSIX 非同期 I/O 処理で使うためのロック可能なシステム内の全物理メモリの最大値をパーセンテージで指定します。

指定可能な値:

最小値	5
最大値	50
デフォルト値	10

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは、ある時点で進行中のすべての POSIX 非同期 I/O 処理によってロック可能なシステム・メモリの最大値を設定します。処理が完了している場合でも、この処理に対する `aio_return()` コールによって処理が正常に終了されるまで、この処理はアクティブな待ち行列上に残っており、メモリが解放されていないことに注意してください。

I/O に対して要求-コールバック機構を使用する非同期 I/O 処理では、使用中のメモリをロックできなければなりません。要求-コールバック機構が使用されるのは、関連するデバイス・ドライバがこの機構をサポートしている場合だけです。メモリは、I/O の転送中のみロックされます。大規模なサーバーの場合、`aio_physmem_pct` を大きめの値(最大 50)に設定することをお勧めします。

`aio_physmem_pct` は、ロック可能な物理メモリに対してシステム全体の最大値を強制的に設定します。プロセスごとにロック可能なメモリについても、アプリケーション・プログラム自身の中で `setrlimit()` コールを使って制限できます(HP-UX Reference の `setrlimit(2)` を参照)。

非同期 I/O 用だけでなく、任意の理由によりロック可能なメモリの合計は、システム全体の制限値 `lockable_mem` によって制御されることにも注意してください。これ以外にも、`plock()` および `mlock()` インタフェースによる明示的メモリ・ロックをはじめとするシステムの動作により、ロック可能なメモリの量が影響を受ける場合があります。

`lockable_mem` という名称のカーネル・パラメータはありませんが、これに影響を与えるパラメータとして `unlockable_mem` があります。`lockable_mem` の値は、システムの起動後に使用可能なシステム・メモリの量から `unlockable_mem` の値を差し引いた値になります。システムの起動時に、ロック可能なメモリの量(および使用可能なメモリと物理メモリの量)がシステム・コンソールに表示されます。これらの値は、`/sbin/dmesg` コマンドを使ってシステムを実行している際に取得できます。

POSIX 非同期 I/O の詳細については、HP-UX Reference の `aio(5)` を参照してください。また、システム・メモリ管理に関するホワイトペーパーも参照してください。ファイル `/usr/share/doc/mem_mgt.txt` として収められています。システム管理の概念に関するその他のホワイトペーパーも同じディレクトリに収められています。

関連するパラメータ

`aio_physmem_pct` によって設定された範囲内でロック可能なメモリの量は、[unlockable_mem](#) によって設定されるシステム全体のロック可能なメモリの上限を超えることはできません。

付加情報

[非同期 I/O に関するパラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

aio_prio_delta_max

aio_prio_delta_max は、POSIX 非同期 I/O 処理に対するスローダウン要素(優先順位のオフセット)の最大値を指定します。このパラメータは、非同期 I/O 制御ブロック(aiocb 構造)内で許容される優先順位の最大オフセット値です。

指定可能な値:

最小値	0
最大値	20
デフォルト値	20

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは、POSIX 非同期 I/O 処理の優先順位をどの程度まで下げられるかを設定します。このパラメータにより、非同期 I/O 制御ブロック構造 aiocb 内の int aio_reqprio に対して許容される値が制限されます。

POSIX 非同期 I/O の詳細については、HP-UX Reference の aio(5)を参照してください。

付加情報

[非同期 I/O に関するパラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

カーネル・パニック・ダンプに関するパラメータの概要

カーネル・パニックが発生し、システム・メモリの一部がディスクにダンプされるときには、以下のパラメータの値に基づいてダンプ処理が行われます。

[alwaysdump](#)

カーネル・パニックの発生時にどのクラスのシステム・メモリをダンプするかを指定します。

dontdump

カーネル・パニックの発生時にどのクラスのシステム・メモリをダンプから除外するかを指定します。

initmodmax

システム・クラッシュ・ダンプ・データの処理時に savecrash によって処理されるカーネル・モジュールの最大数を指定します。

modstrmax

モジュールの名前とファイル・システム内での位置を格納する savecrash カーネル・モジュール・テーブルの最大サイズをバイト単位で指定します。

付加情報

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

alwaysdump

alwaysdump は、カーネル・パニックが発生した場合にどのクラスのカーネル・メモリ・ページをダンプするかをビットマップ値で指定します。

指定可能な値:

最小値	0
最大値	なし
デフォルト値	0

このパラメータには、[整数値を指定してください](#)。詳細は、[パラメータ値の指定](#)を参照してください。

説明

大規模なシステムでは、カーネル・パニック発生時のシステム・メモリのダンプに要する時間が非常に長くなり、システムにインストールされている物理メモリの量によっては、現実的に許容できない長さになる可能性があります。2 種類のパラメータ dontdump と alwaysdump で制御する高速ダンプ機能により、カーネル・ダンプを以下に示す特定の種類の情報に限定することができます。

- 未使用の物理メモリ
- カーネルの静的データ
- カーネルの動的データ
- ファイル・システムのメタデータ
- カーネル・コード
- バッファ・キャッシュ

- プロセス・スタック
- ユーザー・プロセス

alwaydump に保存されるビットマップ値により、カーネル・パニック時に上記のどのメモリ・クラスをメモリ・ダンプの対象とするかを指定します。

関連するパラメータ

alwaydump と dontdump とは、相反する効果を与えるパラメータです。あるメモリ・ページ・クラスに対応するビットを一方のパラメータで設定した場合、もう一方のパラメータで設定してはいけません。両方のパラメータで設定すると、矛盾が起こり、パラメータ値に基づく実際のカーネル動作は未定義の状態となります。SAM を使って値を設定した場合は、このような矛盾は起こりません(SAM の Kernel Configuration:Dump Devices ウィンドウにある[Actions]メニューで **[Modify Page-Class Configuration]** を選択)。

付加情報

[カーネル・パニック・ダンプに関する構成可能パラメータ](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

dontdump

dontdump は、カーネル・パニックの発生時にどのクラスのカーネル・メモリ・ページをダンプ対象外とするかをビットマップ値で指定します。

指定可能な値:

最小値	0
最大値	なし
デフォルト値	0

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

大規模なシステムでは、カーネル・パニック発生時のシステム・メモリのダンプに要する時間が非常に長くなり、システムにインストールされている物理メモリの量によっては、現実的に許容できない長さになる可能性があります。2 種類のパラメータ dontdump と alwaydump で制御する高速ダンプ機能により、カーネル・ダンプを以下に示す特定の種類の情報に限定することができます。

- 未使用の物理メモリ
- カーネルの静的データ
- カーネルの動的データ

- ファイル・システムのメタデータ
- カーネル・コード
- バッファ・キャッシュ
- プロセス・スタック
- ユーザー・プロセス

alwaysdump に保存されるビットマップ値により、カーネル・パニック時に上記のどのメモリ・クラスをメモリ・ダンプの対象とするかを指定します。

関連するパラメータ

alwaysdump と dontdump とは、相反する効果を与えるパラメータです。あるメモリ・ページ・クラスに対応するビットを一方のパラメータで設定した場合、もう一方のパラメータで設定してはいけません。両方のパラメータで設定すると、矛盾が起こり、パラメータ値に基づく実際のカーネル動作は未定義の状態となります。SAM を使って値を設定した場合は、このような矛盾は起こりません(SAM の Kernel Configuration:Dump Devices ウィンドウにある[Actions]メニューで **[Modify Page-Class Configuration]** を選択)。

付加情報

[カーネル・パニック・ダンプに関する構成可能パラメータ](#)
[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

initmodmax

initmodmax は、カーネル・パニックの発生によりシステム・メモリのダンプが行われた場合に savecrash コマンドが処理するカーネル・モジュールの最大数を指定します。

指定可能な値:

最小値	0
最大値	なし
デフォルト値	50

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

カーネル・パニック(システム・クラッシュ)が発生した場合、システムがシャットダウンする前に、システム・メモリの指定された領域がダンプ・デバイスにコピーされます。このとき、savecrash コマンドを使って、ダンプ領域をファイル・システム内のディレクトリにコピーできます(ただし、システム容量とダンプされたメモリ・クラスによっては、大量のスペースが必要となります)。

カーネルが、動的にロード可能なカーネル・モジュール(カーネルに静的にはインストールされないドライバ)を含む場合、savecrash はそれらのモジュールに関する情報を保存するため、その構造体のひとつにスペースを確保しなくてはなりません。ロードされるモジュールの数を予測する方法はないため、このパラメータでは、savecrash が処理できる上限値を設定します。

initmodmax がロードされたカーネル・モジュール数より少なく設定されている場合、上限値に達するまでに遭遇したモジュールだけが savecrash によって処理されます。したがって、システム管理者は、システム動作時にロードされるカーネル・モジュールの数を追跡して、カーネル・パニックによるダンプの発生時に、モジュールを適切に処理するための十分な値を initmodmax に設定することが必要です。

このパラメータは、savecrash コマンドの動作のみに影響を与えます。したがって、通常システム動作時にカーネルにロード可能なモジュール数は制限されません。

付加情報

[カーネル・パニック・ダンプに関する構成可能パラメータ](#)
[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

modstrmax

modstrmax は、モジュールの名前とファイル・システム内での位置を格納するカーネル・モジュール・テーブル savecrash の最大サイズを指定します。

指定可能な値:

最小値	500
最大値	なし
デフォルト値	500

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

カーネル・パニック(システム・クラッシュ)が発生した場合、システムがシャットダウンする前に、システム・メモリの指定された領域がダンプ・デバイスにコピーされます。このとき、savecrash コマンドを使って、ダンプ領域をファイル・システム内のディレクトリにコピーできます(ただし、システム容量とダンプされたメモリ・クラスによっては、大量のスペースが必要となります)。

カーネルが、動的にロード可能なカーネル・モジュール(カーネルに静的にはインストールされないドライバ)を含む場合、savecrash はモジュール名とファイル・システム上のモジュールの位置を追跡するためのスペースを確保します。このスペースには、モジュールが置かれているディレクトリの絶対パスとモジュール名が格納されます。このディレクトリに複数のモジュールが見つかった場合でも、保存されるディレクトリ・パスのコピーは 1 つだけなので、スペースが最適な状態で使用されます。

システムにモジュールが追加され、モジュール名が長くなったり、モジュールがファイル・システム内に散在する場合、増加するデータに見合うよう modstrmax の値を大きくすることが必要です。

このパラメータは savecrash コマンドの動作のみに影響を与えることに注意してください。つまり、通常のシステム動作時にカーネルにロードされるモジュール数を制限するものではありません。

付加情報

[カーネル・パニック・ダンプに関する構成可能パラメータ](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

FibreChannel に関するカーネル・パラメータの概要

Fibre-Channel SCSI サブシステム および、Fibre-Channel Protocol (FCP)を使って相互作用するシステムプロセッサとペリフェラルデバイスとの通信には、2つのカーネル構成パラメータが関係しています。この2つのパラメータは、FCP による読み取り、書き込み、制御といった各種要求の同時実行をサポートするために割り当てられるメモリの量とタイプのデフォルト設定に使われます。

num_tachyon_adapters

FCP 要求用に割り当てられるメモリ領域は、特定の要求条件を満たす必要があります。num_tachyon_adapters パラメータは I/O 仮想アドレッシングを提供していないシステムで、これらの要求を満たすメモリを確保するために使われます。このパラメータに指定した値によって、割り当てるメモリのタイプが決定され、システム内の Tachyon ベースの Fibre-Channel アダプタの数が決定されます。

max_fcp_reqs

FCP アダプタ上で許容される同時 FCP 要求の数は、デフォルトでは、512 までに制限されます。この制限を変更するには、max_fcp_reqs を使います。同時要求の制限数は、構成、デバイスの特性、I/O 負荷、ホスト・メモリ、その他 FPS ソフトウェア側では容易に判別できない値など、さまざまな要素を考慮して適切に設定する必要があります。

付加情報

[LVM パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

num_tachyon_adapters

num_tachyon_adapters は、システムで I/O 仮想アドレッシングがサポートされていない場合にシステム内で使用する FibreChannel Tachyon アダプタの数を指定します。

指定可能な値:

最小値	0
最大値	5
デフォルト値	0

整数値を直接指定するか、または整数式を指定してください。システムが I/O 仮想アドレッシングをサポートしていない場合には、0 でない値を指定する必要があります。システム内にインストールされている Tachyon FCP アダプタの数に一致する値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

num_tachyon_adapters には、Tachyon FCP アダプタがシステムにいくつインストールされているかを指定します。これにより、システムが I/O 仮想アドレッシングを提供していない場合でも、システムの起動時に適切な量のメモリが割り当てられます。

num_tachyon_adapters の値の指定

システムが I/O 仮想アドレッシングを提供していない場合には、マシンに実際にインストールされている Tachyon FCP アダプタの数を num_tachyon_adapters に指定します。これにより、システムのブート時に、その数のアダプタに見合った量のメモリが予約されるようになります。この量は、[max_fcp_reqs](#) の値に応じて変わります。

システムが I/O 仮想アドレッシングを提供している場合は、このパラメータを 0 に設定してください。これにより、メモリが必要になった時点で自動的に割り当てられるようになります。

システムが I/O 仮想アドレッシングをサポートしているかどうか不明の場合に、このパラメータを暫定的に 0 でない値に設定しても特に問題は発生しません。ただし、このパラメータに設定した値がマシンに実際にインストールされている Tachyon FCP アダプタの数よりも大きい場合には、その差分量に相当するメモリをほかの用途に使用できなくなるので、注意が必要です。

関連するパラメータとシステム値

Tachyon FCP アダプタ用に割り当てるメモリは、num_tachyon_adapters の値と max_fcp_reqs の値の組み合わせに基づいて決定されます。

付加情報

[Fibre-Channel に関するカーネル・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

max_fcp_reqs

max_fcp_reqs は、マシン上にインストールされている各 FCP アダプタで同時に処理する Fibre-Channel FCP 要求の最大許容数を定義します。

指定可能な値:

最小値	0
最大値	1024
デフォルト値	512

整数値を直接指定するか、または整数式を指定してください。max_fcp_reqs に設定した値が小さすぎてアダプタが正しく機能できない場合には、自動的に必要最小限の値に変更されます。詳細は、[パラメータ値の指定](#) を参照してください。

説明

max_fcp_reqs は、1 つの FCP アダプタで同時に処理する FCP 要求の最大許容数を指定します。システム出荷時のデフォルト値は、512 です。この値は、実際の構成に応じて変更できます。同時要求の最適な最大許容数は、構成、デバイスの特性、I/O 負荷、ホストのメモリ、および FCP ソフトウェアが容易に決定できないその他の値といった、いくつかの異なる要因に依存します。

関連するパラメータとシステム値

Tachyon FCP アダプタ用に割り当てるメモリは、num_tachyon_adapters の値と max_fcp_reqs の値の組み合わせに基づいて決定されます。

付加情報

[Fibre-Channel に関するカーネル・パラメータの概要](#)
[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ファイル・システムに関するカーネル・パラメータの概要

ファイル・システム性能に関する構成可能なパラメータは、以下のカテゴリに分類されます。

ファイル・システム・バッファ

システムの物理メモリ・リソースをファイル・システムのバッファ・キャッシュ・スペース用に割り当てます。このカテゴリのパラメータには、動的バッファ割り当てパラメータ [dbc_min_pct](#)、[dbc_max_pct](#) と、静的バッファ割り当てパラメータ [bufpages](#)、[nbuf](#) があります。詳しくは、[構成可能なファイル・システムのパラメータ](#) を参照してください。

ディスクの先読み

ディスクの先読み操作で使用可能なメモリ量を制御します。パラメータ [hfs_ra_per_disk](#) と [hfs_max_ra_blocks](#) は、HFS における順方向の先読みを制御し、パラメータ [hfs_revra_per_disk](#) と [hfs_max_revra_blocks](#) は、逆方向の先読みを制御します。パラメータ [vxfs_ra_per_disk](#) と [vxfs_max_ra_kbytes](#) は、VxFS における先読みを制御します。

SCSI デバイス

SCSI デバイスへのアクセスを制御します。このカテゴリのパラメータには、[scsi_maxphys](#)、[scsi_max_qdepth](#)、[st_fail_overruns](#)、[st_large_recgs](#) などがあります。

ファイルのオープンとロック

同時にオープンまたはロック可能なファイル数を指定します。このカテゴリのパラメータには、[maxfiles](#)、[maxfiles_lim](#)、[nfile](#)、[nflocks](#)、[ninode](#) があります。詳しくは、[構成可能なファイル・システムのパラメータ](#) を参照してください。

ジャーナル・ファイル・システム

VxFS ファイル・システムの i ノードに関するディレクトリ名検索キャッシュ(DNLC)にスペースを割り当てます。このカテゴリのパラメータには [vx_ncsize](#) があります。

非同期書き込み

ファイル・システムへの非同期書き込みを許可/禁止します。このカテゴリのパラメータには [fs_async](#) があります。詳しくは、[構成可能なファイル・システムのパラメータ](#) を参照してください。

付加情報

[構成可能なファイル・システムのパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ファイル・システムに関するカーネル・パラメータの一覧

ファイル・システム管理に関するさまざまな設定は以下のパラメータで制御できます。

バッファ・キャッシュに関するパラメータ・グループ

[bufpages](#)

ファイル・システムのバッファ・キャッシュ内の 4 KB 単位のページ数

[dbc_min_pct](#)

動的バッファ・キャッシュに使用するメモリの最低パーセンテージ

[dbc_max_pct](#)

POSIX 非同期 I/O 処理用にロック可能なシステム・メモリの最大容量

nbuf

POSIX 非同期 I/O 制御ブロック(aiocb)で許容される優先順位からの最大オフセット

ディスクの先読みに関するパラメータ・グループ

hfs_ra_per_disk

HFS ファイル・システムにおけるディスク・ドライブあたりの先読みデータ量を、KB 単位で設定します。

hfs_max_ra_blocks

単一の HFS ファイル・システムについて、カーネルが未処理のまま先読みできる最大ブロック数を設定します。

hfs_revra_per_disk

HFS ファイル・システムにおけるディスク・ドライブあたりの逆方向の先読みデータ量を、KB 単位で設定します。

hfs_max_revra_blocks

単一の HFS ファイル・システムについて、カーネルが未処理のまま逆方向に先読みできる最大ブロック数を設定します。

vxfs_ra_per_disk

VxFS ファイル・システムにおけるディスク・ドライブあたりの先読みデータ量を、KB 単位で設定します。

vxfs_max_ra_kbytes

単一の VxFS ファイル・システムについて、カーネルが未処理のまま先読みできる最大データ量を、KB 単位で設定します。

SCSI デバイスに関するパラメータ・グループ

scsi_maxphys

SCSI I/O サブシステムにおける最大レコード・サイズを、バイト単位で設定します。

scsi_max_qdepth

SCSI デバイスについて、待ち行列に入れることのできる SCSI コマンドの最大数を設定します。

st_fail_overruns

データ・オーバーランが発生した SCSI テープ読み込みはエラーとなります。

st_large_recs

SCSI テープにおける大容量レコードのサポートを可能にします。

ファイルのオープンとロックに関するパラメータ・グループ

maxfiles

単一プロセスがオープンまたはロックできるファイル数の論理制限値

maxfiles_lim

単一プロセスがオープンまたはロックできるファイル数の物理制限値

nfile

システム上で同時にオープン可能な最大ファイル数

nflocks

すべてのプロセスに対してシステム全体でロック可能な最大ファイル数

ninode

メモリ内でオープン可能な i ノードの最大数

非同期書き込みに関するパラメータ・グループ

fs_async

ファイル・システムのデータ構造を同期または非同期のどちらでディスクへ書き込むかを指定します。

VxFS (ジャーナル化) ファイル・システムに関するパラメータ

vx_ncsize

VxFS のディレクトリ・パス名キャッシュ用に予約するメモリ・スペース

付加情報

大容量記憶に関するパラメータの概要

ファイルのオープンとロックに関する構成可能パラメータ

構成可能カーネル・パラメータ・ヘルプのブラウズ

bufpages

bufpages は、ファイル・システム・バッファ・キャッシュに使用する 4096 バイトのメモリ・ページの数指定します。

指定可能な値:

最小値	0 または 6 (nbuf×2 または 64 ページ)
最大値	メモリ・サイズに依存。
デフォルト値	0

このパラメータには、整数を直接指定するか、または整数式を指定します。このパラメータに 0 でない値を指定するのは、動的バッファ・キャッシュを使用しない場合だけです。詳細は、パラメータ値の指定 を参照してください。

説明

bufpages は、ファイル・システム・バッファ・キャッシュ用に割り当てる 4096 バイトのメモリ・ページの数を指定します。これらのバッファは、すべてのファイル・システム I/O 操作に使用されるとともに、システムの他のすべてのブロック・モードの I/O 処理 (exec() および mount()) による処理、i ノードの読み込みや一部のデバイス・ドライバ) にも使用されます。

bufpages の値の指定

動的バッファ・キャッシュを割り当てる場合は、bufpages を 0 に設定します。静的バッファ・キャッシュとして固定した量のメモリを割り当てるには、bufpages に 4 K バイトのページ数を設定します。bufpages に 0 でない 64 未満の正の値を指定した場合、その数はブート時に増やされ、変更を知らせるメッセージが表示されます。bufpages がシステムがサポートする最大値よりも大きい場合、その数はブート時に減らされ、変更を知らせるメッセージが表示されます。

関連するパラメータとシステム値

bufpages は、nbuf と互いに関連しています。その関係を以下に示します。

- bufpages = 0 かつ nbuf = 0 の場合: 動的バッファ・キャッシュを使用可能にします。
- bufpages ≠ 0 かつ nbuf = 0 の場合: システム・ブート時に、bufpages×4 K バイトのバッファ・プール・スペースを割り当てて、bufpages/2 個のバッファ・ヘッダを作成します。
- bufpages = 0 かつ nbuf ≠ 0 の場合: ブート時に、nbuf×2 ページのバッファ・プール・スペースを割り当てて、nbuf 個のヘッダを作成します。
- bufpages ≠ 0 かつ nbuf ≠ 0 の場合: ブート時に、bufpages ページのバッファ・プール・スペースを割り当てて、nbuf 個のバッファ・ヘッダを作成します。2 つのパラメータの値が衝突しており、その両方を使ってシステムを構成できない場合は、bufpages の値が優先的に適用されます。

bufpages は、バッファ・プールに実際に割り当てるメモリの量を決定します。nbuf を参照してください。

bufpages に 0 を指定すると、nbuf で定義したバッファ・ヘッダのそれぞれに対して 2 つのページがシステム・ブート時に割り当てられます。bufpages と nbuf をともに 0 に設定すると、動的バッファ・キャッシュ割り当てが有効になり、システムのバッファ要求に応じて、利用可能なメモリの一部が動的にバッファ・キャッシュとして割り当てられます。動的に割り当てられるメモリの量は、dbc_min_pct が下限となり、dbc_max_pct が上限となります。

バッファ・プールに割り当てることのできるメモリの最大量は、メモリのうち、ほかの目的でシステムに割り当てられているメモリの量にも依存します。したがって、システム・メモリに関連するパラメータの値を変更すると、バッファ・プール用に利用できるメモリの最大量が変化する可能性があります。

付加情報

大容量記憶に関するパラメータの概要

ファイル・システムおよび大容量記憶に関するパラメータの一覧

構成可能カーネル・パラメータ・ヘルプのブラウズ

dbc_min_pct

dbc_min_pct は、動的バッファ・キャッシュに使用するメモリの割合の下限をパーセンテージで定義します。

指定可能な値:

最小値	2
最大値	90
デフォルト値	5

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

ファイル・システムの I/O 操作中にデータを格納するバッファ・キャッシュは、ブート時に固定サイズで割り当てるか、または、動的に割り当てるように構成できます。bufpages パラメータおよび nbuf パラメータがともにデフォルト値の 0 に設定されている場合は、システム・メモリに対する要求の競合に応じて、バッファ・キャッシュのサイズが動的に拡大または縮小されません。

dbc_min_pct では、動的バッファ・キャッシュ用に予約する物理メモリの割合の下限をパーセンテージとして指定します。

dbc_min_pct と dbc_max_pct は同じ値に設定することも可能です。例えば、この 2 つのパラメータをともに 12 に設定すると、物理メモリのサイズに関係なく、常に 12% の物理メモリをバッファ・キャッシュに割り当てるようにカーネルが構成されます。

適切な値の決定

dbc_min_pct を小さすぎる値に設定した場合、バッファ・キャッシュの要求が大すぎると、実際にシステムがハングします。また、固定バッファ・キャッシュを使用している場合にも同様な問題が発生する可能性があります。合理的な必要最小限のキャッシュ・サイズ (MB 単位) を決定するには、次の式を使います。

$$(\text{システム・プロセスの数}) \times (\text{ファイル・システムの最大ブロック・サイズ}) \div 1024$$

dbc_min_pct の値を決定するには、上の数式で得た値をコンピュータにインストールされている物理メモリのサイズ (MB 単位) で割り、さらに、その値に 100 を掛けてパーセント値を算出します。

この計算には、ディスク I/O の頻度の高いプロセスだけを含めるようにしてください。ほかのプロセスは除外してかまいません。計算に含めるべきプロセスと除外すべきプロセスの例を以下に示します。

計算に含めるプロセス:

システム上にマウントされている 1 つまたは複数のファイル・システム内のソース・ファイルや出力ファイルの一方または両方をアクセスまたは使用するプロセス。例えば、NFS デーモン、nroff のようなテキスト・フォーマッタ、データベース管理アプリケーション、テキスト・エディタ、コンパイラなどがこれに該当します。

計算から除外するプロセス:

ディスク I/O を行う頻度が非常に低いプロセス。例えば、X ディスプレイ・アプリケーション、hpterm、rlogin、ログイン・シェル、システム・デーモン、telnet 接続または uucp 接続などがこれに該当します。

関連するパラメータ

dbc_min_pct には、dbc_max_pct 以下の値を指定する必要があります。

動的バッファ・キャッシュを使用するには、bufpages と nbuf をともに 0 に設定する必要があります。

付加情報

大容量記憶に関するパラメータの概要

ファイル・システムおよび大容量記憶に関するパラメータの一覧

構成可能カーネル・パラメータ・ヘルプのブラウズ

dbc_max_pct

dbc_max_pct は、動的バッファ・キャッシュに使用するメモリの割合の上限をパーセンテージで定義します。

指定可能な値:

最小値	2
最大値	90
デフォルト値	50

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定 を参照してください。

説明

bufpages パラメータおよび nbuf パラメータの両方にデフォルト値の 0 が設定されていると、システム・メモリに対する競合した要求に応じて、バッファ・キャッシュのサイズが動的に増減します。

dbc_max_pct では、動的バッファ・キャッシュ用に予約する物理メモリの割合の上限をパーセンテージとして指定します。

dbc_max_pct と dbc_min_pct は同じ値に設定することも可能です。例えば、この 2 つのパラメータをともに 12 に設定すると、物理メモリのサイズに関係なく、常に 12% の物理メモリをバッファ・キャッシュに割り当てるようにカーネルが構成されます。

関連するパラメータ

dbc_max_pct には、dbc_min_pct 以上の値を指定する必要があります。dbc_min_pct に指定する必要最小限の値を算出する方法は、[dbc_min_pct](#) を参照してください。

動的バッファ・キャッシュを使用するには、[bufpages](#) と [nbuf](#) をともに 0 に設定する必要があります。

付加情報

[大容量記憶に関するパラメータの概要](#)

[ファイル・システムおよび大容量記憶に関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nbuf

nbuf は、システム全体におけるファイル・システム・バッファおよびキャッシュ・バッファのヘッダの数(システムの全バッファ数の上限を決定)を設定します。下記の「注記」を参照してください。

指定可能な値:

最小値	0 または 16
最大値	メモリ・サイズに依存。
デフォルト値	0

通常は、0 を指定してください(「説明」を参照)。詳細は、[パラメータ値の指定](#) を参照してください。

説明

注: このパラメータは以前のバージョンとの互換性のためのもので、通常は、動的バッファ・キャッシュの方が望ましいので、このパラメータは 0 に設定することが推奨されます。詳細は、[ファイル・システム・バッファに関するパラメータ](#) および [ファイル・システムに関する構成可能パラメータの概要](#) を参照してください。

このパラメータに 0 でない値を指定すると、その値がファイル・システム・バッファ・キャッシュに割り当てるバッファ・ヘッダの数として適用されます。この場合、各バッファには、4096 バイトのメモリが割り当てられます。ただし、この値が [bufpages](#) の値と衝突するときは、bufpages が優先して適用されます。

nbuf では、動的バッファ・キャッシュ用に予約する物理メモリの割合の上限をパーセンテージとして指定します。

なお、nbuf が 0 でない 16 未満の値やシステムがサポートする最大値を越える値の場合、あるいは bufpages の値と矛盾している場合には、その数が適切に増減され、ブート時に変更を知らせるメッセージが表示されます。

関連するパラメータ

nbuf は、bufpages と互に関連しています。その関係を以下に示します。

- `bufpages = 0` かつ `nbuf = 0` の場合: 動的バッファ・キャッシュを使用可能にします。
- `bufpages ≠ 0` かつ `nbuf = 0` の場合: システム・ブート時に、`bufpages×4 K` バイトのバッファ・プール・スペースを割り当てて、`bufpages÷2` 個のバッファ・ヘッダを作成します。
- `bufpages = 0` かつ `nbuf ≠ 0` の場合: ブート時に、`nbuf×2` ページのバッファ・プール・スペースを割り当てて、`nbuf` 個のヘッダを作成します。
- `bufpages ≠ 0` かつ `nbuf ≠ 0` の場合: ブート時に、`bufpages` ページのバッファ・プール・スペースを割り当てて、`nbuf` 個のバッファ・ヘッダを作成します。2つのパラメータの値が衝突しており、その両方を使ってシステムを構成できない場合は、`bufpages` の値が優先的に適用されます。

`bufpages` は、実際にバッファ・プールに割り当てるメモリを制御します。

付加情報

[大容量記憶に関するパラメータの概要](#)

[ファイル・システムおよび大容量記憶に関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

hfs_ra_per_disk

HFS ファイル・システムにおけるディスク・ドライブあたりの先読みデータ量を、KB 単位で設定します。

指定可能な値:

最小値	0
最大値	8192
デフォルト値	64

このパラメータには、整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

ディスク・ドライブからのデータ読み取り時に、システムは要求された以上のデータを余分に読み取ることができます。この「先読み」処理は、データがさらに必要になると予測して、要求される前にそれらのデータをシステム・バッファに読み込んで利用可能にしておくことにより、順次ディスク・アクセスの速度を向上しようというものです。このパラメータは、1つのディスク・ドライブあたりに許可される先読みデータ量を指定します。

先読み量の合計は、`hfs_ra_per_disk` に論理ボリューム内のドライブ数を乗じた値になります。ファイル・システムが論理ボリューム内にはない場合は、ドライブ数は事実上 1 になります。

単一の HFS ファイル・システムについてカーネルが未処理のまま先読みできるブロックの総数は、[hfs_max_ra_blocks](#) によって制約されます。

[hfs_ra_per_disk](#) および [hfs_max_ra_blocks](#) は、そのシステムの作業負荷上の特徴に応じて調整する必要があります。

詳細は、[hfs_max_ra_blocks](#) の説明の中で挙げられている例を参照してください。

関連するパラメータ

HFS での先読み操作:

- [hfs_ra_per_disk](#)
- [hfs_max_ra_blocks](#)

HFS での逆方向への先読み操作:

- [hfs_revra_per_disk](#)
- [hfs_max_ra_blocks](#)

VxFS での先読み操作:

- [vxfs_ra_per_disk](#)
- [vxfs_max_ra_kbytes](#)

付加情報

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

hfs_max_ra_blocks

単一の HFS ファイル・システムについて、カーネルが未処理のまま先読みできる最大ブロック数を設定します。

指定可能な値:

最小値	0
最大値	128
デフォルト値	8

このパラメータには、[整数値を直接指定するか、または整数式を指定してください。](#) 詳細は、[パラメータ値の指定](#) を参照してください。

説明

ディスク・ドライブからのデータ読み取り時に、システムは要求された以上のデータを余分に読み取ることができます。この「先読み」処理は、データがさらに必要になると予測して、要求される前にそれらのデータをシステム・バッファに読み込んで利用可能にしておくことにより、順次ディスク・アクセスの速度を向上しようというものです。このパラメータは、各 HFS ファイル・システムについてカーネルが先読みできるブロック数を制限します。この制限値は、個々の HFS ファイル・システムに適用されるものであり、システム全体に対する制限値の合計ではありません。

`hfs_max_ra_blocks` および `hfs_ra_per_disk` は、そのシステムの作業負荷上の特徴に応じて調整する必要があります。

注: 現在のディレクトリを含んでいるファイル・システムのブロック・サイズを確認するには、以下のコマンドを実行します。

`df -g` .

例:

1. ソフトウェア開発環境では、一般に I/O サイズは小～中程度で、多数のディスク・シークが実行されます。そのため、`hfs_max_ra_blocks` は 8～16 ブロックに、`hfs_ra_per_disk` は 32～64KB に設定してください。
2. MCAE アプリケーションの Out-of-core ソルバには重要な順次 I/O コンポーネントがあるため、`hfs_max_ra_blocks` は 64～128 ブロックに、`hfs_ra_per_disk` は 128～256KB に設定してください。

関連するパラメータ

HFS での先読み操作:

- [`hfs_ra_per_disk`](#)
- [`hfs_max_ra_blocks`](#)

HFS での逆方向への先読み操作:

- [`hfs_revra_per_disk`](#)
- [`hfs_max_revra_blocks`](#)

VxFS での先読み操作:

- [`vxfs_ra_per_disk`](#)
- [`vxfs_max_ra_kbytes`](#)

付加情報

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

hfs_revra_per_disk

逆方向への HFS 先読み処理でディスク・ドライブあたりに使用するメモリ量を、KB 単位で設定します。

指定可能な値:

最小値	0
最大値	8192
デフォルト値	64

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

ディスク・ドライブからのデータ読み取り時に、システムは要求された以上のデータを余分に読み取ることができます。この「先読み」処理は、データがさらに必要になると予測して、要求される前にそれらのデータをシステム・バッファに読み込んで利用可能にしておくことにより、順次ディスク・アクセスの速度を向上しようというものです。このパラメータは、逆方向への読み込み時に、1つのディスク・ドライブあたりに許可される先読みデータ量を指定します。

先読み量の合計は、hfs_revra_per_disk に論理ボリューム内のドライブ数を乗じた値になります。ファイル・システムが論理ボリューム内がない場合は、ドライブ数は事実上 1 になります。

単一の HFS ファイル・システムについてカーネルが逆方向に先読みできるブロックの総数は、hfs_max_revra_blocks によって制約されます。

このパラメータは、当社のフィールド・エンジニアが使用するためのものです。通常、一般の方は hfs_revra_per_disk パラメータをデフォルト値から変更しないでください。

例:

ファイル・システム・ブロック・サイズが小さいファイル・システム上で、逆方向への順次ファイル I/O が頻繁に実行される場合は、このパラメータの値を大きくする必要があります。このパラメータの値を大きくすると、バッファ・キャッシュ内でより多くのメモリが使用されるようになります。

ファイル・システム・ブロック・サイズが大きいファイル・システム上で、逆方向への順次ファイル I/O が少数しか実行されない場合は、このパラメータの値を小さくする必要があります。このパラメータの値を小さくすると、ファイル・スループット率が低下する可能性があります。

関連するパラメータ

HFS 先読み処理に関するパラメータ

- hfs_ra_per_disk
- hfs_max_ra_blocks

逆方向への HFS 先読み処理に関するパラメータ

- [hfs_revra_per_disk](#)
- [hfs_max_revra_blocks](#)

VxFS 先読み処理に関するパラメータ

- [vxfs_ra_per_disk](#)
- [vxfs_max_ra_kbytes](#)

付加情報

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

hfs_max_revra_blocks

単一の HFS ファイル・システムについて、カーネルが未処理のまま逆方向に先読みできる最大ブロック数を設定します。

指定可能な値:

最小値	0
最大値	128
デフォルト値	8

このパラメータには、**整数値**を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

ディスク・ドライブからのデータ読み取り時に、システムは要求された以上のデータを余分に読み取ることができます。この「先読み」処理は、データがさらに必要になると予測して、要求される前にそれらのデータをシステム・バッファに読み込んで利用可能にしておくことにより、順次ディスク・アクセスの速度を向上しようというものです。このパラメータは、各 HFS ファイル・システムに対する逆方向への読み込み時に、カーネルが未処理のまま先読みできるブロック数を制限します。この制限値は、個々の HFS ファイル・システムに適用されるものであり、システム全体に対する制限値の合計ではありません。

カーネル・パラメータ [hfs_revra_per_disk](#) は、当社のフィールド・エンジニアが使用するためのものです。一般の方はこのパラメータをデフォルト値から変更しないでください。

注: 現在のディレクトリを含んでいるファイル・システムのブロック・サイズを確認するには、以下のコマンドを実行します。

df-g .

目的

逆方向へのファイル読み込みを頻繁に実行することが明らかな場合は、このパラメータの値を大きくする必要があります。値を大きくしても性能があまり向上しない場合は、この値をデフォルト値に戻してください。この値を大きくすると、過度の先読み処理によりディスク競合や性能の低下が発生する可能性が高くなります。

関連するパラメータ

HFS 先読み処理に関するパラメータ

- [hfs_ra_per_disk](#)
- [hfs_max_ra_blocks](#)

逆方向への HFS 先読み処理に関するパラメータ

- [hfs_revra_per_disk](#)
- [hfs_max_revra_blocks](#)

VxFS 先読み処理に関するパラメータ

- [vxfs_ra_per_disk](#)
- [vxfs_max_ra_kbytes](#)

付加情報

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vxfs_ra_per_disk

VxFS ファイル・システムにおけるディスクあたりの先読みデータ量を、KB 単位で設定します。

指定可能な値:

最小値	0
最大値	8192
デフォルト値	1024

このパラメータには、整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

ディスク・ドライブからのデータ読み取り時に、システムは要求された以上のデータを余分に読み取ることができます。この「先読み」処理は、データがさらに必要になると予測して、要求される前にそれらのデータをシステム・バッファに読み込んで利用可能にしておくことにより、順次ディスク・アクセスの速度を向上しようというものです。このパラメータは、1つのディスク・ドライブあたりに許可される先読みデータ量を指定します。

先読み量の合計は、`vxfs_ra_per_disk` に論理ボリューム内のドライブ数を乗じた値になります。ファイル・システムが論理ボリューム内にはない場合は、ドライブ数は事実上 1 になります。

単一の VxFS ファイル・システムについてカーネルが先読みできるデータの総量は、`vxfs_max_ra_kbytes` によって制約されます。

詳細は、`vxfs_max_ra_kbytes` の説明の中で挙げられている例を参照してください。

関連するパラメータ

HFS 先読み処理に関するパラメータ

- [hfs_ra_per_disk](#)
- [hfs_max_ra_blocks](#)

逆方向への HFS 先読み処理に関するパラメータ

- [hfs_revra_per_disk](#)
- [hfs_max_revra_blocks](#)

VxFS 先読み処理に関するパラメータ

- [vxfs_ra_per_disk](#)
- [vxfs_max_ra_kbytes](#)

付加情報

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vxfs_max_ra_kbytes

単一の VxFS ファイル・システムについて、カーネルが未処理のまま先読みできる最大データ量を KB 単位で設定します。

指定可能な値:

最小値	0
最大値	65536
デフォルト値	1024

このパラメータには、整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

ディスク・ドライブからのデータ読み取り時に、システムは要求された以上のデータを余分に読み取ることができます。この「先読み」処理は、データがさらに必要になると予測して、要求される前にそれらのデータをシステム・バッファに読み込んで利用可能にしておくことにより、順次ディスク・アクセスの速度を向上しようというものです。このパラメータは、個々の VxFS ファイル・システムについてカーネルが未処理のまま先読みできるデータ量を制限します。この制限値は、個々の VxFS ファイル・システムに適用されるものであり、システム全体に対する制限値の合計ではありません。

このパラメータは `vxfs_ra_per_disk` と連携して、VxFS ファイル・システムの先読みデータ量を制御します。VxFS ファイル・システムにおける先読みデータ量の最大値は、以下の値のいずれか小さい方になります。

`vxfs_ra_per_disk × number_of_disks_in_logical_volume` (論理ボリューム内のディスク数)

または

`vxfs_max_ra_kbytes`

一般に、`vxfs_max_ra_kbytes` の値を大きくするほど、順次 I/O の性能は向上します。

例

Seagate Barracuda 9GB ドライブの 4 ウェイ・パラレル・ストライプを使用する場合は、`vxfs_max_ra_kbytes` を 1024 に、`vxfs_ra_per_disk` を 256 に設定すると、適正な性能が得られます。

関連するパラメータ

HFS 先読み処理に関するパラメータ

- [hfs_ra_per_disk](#)
- [hfs_max_ra_blocks](#)

逆方向への HFS 先読み処理に関するパラメータ

- [hfs_revra_per_disk](#)
- [hfs_max_revra_blocks](#)

VxFS 先読み処理に関するパラメータ

- [vxfs_ra_per_disk](#)
- [vxfs_max_ra_kbytes](#)

付加情報

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

scsi_maxphys

SCSI I/O サブシステムの最大レコード・サイズをバイト単位で設定します。

指定可能な値:

最小値	1048576 (1 MB)
最大値	16777215 (16MB - 1)
デフォルト値	1048576 (1 MB)

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

このパラメータは [st_large_recs](#) と組み合わせて使用され、論理レコードの分割および再結合の必要なしに、大きいサイズのテープ・レコードをサポートできるようにします。

raw I/O のデフォルトの物理レコード・サイズは 1MB です。テープ・ドライブへのアクセスは、通常 raw デバイスを介して提供されるため、この値はテープに書き込めるレコードの最大サイズに直接影響を与えます。SCSI の場合の最大値は (16MB - 1) です。DLT テープや DAT (DDS) テープを含めて、多くのテープ・ドライブでは、最大このサイズまでのレコードをサポートしています。

HP-UX の通常の動作では、MAXPHYS(256K) を超える要求はそれぞれ 256 KB ずつの論理テープ・レコード単位に分割されます。ただし、他のベンダーのオペレーティング・システム上で書き込まれたテープでは、より大きなレコード・サイズが使われて

いる場合があります。このようなテープを読み込む場合は、レコードが不正に切り捨てられて結合されるため、破損したデータがアプリケーションに渡される可能性があります。このパラメータを設定し、`st_large_recs` を有効にすると、SCSI テープ・レコードが論理レコード単位に分割されなくなります。この場合は、`scsi_maxphys` の値がレコード・サイズの上限值となります。`scsi_maxphys` を超えるサイズのレコードがあると、I/O は失敗します。

大部分のアプリケーションでは、このような大きいサイズのレコードを使用することはありません。このパラメータは、こうした機能を必要とする技術者の要求に応じて追加されたものです。

付加情報

[st_large_recs](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

scsi_max_qdepth

SCSI デバイスについて、待ち行列に入れることができる SCSI コマンドの最大数を設定します。

指定可能な値:

最小値	1
最大値	255
デフォルト値	8

このパラメータには、**整数値**を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは、システムのデフォルト値以上の深さの待ち行列をサポートできるデバイスについて、ドライバが同時に待ち行列に入れることができる I/O の数を制御します。有効な値は 1~255 です。ディスク・デバイスによっては、このコマンドで設定できる最大の待ち行列の深さをサポートしていないこともあります。ディスクが対応できる以上の待ち行列の深さをソフトウェアで設定すると、ディスク上で QUEUE FULL 状態が発生したときに、I/O が処理されないままになります。

このパラメータに `queue_depth` (待ち行列の深さ) を設定することは、以下のコマンドと同じ意味を持ちます。

```
scsictl -m queue_depth
```

ただし、`scsictl` コマンドを使用する場合は、マシンを次回再ブートするまでの間のみ待ち行列の深さが変更されます。再ブートすると、値はシステムのデフォルト値に戻されます。一方このパラメータを使用すると、システムのブート時に使用されるデフォルトの待ち行列の深さが変更されます。

このパラメータは、SCSI タグ待ち行列機構を使用するデバイス(主としてディスクや CD-ROM などのデバイス)にのみ適用されます。デフォルト値は 8 であり、単一の SCSI デバイスに対して最大 8 つのコマンドが待ち行列に入れられます。

このパラメータに過度に大きな値を設定すると、待ち行列に入れられたコマンドが処理されるまでに長時間かかってしまい、タイムアウトが発生する可能性があります。このパラメータの値を大きくした場合の実際の影響を予測することは極めて困難です。そのため、このパラメータは経験豊富なシステム管理者のみが変更するようにし、また値の変更は慎重に行ってください。

付加情報

- `scsictl(1M)`
- [構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

st_fail_overruns

このパラメータを設定すると、データ・オーバーランが発生した SCSI テープ読み込みはエラーとなります。

指定可能な値:

最小値	0
最大値	1
デフォルト値	0

このパラメータには、0 または 1 を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

特定の技術アプリケーションでは、実際のテープ・レコード・サイズよりも短いレコードを読み取った場合に、エラーを生成する必要があります。

例えば、1MB のレコード長で書き込まれたテープに対して、256KB 単位での読み込み要求が実行されることがあります。HP-UX の通常の動作では、これをエラーとはせずに 256KB の長さで転送します。残りの 768KB が破棄されたことを示すエラー・メッセージは返されません。このパラメータを有効にした場合は、読み取りコマンドから EFBIG エラーが返されるようになります。ただし、この場合も要求された 256KB のデータは転送されます。

各プログラムでは、`MT_GET_FAILOVR` および `MT_SET_FAILOVR` を指定して `ioctl(2)` を実行することにより、システム全体ではなく個々の LUN についてこの値を取得および設定することもできます。

付加情報

- `ioctl(2)`

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

st_large_recs

このパラメータを設定すると、大きいサイズの SCSI テープ・レコードがサポートされるようになります。

指定可能な値:

最小値	0
最大値	1
デフォルト値	0

このパラメータには、0 または 1 を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは `scsi_maxphys` と組み合わせて使用され、論理レコードの分割および再結合の必要なしに、大きいサイズのテープ・レコードをサポートできるようにします。

HP-UX の通常の動作では、MAXPHYS(256K) を超える要求はそれぞれ 256 KB ずつの論理テープ・レコード単位に分割されます。ただし、他のベンダーのオペレーティング・システム上で書き込まれたテープでは、より大きなレコード・サイズが使われている場合があります。このようなテープを読み込む場合は、レコードが不正に切り捨てられて結合されるため、破損したデータがアプリケーションに渡される可能性があります。このパラメータおよび `scsi_maxphys` を設定すると、SCSI テープ・レコードが論理レコード単位に分割されなくなります。この場合は、`scsi_maxphys` の値がレコード・サイズの上限值となります。`scsi_maxphys` を超えるサイズのレコードがあると、I/O は失敗します。

大部分のアプリケーションでは、このような大きいサイズのレコードを使用することはありません。このパラメータは、こうした機能を必要とする技術者の要求に応じて追加されたものです。

各プログラムでは、`MT_GET_LARGERECs` および `MT_SET_LARGERECs` を指定して `ioctl(2)` を実行することにより、システム全体ではなく個々の LUN について、`st_large_recs` 値を取得および設定することもできます。

付加情報

- `ioctl(2)`

[scsi_maxphys](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ファイルのオープンとロックに関する構成可能パラメータ

ファイルのオープンとロック

個々のプロセスは、1 つまたは複数のファイルをオープンして読み書きを行います。また、大規模なデータベース・アプリケーションのように、単一のプロセスが多数のファイルを同時にオープンすることもあります。

さらに、メールやデータベースなどのアプリケーションでは、1 つのファイルに対して複数のプロセスが同時にアクセスすることがよくあります。このような場合には、複数のプロセスが同じファイルを同時に変更するとファイルやデータが破損する可能性があるため、それらのプロセスのうち、ファイルの内容を変更しているプロセスがファイルをロックし、ほかのプロセスがそのファイルを変更できないようにします。ロックされたファイルは、ほかのプロセスがデータを安全に書き込める状態になった時点で解放されます。

ファイルのオープンとロックには、メモリおよびその他のシステム・リソースが必要になります。システム全体の性能を最適なレベルに維持するには、ほかのシステム・ニーズとのバランスを取りながら、これらのリソースを割り当てる必要があります。システム全体において、すべてのプロセスが行うファイル・ロックの合計数の最大許容値を指定するには、[nlocks](#) を使います。

制限値

1 人のユーザーが同時にオープンするファイルの総数には、論理制限値と物理制限値を設定できます。論理制限値を設定するには [maxfiles](#) を、物理制限値を設定するには [maxfiles_lim](#) を使います。システム全体で同時にオープンするファイルの最大許容数を指定するには、[nfile](#) を使います。

付加情報

[maxfiles](#)

[maxfiles_lim](#)

[nfile](#)

[nlocks](#)

[ninode](#)

[ファイル・システムに関するパラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

maxfiles

maxfiles は、1 つのプロセスが同時にオープンするファイル数に対する制限値を設定します。

指定可能な値:

最小値	30
最大値	60000
デフォルト値	60

このパラメータには、0 または 1 を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

maxfiles は、1 つのプロセスが同時にオープンするファイルの数に対するシステム・デフォルトの論理制限値を設定します。論理制限値は、プロセスから変更可能な制限値です。maxfiles 個以上のファイルをオープンする必要があるプロセスは、この値を増やすことができます。

非スーパーユーザー・プロセスでは、[maxfiles_lim](#) による物理制限値に達するまで、論理制限値を増やすことができます。

関連するパラメータ

maxfiles を論理制限値として適用させるには、このパラメータに [maxfiles_lim](#) よりも小さい値を指定する必要があります。また、maxfiles_lim を物理制限値として適用させるには、[nfile](#) および [ninode](#) による制限を超過しないように設定する必要があります。

付加情報

ファイルのオープンとロックに関するパラメータの概要は、[ファイルのオープンとロックに関する構成可能パラメータ](#) を参照してください。

ファイル・システムに関するカーネル・パラメータの概要は、[ファイル・システムに関するパラメータの概要](#) を参照してください。

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

maxfiles_lim

1 つのプロセスが同時にオープンするファイルの数に対する物理制限値を設定します。

指定可能な値:

最小値	30
最大値	nfile
デフォルト値	1024

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

maxfiles_lim は、1 つのプロセスが同時にオープンするファイルの数に対するシステム・デフォルトの物理制限値を設定します。非スーパーユーザー・プロセスが論理制限値 (maxfiles) を増やすときには、このパラメータによる物理制限値がその上限となります。

maxfiles_lim は、カーネルを再構築するか、または実行中のカーネル内で settune() を使用することにより設定できます。SAM および kmtune は settune() を使用します。このパラメータに対する動的な変更は、以下の条件にあてはまるプロセスを除き、システム内のすべての既存プロセスに影響を与えます。

1. 新しい制限値をすでに超えているプロセス
2. setrlimit() コールまたは ulimit コールにより、独自の制限値が設定されているプロセス

関連するパラメータ

maxfiles を論理制限値として適用させるには、maxfiles_lim よりも小さい値を指定する必要があります。また、maxfiles_lim を物理制限値として適用させるには、nfile および ninode による制限を超過しないように設定する必要があります。

付加情報

[ファイルのオープンとロックに関する構成可能パラメータ](#)

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nfile

システム上で同時にオープンするファイルの最大許容数を設定します。

指定可能な値:

最小値	14
最大値	メモリ・サイズに依存。
デフォルト値	$((16 * (NPROC + 16 + MAXUSERS)) / 10) + 32 + 2 * (NPTY + NSTRPTY + NSTRTEL)$

整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

nfile は、システム全体において同時にオープンするファイルの最大許容数を定義します。

ファイル記述子テーブルのスロット数は、このパラメータの値と同じになります。スロット数が不足するとシステムの処理能力が低下するので、このパラメータには、ある程度余裕を持たせた値を指定してください。このパラメータの値を増やしても、あまり多くのメモリは必要になりません。

関連するパラメータおよびシステム要素

nfile には、[nproc](#)、[maxusers](#)、[npty](#)、および [nstrpty](#) の各パラメータの組み合わせによって決定されるユーザーおよびプロセスの最大許容数に十分に見合った値を指定する必要があります。

各プロセスは、少なくとも 3 つのファイル記述子を使用します (標準入力、標準出力、および標準エラーの 3 つ)。

各プロセスには、出力側と入力側に 1 つずつ、合わせて 2 つのパイプが関連付けられます。各パイプには、pty が 1 つずつ必要です。ストリーム・パイプには、さらにストリーム pty が使用されますが、これらの pty の数は nstrpty によって制限されます。

付加情報

ファイルのオープンとロックに関するパラメータの概要は、[ファイルのオープンとロックに関する構成可能パラメータ](#) 参照してください。

ファイル・システムに関するカーネル・パラメータの概要は、[ファイル・システムに関するパラメータの概要](#) を参照してください。

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nflocks

nflocks は、システム全体において、すべてのプロセスが行うファイル・ロックの合計数の最大許容値を定義します。

指定可能な値:

最小値	2
最大値	メモリ・サイズに依存。
デフォルト値	200

整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

nflocks は、システム全体において、利用可能なファイル/レコード・ロックの最大許容数を指定します。このパラメータに指定する値を決定するときには、1 つのファイルに複数のロックが適用される可能性があること、さらに、lockf() を使用するデータベースには非常に多数のロックが必要になる可能性があることを考慮してください。

ファイルのオープンとロックを行うと、メモリおよびその他のシステム・リソースが消費されます。システム全体の性能を最適なレベルに維持するには、システムの他の要求とのバランスを取りながら、これらのリソースを割り当てる必要があります。最適なバランスを設定するには、各システムで使用されるアプリケーションの種類、同時に実行される可能性があるアプリケーションの数とタイプ、システム上のローカル・ユーザーおよびリモート・ユーザーの数など、さまざまな要素を考慮しなければなりません。特に、規模の大きいシステムでは、これらの要素が大きく変化するため、入念な検討が必要です。

関連するパラメータ

なし。

付加情報

ファイルのオープンとロックに関するパラメータの概要は、[ファイルのオープンとロックに関する構成可能パラメータ](#) 参照してください。

ファイル・システムに関するカーネル・パラメータの概要は、[ファイル・システムに関するパラメータの概要](#) を参照してください。

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ninode

メモリ内で存在可能な、HFS ファイルシステムのオープン i ノードの最大数

指定可能な値:

最小値	14
最大値	メモリ・サイズに依存
デフォルト値	$nproc+48+maxusers+(2*npty)$

整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

ninode は、i ノード・テーブル内のスロットの数を定義します。メモリ内で同時にオープンできる i ノードの数は、この値が上限となります。i ノード・テーブルは、キャッシュ・メモリとして使用されます。効率性を向上するために、最近オープンされた Ninode 番目までの i ノードは、メイン・メモリ内に保持されます。このテーブルは、ハッシュされます。

ファイルを独自にオープンすると、それぞれ 1 つの i ノードがオープンされ、そのファイルに関連付けられます。したがって、独自にオープンするファイルの数が多い場合は、それに十分見合う値を ninode に指定する必要があります。

関連するパラメータ

ninode のデフォルト値は、[nproc](#)、[maxusers](#)、および [npty](#) の各パラメータの値の組み合わせに基づいて決定されます。

付加情報

ファイルのオープンとロックに関するパラメータの概要は、[ファイルのオープンとロックに関する構成可能パラメータ](#) を参照してください。

ファイル・システムに関するカーネル・パラメータの概要は、[ファイル・システムに関するパラメータの概要](#) を参照してください。

[ファイル・システムに関するパラメータの概要](#)

[ファイル・システムに関するパラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

fs_async

fs_async は、ディスクへのデータ構造の書き込みを同期のみで行うか、それとも同期と非同期の両方で行うかを指定します。

指定可能な値:

最小値 0 (ディスクへの書き込みを同期のみで行います)

最大値 1 (ディスクへの非同期書き込みを可能にします)

デフォルト値 0

0 または 1 のいずれかを指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

fs_async は、ディスクにファイル・システム形式のデータ構造を書き込むときに、非同期書き込みを行えるようにするかどうかを指定します。fs_async の値を省略した場合は、同期書き込みのみが行われます。

同期書き込みには、ファイル・システム上のファイル・システム形式のデータ構造の更新中にシステム・クラッシュが発生した場合に、ファイル・システムの完全性を復元しやすいという利点があります。

非同期書き込みを有効に設定した場合でも、HP-UX ファイル・システムの NFS クラスタ環境に対するセマンティクスが保持されます。さらに、カーネル内に非同期書き込み機能が組み込まれている場合でも、open() を O_SYNC フラグ (同期書き込み) 付きで呼び出してオープンしたファイルに対しては、継続して同期書き込みが行われます。

ディスクへの非同期書き込みには、ファイル・システムの性能を大幅に向上できるという利点があります。ただし、システム・クラッシュが発生すると、それまでディスクに非同期で書き込まれていたファイル・システム・データ構造が復元不能になる可能性

があります。同期書き込みと非同期書き込みの使い分けは、このヘルプ・ページに用意されているチュートリアルを参照してください。

関連するパラメータ

なし。

付加情報

[大容量記憶に関するパラメータの概要](#)

[構成可能なファイル・システムのパラメータ](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

チュートリアル: 同期書き込みと非同期書き込みについて

書き込み可能でオープンしたファイルにデータを書き込むときには、データがいったんバッファに蓄積され、定期的にディスクに書き込まれます。end-of-file 条件によりファイルをクローズするときには、それまでにバッファ内に蓄積されたデータがディスクに書き込まれ、新しいファイル・サイズとブロック・ポインタ情報を反映するように i ノードが更新され、さらに、ファイル・システムの空きディスク・ブロック・リストが更新されます。ディスクへの書き込み中にシステム・クラッシュや電源異常が発生してもファイル・システムが壊れる危険性が最小限になるようなシーケンスで更新処理を実行すれば、ファイル・システムの完全性を最大限に保護することができます。このようなシーケンスで行われる更新処理を同期書き込みと呼びます。

HP-UX ファイル・システムでは、空きスペース・リスト、ブロック、i ノードなどのファイル構成要素をディスク・デバイス上のさまざまな場所にランダムに保存します。このため、特定のシーケンスでファイル情報ブロックを書き込む場合には、ディスク上の目的の位置に移動してから書き込み処理を行うことになり、その分、時間が必要になります。このシーケンスの途中で電源異常やシステム・クラッシュが発生すると、1 つまたは複数のブロックが更新されずに残され、そのためにファイル・システムの整合性が失われる可能性があります。このような不整合を修復するには、fsck コマンドを使用します。詳細は、HP-UX Reference の fsck(1M) を参照してください。

fs_async カーネル・パラメータによる非同期書き込みでは、安全性よりも速度を重視してディスク上のファイル・システム情報を更新するので、書き込み先の検索とその場所への移動による遅延が少なくなります。しかし、非同期書き込みの実行中にシステム・クラッシュが発生した場合、fsck により自動的に修復できない不整合がファイル・システムに発生する可能性は同期書き込みの場合よりもかなり高くなります。

クラッシュ後の復元

同期書き込みだけを使用しているときには、ディレクトリ、ファイル i ノード、空きスペース・リストなどに対する更新は、いずれも fsck による追跡が可能なシーケンスで行われます。このシーケンスでディスク・ブロックを更新しているときにクラッシュが発生した場合は、fsck によって、クラッシュの発生場所を特定し、欠落している更新情報を修復することができます。しかも、多くの場合は、システム管理者が fsck を対話式に操作しなくても、データが自動的に修復されます。

一方、fs_async によって非同期書き込みを使用可能に設定しているときにクラッシュが発生した場合には、fsck は、書き込みのシーケンスを認識できないため、データを自動修復できなくなる可能性が高くなります。自動修復ができない場合は、システム管理者が fsck を対話式に操作して、システム情報の不整合を復元するとともに、ディレクトリや i ノード・エントリを修復しなければなりません。

非同期書き込みを使用する理由

同期書き込みでは、データを書き込んだファイルをクローズするときにディスク・ブロックの書き込みと更新が完了するまで待機しなければなりません。このため、ファイルとディレクトリに対して書き込みとクローズを頻繁に行うプログラムやアプリケーションの場合は、性能が低下します。一方、非同期書き込みを可能に設定すると、このような遅延が大幅に減り、その分性能が向上します。ただし、ディスク I/O よりも CPU に対する負荷の方が大きいアプリケーションの場合、非同期書き込みを使用しても性能はほとんど向上しません。

非同期書き込みを使用する理由

非同期書き込みによってシステムの性能を効果的に向上できるのは、以下のような場合です。

- 電源異常が発生する可能性が低い場合 (電源や無停電電源の信頼性が非常に高い場合)。
- データのセキュリティを強化するための予防策が講じられている場合 (ファイル・システムのバックアップや冗長性が十分に確保されている場合) か、または、システム・クラッシュに対するデータの保護よりもシステムの性能を重視している場合。
- ユーザーがディスク・ファイルおよびディレクトリのオープン、書き込み、クローズを頻繁に行うアプリケーションを使用する場合。
- 非同期書き込みによって、そのリスクを十分に補う性能向上が見込める場合。

非同期書き込みを使用可能にするには、`fs_async` の値をデフォルト値の 0 から 1 に変更します。

vx_ncsize

`vx_ncsize` は、VxFS ファイル・システムのディレクトリ・パス名キャッシュ用に予約するメモリ・スペースをバイト単位で指定します。

指定可能な値:

最小値	0
最大値	なし
デフォルト値	1024

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

VxFS ファイル・システムでは、ファイル・システム内で最近アクセスされたディレクトリに関するディレクトリ・パス名情報を名前キャッシュに格納します。最近アクセスされたディレクトリに関するパス名情報は名前キャッシュから取得できるので、頻繁に使用されるディレクトリを検索するときには、その都度ディスクに直接アクセスしてディレクトリ・ツリーを下層にたどる必要がなくなり、ディレクトリおよびその内容にすばやくアクセスできるようになります。名前キャッシュをこのように使用することにより、オーバーヘッドを大幅に軽減できます。特定のディレクトリやディレクトリ・パスに繰り返しアクセスする大規模アプリケーション (データベースなど) の場合には、特に高い効果が得られます。

vx_ncsize は、VxFS ファイル・システム・マネージャが名前キャッシュとして使用するスペースの量をバイト単位で指定します。このパラメータには、標準的な構成の HP-UX システムの大半において必要なスペースを十分に確保できるデフォルト値が用意されていますが、VxFS ディスク I/O の頻度が非常に高いアプリケーションを使用する大規模なシステムの場合は、デフォルト値よりも大きい値を設定することによって性能を向上できることがあります。ただし、このパラメータの値を増やすことでどの程度性能を向上できるかは、アプリケーションで使用されるディレクトリ・パスの多様性によって異なり、さらに、VxFS ファイル・システムとの対話動作中に消費される総プロセス時間のパーセンテージによって異なります。

付加情報

[ファイル・システムに関するカーネル・パラメータの概要](#)

[ファイル・システムに関するカーネル・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

LVM の動作の概要

Logical Volume Manager (LVM) は、ファイル・システムおよびディスク記憶スペースの論理ボリュームを管理するためのサブシステムです。論理ボリュームは、物理ディスクのサイズとは無関係に作成できます。1つのディスクまたはディスク・アレイの中に複数の論理ボリュームを作成することも、また、複数のディスクまたはディスク・アレイから1つの論理ボリュームを作成することもできます。複数のディスクまたはアレイを使用する場合、論理ボリュームの境界を物理ディスクの境界と一致させる必要はありません。

論理ボリュームの管理は、カーネルではなく、Logical Volume Manager によって行われます。ただし、システム上の各ボリューム・グループのデータ構造はカーネルに格納されます。さらに、システム上のボリューム・グループの数に十分見合ったサイズのスペースを LVM データ構造用に予約する必要があります。このスペースを構成するには、カーネル構成パラメータの maxvgs を使います。もう1つの no_lvm_disks パラメータは、ブート時にシステム上に論理ボリュームが存在しないことをカーネルに通知するためのパラメータです。これにより、システムがブート時に論理ボリュームの識別とアクティブ化を行う必要がなくなります。

ボリュームおよびグループの境界

論理ボリューム・グループは、1つまたは複数の論理ボリュームで構成されます。ボリューム・グループ内の論理ボリューム境界は、ディスク上の任意の位置に構成できます。ただし、複数のボリューム・グループ間で同じディスク・デバイスを共有することはできません。データ保護用の RAID (独立した複数のディスクからなる冗長アレイ) として構成されたディスク・アレイは、単一のディスク・デバイスとして扱われるため、複数のボリューム・グループの間で共有することはできません。一方、RAID として構成されていないアレイ内の個々のディスクは、個別のデバイスとして扱われるため、管理者が任意のボリューム・グループに必要な応じて割り当てることができます。

付加情報

[LVM パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

Logical Volume Manager に関する構成可能パラメータ

構成可能カーネル・パラメータには、カーネルと Logical Volume Manager に関するパラメータが 2 つあります。

maxvgs

システム上の Logical Volume Manager によって構成されるボリューム・グループの最大数を指定します。

no_lvm_disks

システム上に論理ボリュームが存在しないことをカーネルに通知するためのフラグです。このフラグが設定されている場合は、すべてのファイル・システムおよびその境界が、システムの物理ディスクおよび境界と一致します。ただし、ディスク上にパーティションが存在する場合、またはディスクの一部がスワップ用やファイル・システム以外の目的のために予約されている場合は、この限りではありません。

付加情報

[LVM の動作の概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

maxvgs

maxvgs は、システム上のボリューム・グループの最大数を指定します。

指定可能な値:

最小値	1
最大値	256
デフォルト値	10

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

maxvgs は、システム上のボリューム・グループの最大数を指定します。システム上の各論理ボリューム・グループごとに、データ構造のセットがカーネル内に作成されます。このパラメータを、システム上のボリューム・グループ数と一致するように設定すると、システムからの実際の要求に合うデータ構造を作成するだけでよいので、カーネルの記憶スペースを節約できます。

maxvgs は、デフォルトで最大 10 個のボリューム・グループ用に設定されます。許容されるボリューム・グループ数を変更するには、maxvgs を変更して新しい最大値を反映させてください。

関連するパラメータ

なし

付加情報

[LVM の動作の概要](#)

[Logical Volume Manager \(LVM\) に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

no_lvm_disks

no_lvm_disks は、システム上に論理ボリューム・グループが存在しないことをカーネルに通知します(ワークステーションのみ)。

指定可能な値:

最小値	0 (LVM ディスクがないかチェックする場合)
最大値	1 (システムに LVM ディスクがない場合)
デフォルト値	0

このパラメータには、整数値 0 または 1 を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

デフォルトでは、システムは構成済みのルート・ディスク、スワップ・ディスク、ダンプ・ディスク上にそれぞれ LVM データ構造がないかどうかをブート時にチェックします。システム上に LVM ディスクが存在しない場合は、no_lvm_disks を 1 に設定しておく、LVM データ構造のチェックが行われなためブート・プロセスがスピードアップします。

注記

LVM が使用されているシステムで、このパラメータを 0 でない値に設定すると、カーネル・パニックの原因となります。これは、カーネルが通常のブート・プロセス中にシステム上にある論理ボリュームについて必要な情報を取得できないためです。

関連するパラメータ

なし

付加情報

[LVM の動作の概要](#)

[Logical Volume Manager \(LVM\) に関する構成可能パラメータの一覧](#)
[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

メモリ・ページングに関するパラメータの概要

メモリ・ページングに関する構成可能カーネル・パラメータでは、仮想メモリ (スワップ・スペース) に適用する動作規則と制限を設定できます。これらのパラメータは、以下のカテゴリに分類できます。

システム・スワップの総量

システム全体において割り当て可能なスワップ・スペースの最大サイズ。このカテゴリには、[maxswapchunks](#) パラメータと [swchunk](#) パラメータがあります。

デバイス・スワップ

ハード・ディスク・デバイス上で割り当てられるスワップ・スペース。このカテゴリには、[nswapdev](#) パラメータがあります。

ファイル・システム・スワップ

マウントされているファイル・システム上で割り当てられるスワップ・スペース。このカテゴリには、[allocate_fs_swapmap](#) パラメータと [nswapfs](#) パラメータがあります。

擬似スワップ

インストールされている RAM を擬似スワップとして使用することにより、ディスク・デバイス上のスワップ・スペースに制限されずに仮想メモリ・スペースを割り当てることができます。このカテゴリには、[swapmem_on](#) パラメータがあります。

可変ページ・サイズ

特定のアプリケーションでのスワップ処理がより効率的になるように仮想メモリ・ページのサイズを変更することができます。このカテゴリには、[vps_ceiling](#)、[vps_chattr_ceiling](#)、および [vps_pagesize](#) の各パラメータがあります。

システム・スワップの総量

システム全体に存在するスワップ・スペースの総量 (デバイス・スワップ・スペース、ファイル・システム・スワップ・スペース、およびリモート NFS スワップ・スペースの合計値) の最大許容値を決定する構成可能カーネル・パラメータには、[maxswapchunks](#) と [swchunk](#) の 2 つがあります。maxswapchunks は、システム全体において作成と割り当ての一方または両方が可能なスワップ・チャンクの数指定します。swchunk は、複数のデバイスまたは複数のファイル・システムにまたがって作成される各チャンクのサイズを決定します。このパラメータを適切に設定するには、カーネルの動作とシステム内部に関するきわめて高度な知識が必要です。詳しい知識がない場合は、swchunk をデフォルト以外の値に変更しないでください。

デバイス・スワップ

デバイスをシステムに接続して構成するとき、ディスクの一部をデバイス・スワップ用に予約することができます。デバイス・スワップに関連する構成可能カーネル・パラメータは、nswapdev の 1 つだけです。このパラメータは、デバイス・スワップ用に

スペースを予約するデバイスの数を指定します。データ構造記憶スペースが、カーネル内で `nswapdev` 個のデバイス用に予約されます。

- `nswapdev` の値がスワップ・デバイスの実際の数よりも大きい場合は、実際には存在しないデバイスにも少量のメモリ・スペース (デバイスあたり 50 バイト未満) が割り当てられることになります。このスペースは、他の用途には使用できません。
- `nswapdev` の値が利用可能なスワップ・デバイスの数よりも小さい場合は、`nswapdev` 個のデバイスに対してのみデータ構造記憶スペースが割り当てられるため、この個数を超える分のデバイスにはアクセスできなくなります。

ファイル・システム・スワップ

スワップ・スペースは、個々のデバイス上に作成できるだけでなく、マウント済みのファイル・システム上に作成することもできます。ファイル・システム・スワップでは、データをオペレーティング・システムとディスク・デバイス間で直接転送するのではなく、オープンされているファイルに対してデータを読み書きします。このため、ファイル・システム・スワップの動作は、デバイス・スワップよりも遅くなります。システム全体において、ローカルにマウントされているファイル・システムのうち、ファイル・システム・スワップをサポートするように構成できるファイル・システムの数を実際の数に一致するように定義するには、`nswapfs` を使います (このパラメータは、デバイス・スワップにおける `nswapdev` パラメータに相当します)。`nswapdev` の場合と同様、`nswapfs` は、常時マウントされており、ファイル・システム・スワップに使用したいファイル・システムの実際の数に一致するように設定する必要があります (データ構造の記憶には、`nswapfs` に 300 バイトを掛けた分のカーネル・スペースが必要です)。ファイル・システム・スワップに使用できるのは、ハード・ディスク読み込み/書き込みファイル・システムだけです。

ファイル・システム・スワップのパラメータとして、さらに `allocate_fs_swapmap` パラメータが用意されています。通常は、`malloc()` によってスペースが割り当てられるまで待機してからスワップ・スペースが割り当てられますが、このパラメータを使うと、`swapon()` が呼び出された時点でスワップ・スペースを割り当てることができます。スペースを事前に割り当てておけば、`malloc()` が呼び出されたときにファイル・システム・スワップ・スペースが無条件で利用可能になります (事前に割り当てていない場合は、特定の条件下で、ファイル・システム・フルのエラーが発生する可能性があります)。`swapon()` の呼び出し時にファイル・システム・スワップ・スペースを割り当てるように構成すると、可用性が向上します。ただし、リソースを予約したプロセスが実際にはスペースを使用していない場合でも、他のプロセスがそのリソースを使用することはできなくなります。

擬似スワップ

メモリの割り当ては、通常は、仮想メモリの可用性 (ディスクおよびファイル・システム上で利用可能なスワップ・スペース) に基づいて行われます。利用可能 (スワップ可能) なメモリの総量は、スワップ・スペースの総量と同じになります。しかし、大量の RAM をインストールした大規模なシステムの場合は、これが効率性の低下を招くことがあります。

例えば、200 MB の RAM が搭載されているシステムのルート・ディスクに 1 GB のスワップ・スペースがある場合を考えてみましょう。このようなシステムがシングル・ユーザー・モードで動作しており、なおかつカーネルが利用可能なシステム RAM の 10% 以内しか使用していない場合に、ルート・ディスクだけを 1 GB のメモリ・スペースにマウントするように制限するのは適切ではありません。擬似スワップを可能に設定すれば、RAM のうち、使用されていない領域をスワップ・システム用に割り当てることが可能になり、規模と負荷のより大きいプロセスを実行できるようになります。ただし、16 MB の RAM が搭載されており、100 MB または 200 MB のスワップ・スペースがルート・デバイス上に作成されているワークステーションの場合は、擬似スワップを可能に設定しても、ほとんど効果が得られません。

システム RAM 内における擬似スワップの割り当てを使用可能/使用不可能にするには、`swapmem_on` を使います。

付加情報

[ページング・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

メモリ・ページングに関する構成可能パラメータ

allocate_fs_swapmap

swapon()が呼び出されたときにファイル・システム・スワップ・スペースを割り当てることができるようにするかどうかを指定します (malloc()が呼び出されたときにスワップ・スペースを割り当ててわけではありません)。この割り当てを可能に設定すると、スワップ・スペースが不足する可能性が少なくなります。この割り当てを可能に設定するのは、主に、高可用性が重視される場合です。

maxswapchunks

システム全体において構成可能なデバイス・スワップ・スペースとファイル・システム・スワップ・スペースの合計量の最大値を決定します。このパラメータの値は、swchunk にシステム・ブロック・サイズの値 (DEV_BSIZE) を掛けた値で乗算されます。

nswapdev

システム全体においてデバイス・スワップに使用するデバイスの最大許容数。実際のシステム構成に一致する値を設定してください。

nswapfs

システム全体において、マウントされるファイル・システムのうち、ファイル・システム・スワップに使用可能なファイル・システムの最大数。実際のシステム構成に一致する値を設定してください。

page_text_to_local

クラスタ・クライアント上で、プログラムのテキスト・セグメントをローカル・スワップ・デバイスにスワップするかどうかを設定します。有効にした場合は、メモリに新しい内容をロードするためのロード時間は増加しますが (テキストはいったん破棄されてサーバーから再ロードされるのではなく、ローカル・ディスクに書き込まれます)、サーバーからクライアントへのネットワーク・トラフィックは減少します。

swapmem_on

マウントされているリモート NFS ファイル・システムへのスワップを使用可能/使用不可能にします。このパラメータは、クラスタ・クライアントで、NFS によってマウントされたファイル・システムへのスワップを行うときに使用されます。

swchunk

マウントされているリモート NFS ファイル・システムへのスワップを使用可能/使用不可能にします。このパラメータは、クラスタ・クライアントで、NFS によってマウントされたファイル・システムへのスワップを行うときに使用されます。

可変ページ・サイズに関するパラメータ

vps_ceiling

ユーザーがページ・サイズを指定しなかった場合にシステムが自動的に選択するページ・サイズの最大値を定義します。

vps_chatr_ceiling

ユーザーがプログラム内で chatr コマンドを使って指定できるページ・サイズの最大値を定義します。

vps_pagesize

ページ・サイズが chatr を使って指定されていない場合に適用するデフォルトのユーザー・ページ・サイズの最小値を定義します。

付加情報

[メモリ・ページング・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

allocate_fs_swapmap

allocate_fs_swapmap により、カーネルはファイル・システム・スワップで使用するための十分なカーネル・データ構造をブート時に前もって割り当てます。

指定可能な値:

最小値	0 (必要に応じてスワップ・データ構造を割り当てます)
最大値	1 (必要なカーネル・データ構造を前もって割り当てます)
デフォルト値	0

このパラメータには、整数値 0 または 1 を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

allocate_fs_swapmap では、ファイル・システム・スワップ用のカーネル・データ構造を必要な場合だけ割り当てるか、または前もって予約しておくかを指定します(ただしデバイス・スワップには適用されません)。有効な値は以下の 2 つだけです。

allocate_fs_swapmap = 0

(デフォルト値) システムは、必要に応じてデータ構造を割り当てて、システム・メモリを確保します。状況によっては、ファイル・システムにスワップ用の十分なスペースがあっても、システムに使用可能なデータ構造が不足しているために、システムがスワップ要求を拒否する場合があります。

allocate_fs_swapmap = 1

システムは、swapon() システム・コールまたは swapon コマンドによって指定されているファイル・システム・スワップの最大スペースに見合う十分なデータ構造を割り当てます。これにより、ファイル・システムに使用可能なスワップ・スペースがある

限り、メモリ内のスペースはスワップ要求に対応できます。このモードは、高可用性システムで最も頻繁に使用されます。これは、高可用性システムでは、リソースを必要になる前に予約しておくことによるシステム効率の低下よりも、リソースが使用不可能なために発生するプロセスの障害を防ぐことを重視するためです。

関連するパラメータ

なし

付加情報

[メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

maxswapchunks

システム上で構成可能なスワップ・スペースの最大容量を設定します。

指定可能な値:

最小値	1
最大値	16384
デフォルト値	256

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

maxswapchunks は、システム上で構成可能なスワップ・スペースの最大容量を指定します。スワップ・スペースの最大容量の上限は、以下の手順で算出します。

- 各ディスク・ブロックには DEV_BSIZE (1024) バイトが含まれます。DEV_BSIZE は、システム全体の大容量記憶ブロック・サイズであり、構成は不可能です。
- スワップ・スペースはチャンク単位でデバイスごとに割り当てられます。各チャンクは、swchunk のブロックを含みません。swchunk に適切な値を設定するには、システム内部に関するきわめて高度な知識が必要です。詳しい知識がない場合は、swchunk の値を標準のデフォルト値から変更しないでください。
- システム全体におけるスワップ・スペースの最大チャンク数は maxswapchunks で指定されています。
- スワップ・スペースの最大バイト数は、以下の式で算出します。

$\text{maxswapchunks} \times \text{swchunk} \times \text{DEV_BSIZE}$

例えば、swchunk のデフォルト値(2048)と maxswapchunks のデフォルト値(256)を使用し、DEV_BSIZE が 1024 バイトとすると、構成可能な全スワップ・スペースは 537 MB となります。

値を指定する

システムのディスク・デバイス上で使用可能なスワップ・スペースの容量は、ファイル/etc/fstab の内容によって決定され、カーネル構成の影響は受けません。

スタンドアロン・システムまたはローカル・スワップ・スペースを持つクラスター・クライアントの場合、maxswapchunks には、予測されるすべてのスワップに見合う十分なスワップ・スペース容量をサポートする値に設定することが必要です。このパラメータを十分な大きさの値に設定して、カーネルを再構成する必要がないようにしてください。

サーバー・ノードの場合、このパラメータには、サーバーのローカル・スワップに必要なスペースだけでなく、スワップを使用する各クライアント・ノードにも十分なスワップが確保される値に設定してください。少なくとも、各クライアントが使用するメモリ量と同等のスワップ・スペースを割り当ててください。

関連するパラメータ

構成可能なスワップ・スペースは以下の式により定義されます。

$$\text{maxswapchunks} \times \text{swchunk} \times 1024 \text{ バイト}$$

システム側が使用する最大スワップ・スペースは 1 テラバイト (1.09 兆バイト) です。

付加情報

[メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nswapdev

システム上で構成可能なスワップ・スペースの最大容量を設定します。

指定可能な値:

最小値	1
最大値	25
デフォルト値	10

デバイス・スワップ用に構成されている物理ディスク・デバイスの数を整数で指定します(最大 25)。このパラメータには整数しか指定できません(数式が返す値を指定しても正しく動作しません)。

詳細は、[パラメータ値の指定](#) を参照してください。

説明

nswapdev は、デバイス・スワップに使用できるデバイスの数を指定します。

カーネルは、システム・スワップ領域用に予約されている物理デバイスの数だけデバイス・スワップをサポートするのに十分な内部データ構造を、システムのブート時に作成します。指定した値が使用可能なデバイス数を上回っている場合、余分なデータ構造スペースは使用されず、浪費されるメモリもわずかで済みます(1つの構造当たり 50 バイト未満)。指定した値が使用可能なデバイス数を下回っている場合、カーネル内のデータ構造不足のため、一部のデバイスをスワップに使用できなくなります。

関連するパラメータ

なし

付加情報

[メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nswapfs

nswapfs は、ファイル・システム・スワップに使用可能なファイル・システムの最大数を指定します。

指定可能な値:

最小値	1
最大値	25
デフォルト値	10

ファイル・システム・スワップに使用できるファイル・システムの数を整数で指定します(最大 25)。

詳細は、[パラメータ値の指定](#) を参照してください。

説明

nswapfs は、ファイル・システム・スワップに使用できるファイル・システムの最大数を指定します。

カーネルは、指定されているファイル・システムの数だけファイル・システム・スワップをサポートするのに十分な内部データ構造(1つの構造当たり約 300 バイト)を、システムのブート時に作成します。指定した値が使用可能なファイル・システム数を上回っている場合、余分なデータ構造スペースは使用されないため、その分のメモリが無駄になります。指定した値が使用可能なフ

ファイル・システム数を下回っている場合、データ構造不足のため、一部のファイル・システムをスワップに使用できなくなります。

関連するパラメータ

なし

付加情報

[メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

page_text_to_local

NFS クラスタ・クライアント上で、プログラムのテキスト・セグメントをローカル・スワップ・デバイスにスワップするかどうかを設定します。

指定可能な値:

最小値	0 (スタンドアロン・システム、またはファイル・システム・サーバーを使用するクライアントの場合)
最大値	1 (クライアントのローカル・スワップを使用)
デフォルト値	1 (クライアントのローカル・スワップを使用)

このパラメータには、整数値 0 または 1 を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

通常プログラム内には、以下の 3 つのセグメントが含まれています。

テキスト・セグメント

動作中不変な、プログラムの実行可能コード部分

データ・セグメント

配列およびその他の固定データ構造体

DSS セグメント

動的記憶領域、スタック・スペースなど

不要なネットワーク・トラフィックをできる限り抑えるため、ローカルなスワップ・デバイスを持たない NFS クラスタ・クライアントでは、メモリをスワップする必要が生じた時点、プログラムのテキスト・セグメントを破棄することにより、他のプログラムやアプリケーション用のスペースを確保します。このようにテキスト・セグメントを破棄するのは、ローカル・ディスクを使用

できない場合にサーバー上のスワップ・スペースにいったんスワップして、(元のプログラム・ファイル内にも存在している) そのデータを後から再取得することは、サーバーのディスク・スペースの無駄であり、またネットワークのデータ・トラフィックも増加するためです。

ただし、クライアント・マシンに接続されたローカル・ディスク・デバイス上で十分なスワップ・スペースを確保できる場合は、テキスト・セグメントをローカル・スワップに書き出しておき、このデータを後から取り出す方が効率的です。この方法では、テキスト・セグメントをサーバーに転送したり、サーバーから再取得したりする必要がないため、クラスタの性能が向上します(向上の度合いは、使用しているアプリケーションやプログラムにより変動します)。ローカル・スワップをこのように使用するには、使用可能なスワップ・スペースが、システム上で同時に実行される可能性があるすべてのプロセスで必要とされる最大の総スワップ・スペースよりも大きくなければなりません。スペースが不十分な場合は、スペース競合が生じた時点で、システムのメモリ割り当てエラーが発生します。

`page_text_to_local` は、ネットワーク・トラフィックを低減させるために、テキスト・セグメントを破棄するか、またはローカル・デバイスにスワップするかを決定するための構成可能カーネル・パラメータです。

0. ローカルなクライアント・スワップ・デバイスを使用しません。クライアントがローカルなクライアント・スワップ・デバイスを持っていない場合、またはテキスト・スワップを完全にサポートできるだけのスペースがない場合に指定してください。メモリ・スペースが必要になった時点でテキスト・セグメントは破棄され、再度実行可能になった時点でサーバーから元のファイルが再取得されます。
1. 他の用途でメモリ・スペースが必要になった時点で、テキスト・ページはローカルなスワップ・デバイスにスワップされ、セグメントが必要になった時点でスワップ・デバイスから取り出されます。通常この方法をとると、クライアントの性能が向上し、クラスタのネットワーク・データ・トラフィック量も減少します。この値を指定するには、ローカル・スワップが有効でなければならず、さらにクライアントのローカル・ディスク上のデバイス・スワップ・スペースが、クライアント上で同時に実行される可能性があるすべてのプロセスで必要とされる最大の仮想メモリ・サイズよりも大きくなければなりません。スペースが不十分な場合は、メモリ不足によりプロセスが異常終了する可能性があります。

スタンドアロンの非クラスタ・システムの場合は、`page_text_to_local` を 0 に設定してください。

関連するパラメータ

なし

付加情報

[メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

swapmem_on

swapmem_on は、擬似スワップ予約を使用可能にします。

指定可能な値:

最小値	0 (擬似スワップ予約は使用不可能)
最大値	1 (擬似スワップ予約を使用可能)
デフォルト値	1

このパラメータには、整数値 0 または 1 を指定してください。

詳細は、[パラメータ値の指定](#) を参照してください。

説明

swapmem_on は、「擬似スワップ」予約を使用可能/使用不可能にします。擬似スワップとはシステム・メモリ内の領域であり、ディスク上のデバイス・スワップ・スペースに加えて使用できる仮想メモリ・スペースとみなされます。デフォルトでは、擬似スワップを使用できます。

仮想メモリ (スワップ) スペースは、通常システム・ディスクのデバイス・スワップ領域から割り当てられます。ただし大量の RAM を搭載し、大容量のディスクまたはディスク・アレイを持つシステムでは、割り当てられたデバイス・スワップ・スペースのみに制限しない方がよい場合もあります。

例えば、200 MB の RAM を搭載し、そのうち 20 MB だけがカーネルで使用されており、ルート・ディスク・アレイ上に 1 GB のスワップ領域があるシステムを、シングル・ユーザー・モードで実行している場合を考えてみましょう。

このとき、1.1 GB のスワップ・スペースが必要なプロセスを実行するとします。

他のユーザーはシステム上でプロセスを実行していないため、使用されていない RAM 領域にスワップ・システムからアクセスできるようにすると、十分なスワップ・スペースが提供されます。このような場合に、swapmem_on を使用します。

ワークステーションや小規模なシステムの管理者は、システムやユーザーの要件によっては、この機能を使用不可能にした方がよい場合もあります。

関連するパラメータ

なし

付加情報

[メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

swchunk

スワップに使用するチャンク・サイズを指定します。

指定可能な値:

最小値	2048
最大値	65536
デフォルト値	2048

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

注記

32 GB を超えるスワップを構成する必要がある場合を除き、デフォルト値の 2048 を使用してください。このパラメータを変更する場合は、まず [maxswapchunks](#) の説明をお読みください。

説明

swchunk は、スワップのチャンク・サイズを定義します。この値には 2 の累乗となる整数値を指定してください。

スワップ・スペースは「チャンク」単位で割り当てられ、各チャンクには、それぞれが DEV_BSIZE バイトのブロックが swchunk 分含まれています。システムがスワップ・スペースを必要とするときには、デバイスまたはファイル・システムから 1 つのスワップ・チャンクが取得されます。このチャンクが使い尽くされて新しいスペースが必要になると、新しいチャンクが別のデバイスまたはファイル・システムから取得されます。このように、スワップが複数のデバイスやファイル・システムに分散されるため、システムの効率が向上し、スワップ・システムによる特定のデバイスの占有が最小限に抑えられます。

関連するパラメータ

スワップ・スペースの合計値は、以下のように計算されます。

$swchunk \times \text{maxswapchunks} \times DEV_BSIZE$

DEV_BSIZE は 1024 バイトです。

システムにより設定されるスワップ・スペースの最大合計値は 1 TB (1.09 兆バイト) です。

付加情報

[メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vps_ceiling

vps_ceiling は、カーネルがシステム構成およびオブジェクト・サイズに基づいてページ・サイズを選択するときの、選択可能な最大ページ・サイズを KB 単位で指定します。

指定可能な値:

最小値	4
最大値	65536
デフォルト値	16

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは、TLB (変換索引バッファ)ミスによるサイクル時間の損失を最小限に抑える手段として用意されています。そのような損失は TLB が小さく、ハードウェアによる TLB ウォークを持たない PA-8000 のような、新しい PA-RISC で起こります。

ユーザー・アプリケーションで `chattr` コマンドを使ってプログラム・テキストやデータ・セグメントのページ・サイズを指定しない場合は、システム構成とオブジェクト・サイズに基づいて、適切なページ・サイズがカーネルにより選択されます。これを透過的な選択と呼びます。次に、選択されたサイズと、システム・ブート時に構成される vps_ceiling によって定義されるデフォルトの最大ページ・サイズとが比較されます。選択された値が vps_ceiling を超える場合は、vps_ceiling の値が使用されます。

選択された値は、システム・ブート時に構成される `vps_pagesize` によって定義されるデフォルトの最小ページ・サイズとも比較されます。選択された値が vps_pagesize より小さければ、vps_pagesize の値が使用されます。

vps_ceiling に指定したページ・サイズが適切でなければ、カーネルにより、指定した値を下回る範囲で最大の適切な値に自動的に変更される点にも注意してください。

これらのパラメータの使い方と、ユーザー・アプリケーションでページ・サイズを指定した場合にシステム処理に与える影響については、`/usr/share/doc/var_pages.ps` に PostScript ファイル形式で用意されているホワイトペーパーを参照してください。

付加情報

[x メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vps_chatr_ceiling

vps_chatr_ceiling は、ユーザー・プロセスで chatr コマンドを使ってページ・サイズを指定するときの、指定可能な最大ページ・サイズを KB 単位で指定します。

指定可能な値:

最小値	4 KB
最大値	1048576 KB
デフォルト値	1048576 KB

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは、TLB (変換索引バッファ)ミスによるサイクル時間の損失を最小限に抑える手段として用意されています。そのような損失は TLB が小さく、ハードウェアによる TLB ウォークを持たない PA-8000 のような、新しい PA-RISC で起こります。

ユーザー・アプリケーションでは、chatr コマンドを使用して、システム構成とオブジェクト・サイズに基づいてプログラム・テキストやデータ・セグメントのページ・サイズを指定でき、全体的な性能を柔軟に向上できます。アプリケーションで指定した値は、システム・ブート時にカーネル内で構成される vps_chatr_ceiling によって定義される最大ページ・サイズと比較されます。指定した値が vps_chatr_ceiling を超える場合は、vps_chatr_ceiling の値が使用されます。

これらのパラメータの使われ方と、ユーザー・アプリケーションでページ・サイズを指定した場合にシステム処理に与える影響については、/usr/share/doc/var_pages.ps に PostScript ファイル形式で用意されているホワイトペーパーを参照してください。

付加情報

[メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vps_pagesize

vps_pagesize は、ユーザー・アプリケーションで chatr コマンドを使ってページ・サイズを指定しない場合に、カーネルにより使用されるデフォルトのユーザー・ページ・サイズを KB 単位で指定します。

指定可能な値:

最小値	4
最大値	65536
デフォルト値	4

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定 を参照してください。

説明

このパラメータは、TLB (変換索引バッファ)ミスによるサイクル時間の損失を最小限に抑える手段として用意されています。そのような損失は TLB が小さく、ハードウェアによる TLB ウォークを持たない PA-8000 のような、新しい PA-RISC で起こります。

ユーザー・アプリケーションで `chatr` コマンドを使ってプログラム・テキストやデータ・セグメントのページ・サイズを指定しなければ、システム構成とオブジェクト・サイズに基づいて、適切なページ・サイズがカーネルにより選択されます。これを透過的な選択と呼びます。次に、選択されたサイズと、システム・ブート時に構成される `vps_pagesize` によって定義されるページ・サイズとが比較されます。選択された値が `vps_pagesize` より小さければ、`vps_pagesize` の値が使用されます。

これらのパラメータの使われ方と、ユーザー・アプリケーションでページ・サイズを指定した場合にシステム処理に与える影響については、`/usr/share/doc/var_pages.ps` に PostScript ファイル形式で用意されているホワイトペーパーを参照してください。

付加情報

[メモリ・ページング・パラメータの概要](#)

[メモリ・ページングに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

プロセス管理に関するパラメータの概要

- システム上のプロセス数およびユーザーあたりのプロセス数を管理して、システム・リソースがユーザー間で効果的に分散されるように維持し、システム全体の動作を最適な状態に保つ。
- 同じ優先度または異なる優先度の互いに競合する複数のプロセスに対し、CPU 時間を割り当てる。
- プロセス間で仮想メモリを割り当てて、リソースを過度に消費するプロセスや暴走したプロセスから、システムやリソースが競合しているユーザーを保護する。

プロセス管理

プロセスに関する必要条件は、個々のシステムによって大きく異なります。例えば、CDE 環境や Motif 環境が稼動している平均的なワークステーションでは、1 人のシステム・ユーザーをサポートするために、100 以上のプロセスを同時に実行することがあります。一方、シンプルな ASCII ユーザー端末が接続された大規模なマルチユーザー・システムの場合には、個々の端末で実行されるプロセスの数は 2 つまたは 3 つに過ぎなくても、端末の数は合計 1000 台にも達することがあります。この 2 つの対照的

な例を見ても分かるように、カーネルによってシステム上のプロセスがどのように制限されるかは、システムの構成によって大きく異なります。

システム・プロセスの管理に関する構成可能パラメータには、以下の 2 つがあります。

nproc

システム全体で同時に実行するプロセスの最大許容数を定義します。この数には、他のシステムから remsh などのネットワーク・コマンドを通じて開始されたリモート実行プロセスも含まれます。

maxuprc

任意の 1 人のユーザーが同時に実行するプロセスの最大許容数を定義します。この数には、ログイン・シェル、ユーザー・インターフェイス・プロセス、実行中のプログラムと子プロセス、I/O プロセスなどが含まれます。X-window、CDE、および Motif では、1 人のユーザーが同じログイン名 (ユーザーID) を使って同時に複数のログインを行うことがあります。このような場合には、プロセスの所属先のプロセス・グループの違いにかかわらず、すべてのプロセスの個数が合算されます。プロセスが親プロセス・グループから切り離された場合、そのプロセスはカウントされなくなります (ライン・プリンタ・スプーラ・ジョブや特定の専用アプリケーションなど)。

nproc の値は、通常時の最大数のユーザーがログインしているときに、すべてのユーザーに対して十分な数のプロセスを提供できるように設定する必要があります。

maxuprc の値は、すべてのシステム・ユーザーの通常のニーズを満たすことができるように設定する必要があります。ただし、この値を大きく設定しすぎると、暴走したプログラムがあまりにも多数のプロセスを起動する可能性が生じ(そのため、他のユーザーが新しいプロセスを起動できなくなる)、また、システムを悪用するユーザーがいた場合に他のユーザーを保護できなくなるので注意が必要です。

カーネル・スレッド

複数のプロセッサを備えた大規模なシステムでは、プロセスの一部を複数のスレッドに分割し、異なるプロセッサを使って同時に実行することがあります。スレッド・プロセスによるシステム・リソースの消費は、以下の 2 つのカーネル・パラメータを使って管理できます。

max_thread_proc

1 つのプロセスが作成して同時に実行するスレッドの最大許容数を定義します。

nkthread

システム上に同時に存在するカーネル・スレッドの最大許容数を定義します。

CPU 時間共有の管理

カーネルは、他のプロセスが CPU 時間を要求していないかどうかを頻繁にチェックします。CPU 要求のチェックは、優先順位の高いプロセスから順に 10 ミリ秒おきに行われます。同じ優先順位レベルの複数のプロセスが同時に CPU 時間を要求している場合は、1 つのプロセスが CPU を排他的に使用しないように、CPU 時間が複数のセグメントに「スライス化」され、総当たり方式でプロセスからプロセスに受け渡されます。

timeslice

1つのプロセスがCPUを継続して使用する時間の最大許容値を設定します。この時間が経過すると、同じ優先順位の競合プロセスにCPU時間が渡されます。timesliceに0より大きい値を指定すると、その値に10ミリ秒を掛けた時間間隔でCPUアクセスのタイム・スライス割り当てが行われます。-1を指定すると、総当たり式のCPUスケジュールが無効になりますが、特別なシステム・ニーズがない限りこの設定を使用するのは避けてください。

メモリ割り当ての管理

PA-RISCハードウェアでは、最大で2GBの仮想メモリを単一のプロセスに割り当てることができます。このメモリは、テキスト・スペース、データ・スペース、動的記憶スペース、およびその他の用途に予約される約200MBのスペースに分割されます。テキスト・スペースにはプログラムが格納され、データ・スペースにはグローバル変数、静的変数、main()内で適用されるローカル変数、文字列などが格納されます。動的記憶スペースには、スタック、レジスタ、malloc()で割り当てられたスペースなどが含まれます。ただし、大半のプロセスでは2GBものスペースが必要になることはなく、しかも、多くのシステムでは、実際にインストールされているディスク・スペースは2GB以上です。

プロセス・スペースの割り当てに対して保護制限を設定できるように、以下の構成可能カーネル・パラメータが用意されています。

maxtsiz

テキスト・セグメント・スペースの最大許容サイズを設定します。このスペースには、プログラムのうち、不変の実行可能コード部分が格納されます(32ビット・プロセスの場合)。

maxtsiz_64bit

テキスト・セグメント・スペースの最大許容サイズを設定します。このスペースには、プログラムのうち、不変の実行可能コード部分が格納されます(64ビット・プロセスの場合)。

maxdsiz

静的変数や文字列のほか、sbrk()やmalloc()により割り当てられる動的なデータ・スペースが格納されるデータ記憶セグメント・スペースの最大許容サイズを設定します(32ビット・プロセスの場合)。

maxdsiz_64bit

静的変数や文字列のほか、sbrk()やmalloc()により割り当てられる動的なデータ・スペースが格納されるデータ記憶セグメント・スペースの最大許容サイズを設定します(64ビット・プロセスの場合)。

maxssiz

スタック・スペースに使用される動的記憶セグメント(DSS)スペースの最大許容サイズを設定します(32ビット・プロセスの場合)。

maxssiz_64bit

スタック・スペースに使用される動的記憶セグメント(DSS)スペースの最大許容サイズを設定します(64ビット・プロセスの場合)。

プロセス・スペースに関するニーズは、ユーザーごと、システムごとに千差万別です。したがって、どのシステムのどのユーザーに対してもテキスト・スペースとデータ・スペースを同じサイズに固定するには無理があります。例えば、研究システムや開発システムの場合、テキスト・セグメントについてはごく標準的なスペースで十分でも、データ・セグメントについてはかなり大きなスペースを必要とするユーザーがいる一方で (例えば、サイズの小さいプログラムで非常に大きな配列を処理するユーザーなど)、データ・セグメントはほとんど必要としないが、プログラムを格納するためのテキスト・セグメントについては非常に大きなスペースを必要とするユーザーがいることがあります。

一方、数百人のユーザーをサポートする運用システムの場合は、ユーザー1人あたりのデータ記憶に要するスペースはごく標準的なサイズでも、システム上のすべてのユーザーをサポートするためには膨大なスワップ・スペースが必要になることがあります。このような場合には、いずれかのユーザーのプロセスが誤動作したり、暴走して無限ループに入ったときでも、他のユーザーを保護できるように、スペースを比較的小さいサイズに制限するのが妥当です。

最適な制限の設定

スペースの制限は、システム上のすべてのユーザーの要求を満たすとともに、プログラムが暴走したときや、いずれかのユーザーが不適切な操作を行ったときに、すべてのユーザーを保護できるように最適な値に設定する必要があります (このためには、個々のユーザーのプログラム・ニーズを十二分に理解しなければなりません)。maxtsiz、maxdsiz、および maxssiz の各パラメータは、必要なスペースにアクセスしようとする個々のユーザーに対してアクセスを制限するための手段ではなく、あくまで、仮想メモリ・リソースがいずれかのプロセスやユーザーによって独占的に使用されるのを防止するための安全策として用意されています。このことを十分に認識した上で、スペースの制限を設定する必要があります。

通常、maxtsiz、maxdsiz、および maxssiz の各パラメータは、それらの値の合計値が利用可能なスワップ・スペースを上回るように設定します。このような設定が妥当となる理由は、まず、大半のユーザー・プログラムが必要とするスペースの量は、これらのパラメータによる制限値よりもかなり小さくなるからです。さらに、比較的小さい値を設定すると、ユーザーに割り当てるスペースを制限することになりがちですが、むしろ、ユーザーを誤動作から保護することを優先して値を設定することにより、システム上のどのユーザーに対しても、システム全体の性能を向上することが可能になるからです。

付加情報

[プロセス・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

プロセス管理に関する構成可能パラメータ

プロセスで使用する仮想メモリ記憶スペースに対する制限

[maxdsiz](#)

プロセス・データ・セグメントの最大許容サイズ (32 ビット・プロセスの場合)。

[maxdsiz_64bit](#)

プロセス・データ・セグメントの最大許容サイズ (64 ビット・プロセスの場合)。

maxssiz

プロセス・スタック・セグメントの最大許容サイズ (32 ビット・プロセスの場合)。

maxssiz_64bit

プロセス・スタック・セグメントの最大許容サイズ (64 ビット・プロセスの場合)。

maxtsiz

プロセス・テキスト・セグメントの最大許容サイズ (32 ビット・プロセスの場合)。

maxtsiz_64bit

プロセス・テキスト・セグメントの最大許容サイズ (64 ビット・プロセスの場合)。

システム・プロセスおよびユーザー・プロセスに対する制限

max_thread_proc

1 つのプロセスで作成されるスレッドの最大許容数。

maxuprc

任意のユーザーが同時に実行するプロセスの最大許容数。

nkthread

システム上で同時に実行するカーネル・スレッドの最大許容数。

nproc

システム全体で同時に実行するプロセスの総数の最大許容値。

システム・プロセッサ時間の割り当てに対する制限

timeslice

1 つのプロセスが CPU を使用する時間の最大値。この時間が経過すると、そのプロセスと同じ実行優先順位を持つ別のプロセスが CPU を利用できるようになります。この機能は、暴走したプロセスがシステムのロックアップを引き起こすのを防止する目的で使うこともできます。

付加情報

プロセス・パラメータの概要

構成可能カーネル・パラメータ・ヘルプのブラウズ

maxdsiz

プロセスを実行するための最大データ・セグメント・サイズをバイト単位で指定します。

指定可能な値:

maxdsiz (32 ビット・プロセスの場合)

最小値	0x40000 (256 KB)
最大値	0xffff000 (約 4 GB)
デフォルト値	0x10000000 (256 MB)

maxdsiz_64bit (64 ビット・プロセスの場合)

最小値	0x40000 (256 KB)
最大値	0x3FFBFFFFFF (約 4 TB)
デフォルト値	0x40000000 (1 GB)

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

値はバイト単位で入力してください。

maxdsiz および maxdsiz_64bit は、それぞれ 32 ビット・プロセスおよび 64 ビット・プロセスについて、プロセスを実行するためのデータ記憶セグメントの最大サイズを定義します。このデータ記憶セグメントには、静的変数や文字列などの固定データ記憶のほか、sbrk() や malloc() を使用して割り当てられる動的なデータ・スペースなどが格納されます。

maxdsiz または maxdsiz_64bit の値は、1 つまたは複数のプロセスで大量のデータ記憶スペースを使用する場合以外は増やさないようにしてください。

システムがプロセスをロードする場合、または実行中のプロセスがデータ記憶セグメントを拡張しようとする場合は必ず、システムはプロセスのデータ記憶セグメントのサイズをチェックします。プロセスの要求が maxdsiz または maxdsiz_64bit の設定値を超えている場合、システムは呼び出し元プロセスにエラーを返し、プロセスを強制終了する場合があります。

関連するパラメータ

個々のメモリ・マップ・ファイルと共有ライブラリ・データも動的記憶領域のスペースを使用することに注意してください。このため、maxdsiz または maxdsiz_64bit に設定可能な最大値を指定した場合に、衝突が起こる可能性があります。

[maxssiz](#)

[maxtsiz](#)

付加情報

[プロセス管理パラメータの概要](#)

[プロセス管理に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

maxdsiz と maxdsiz_64bit

プロセスを実行するための最大データ・セグメント・サイズをバイト単位で指定します。

指定可能な値:

maxdsiz (32 ビット・プロセスの場合)

最小値	0x40000 (256 KB)
最大値	0xfffff000 (約 4 GB)
デフォルト値	0x10000000 (256 MB)

maxdsiz_64bit (64 ビット・プロセスの場合)

最小値	0x40000 (256 KB)
最大値	0x3FFBFFFFFF (約 4 TB)
デフォルト値	0x40000000 (1 GB)

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

値はバイト単位で入力してください。

maxdsiz および maxdsiz_64bit は、それぞれ 32 ビット・プロセスおよび 64 ビット・プロセスについて、プロセスを実行するためのデータ記憶セグメントの最大サイズを定義します。このデータ記憶セグメントには、静的変数や文字列などの固定データ記憶のほか、sbrk() や malloc() を使用して割り当てられる動的なデータ・スペースなどが格納されます。

maxdsiz または maxdsiz_64bit の値は、1 つまたは複数のプロセスで大量のデータ記憶スペースを使用する場合以外は増やさないようにしてください。

システムがプロセスをロードする場合、または実行中のプロセスがデータ記憶セグメントを拡張しようとする場合は必ず、システムはプロセスのデータ記憶セグメントのサイズをチェックします。プロセスの要求が maxdsiz または maxdsiz_64bit の設定値を超えている場合、システムは呼び出し元プロセスにエラーを返し、プロセスを強制終了する場合があります。

関連するパラメータ

個々のメモリ・マップ・ファイルと共有ライブラリ・データも動的記憶領域のスペースを使用することに注意してください。このため、maxdsiz または maxdsiz_64bit に設定可能な最大値を指定した場合に、衝突が起こる可能性があります。

[maxssiz](#)

[maxtsiz](#)

付加情報

[プロセス管理パラメータの概要](#)

[プロセス管理に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

maxssiz

動的記憶セグメント(DSS)のサイズをバイト単位で指定します。

指定可能な値:

maxssiz (32 ビット・プロセスの場合)

最小値	0x4000 (16 KB)
最大値	0x17F00000 (約 400 MB)
デフォルト値	0x800000 (8 MB)

maxssiz_64bit (64 ビット・プロセスの場合)

最小値	0x4000 (16 KB)
最大値	1073741824
デフォルト値	0x800000 (8 MB)

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

値はバイト単位で入力してください。

maxssiz および maxssiz_64bit は、それぞれ 32 ビット・プロセスおよび 64 ビット・プロセスについて、動的記憶セグメント(DSS)の最大サイズを定義します。動的記憶セグメントはユーザー・スタック・セグメント、または実行プロセスのランタイム・スタックとも呼ばれます。このセグメントには、一般にローカル変数に使用されるスタックやレジスタ記憶スペースが格納されません。

デフォルトの DSS サイズは、大部分のプロセスの要求を満たしています。したがって、maxssiz または maxssiz_64bit の値は、1 つまたは複数のプロセスで大量の動的記憶スペースを必要とする場合以外は増やさないようにしてください。

スタックは動的に拡大します。スタックが大きくなると、システムはプロセスのスタック・セグメントのサイズをチェックします。スタック・サイズの要件が maxssiz または maxssiz_64bit の値を超えた場合、システムはプロセスを強制終了します。

関連するパラメータ

[maxdsiz](#)

[maxtsiz](#)

付加情報

[プロセス管理パラメータの概要](#)

[プロセス管理に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

max_thread_proc

max_thread_proc は、1 つのプロセスで作成できるスレッドの最大数を指定します。

指定可能な値:

最小値	64
最大値	30000
デフォルト値	64

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

max_thread_proc は、1 つのプロセスで作成できるスレッドの最大数を指定します。これにより、暴走したプロセスが通常の動作時より多くのスレッドを作成した場合でも、システムがシステム・リソースを過剰に使用のを防止できます。このパラメータには、/usr/include/limits.h に定義されている変数 PTHREAD_THREADS_MAX と _SC_THREAD_THREADS_MAX に指定されている値が設定されます。

プロセスが複数のスレッドに細分された場合、プロセス・スペースの一部が各スレッド用に複製されるため、追加のメモリと他のシステム・リソースが必要となります。暴走したプロセスが大量のプロセスを作成した場合、またはユーザーが故意に大量のスレッドを作成することによりシステムに負担をかけた場合、システム性能が著しく低下したり、他の誤動作が誘発されるおそれがあります。

max_thread_proc に値を指定する場合は、システムが実行するスレッド化アプリケーションのうち最も複雑なものについて、最も条件の悪い状態で使用した場合にこのアプリケーションが必要とする、または作成するスレッド数を調べてください。

max_thread_proc には、少なくとも上記の値を設定することが必要ですが、トラブルが発生した場合に他のシステムの要求が満たされなくなることがないように、あまり大きな値に設定してはいけません。

付加情報

[プロセス管理パラメータの概要](#)

[プロセス管理に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

maxtsiz

共有テキスト・セグメントのサイズをバイト単位で指定します。

指定可能な値:

maxtsiz (32 ビット・プロセスの場合)

最小値	262144 (256 KB)
最大値	1073741824 (1 GB)
デフォルト値	0x4000000 (64 MB)

maxtsiz_64bit (64 ビット・プロセスの場合)

最小値	262144 (256 KB)
最大値	4398046507008 (約 4 TB)
デフォルト値	0x4000000 (64 MB)

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

maxtsiz および maxtsiz_64bit は、それぞれ 32 ビット・プロセスおよび 64 ビット・プロセスについて、実行中プロセスの共有テキスト・セグメント(プログラム記憶スペース)の最大サイズを定義します。プログラムの実行可能オブジェクト・コードは読み取り専用データとして保存されるので、プログラムを 2 つ以上のプロセスが同時に実行している場合、これらのプロセスはこのオブジェクト・コードを共有できます。

通常のデフォルト値で、大部分のプロセスのテキスト・セグメントを格納できます。テキスト・セグメントが 64 MB を超えるプロセスを実行する予定がない限り、maxtsiz または maxtsiz_64bit を変更しないでください。

システムが共有テキストを持つプロセスをロードする度に、システムは共有テキスト・セグメントのサイズをチェックします。プロセスのテキスト・セグメントが `maxtsiz` または `maxtsiz_64bit` の値を超えた場合、システムはエラー・メッセージを出力して、プロセスを中止します。

`maxtsiz` および `maxtsiz_64bit` は、カーネルを再構築するか、または実行中のカーネル内で `setttune()` を使用することにより設定できます。SAM および `kmtune` は `setttune()` を使用します。`maxtsiz` および `maxtsiz_64bit` に対する動的な変更は、以降に実行される `exec()` コールにのみ影響を与えます。動的な形でこれらのパラメータ値を小さくしても、`exec()` コールが実行されるまでは、実行中のプロセスに対する影響はありません。

関連するパラメータ

[maxtsiz](#)

[maxssiz](#)

付加情報

[プロセス管理パラメータの概要](#)

[プロセス管理に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

maxuprc

同時に実行できるユーザー・プロセスの最大数を指定します。

指定可能な値:

最小値	3
最大値	Nproc-5
デフォルト値	50

このパラメータには、**整数値**を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

`maxuprc` は、各ユーザーがシステム上で同時に実行できるプロセスの最大数を指定します。ユーザーは、ログイン・インスタンスではなくユーザーID 番号によって識別されます。各ユーザーには、ログイン・シェル用の最低1つのプロセスと、このプロセス・グループ内で生成される他のすべてのプロセスのために追加のプロセスが必要です(通常はデフォルト値が適しています)。

スーパーユーザーにはこの制限は適用されません。

パイプラインでは、"`|`" の両側で同時に実行するプロセスが最低1つずつ必要です。一部のコマンド(`cc`、`fc`、`pc` など)は、1回実行するごとに複数のプロセスを使用します。

ユーザーが起動しようとした新しいプロセスにより、総プロセス数が maxuprc の設定値を超えた場合、以下のエラー・メッセージが出力されます。

```
no more processes
```

ユーザーが fork() システム・コールを実行して新しいプロセスを作成することにより、このユーザーの総プロセス数が maxuprc の設定値を超えた場合、fork() は -1 を返し、errno を EAGAIN に設定します。

maxuprc は、カーネルを再構築するか、または実行中のカーネル内で settune() を使用することにより設定できます。SAM および kmtune は settune() を使用します。maxuprc に対する動的な変更は、以降に実行される fork() コールにのみ影響を与えます。maxuprc の値をユーザーの現在のプロセス数より小さくしても、実行中のプロセスへの影響はありません。ただし実行中のプロセスがいくつか終了して、ユーザー・プロセス数が新しい制限値未満になるまでは、そのユーザーのプロセスから fork() を実行することはできません。

関連するパラメータ

maxuprc は常に、nproc (システム上の最大プロセス数) より小さい値でなければなりません。この値を超えた場合、maxuprc は上限値ではなくなるため、1 人のユーザーによってシステム・リソースが占領されるおそれがあります。

付加情報

[プロセス管理パラメータの概要](#)

[プロセス管理に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nkthread

nkthread は、すべてのプロセスが同時に実行可能なシステム全体におけるスレッドの最大総数を指定します。

指定可能な値:

最小値	50
最大値	30000
デフォルト値	(nproc*2)+16

整数値または数式が返す値を指定します。詳細は、[パラメータ値の指定](#) を参照してください。

説明

性能を向上させるためにスレッドを使用するプロセスは、プロセス・スペースの一部について複数のコピーを作成します。これにより、スレッドの保存用メモリ・スペースだけでなく、スレッド管理に関するプロセッサとシステムのオーバーヘッドも必要に

なります。大規模なスレッド・アプリケーションを実行するシステムでは、大量のスレッドが必要になる場合があります。1つのプロセスが作成できるスレッド数はカーネル・パラメータ `max_thread_proc` で制限されますが、システム上には、他にも大量のスレッドを使用するスレッド・アプリケーションが存在する場合や、それらのアプリケーションが必要とするスレッド数がそれほど多くない場合もあります。

`nkthread` は、システム上のすべてのプロセスから同時に実行可能なスレッドの総数を制限します。このパラメータ値により、通常の適切な動作状態を超える大量のスレッドによってシステムが負荷を受けることを防止できます。このパラメータは、多数の大規模なアプリケーションが実行されている場合にシステムの過負荷を防止したり、故意に大量のスレッド・プログラムを起動してシステム動作を妨害し、通常のシステム動作に必要なリソースを使用不可能にしようとするユーザーからシステムを保護します。

デフォルト値では、1 プロセスにつき平均 2 つのスレッドと、システム用にいくつかのスレッドを使用できます。これより多くのスレッドを使用する必要がある場合は、以下の手順を行います。

- システム上の各スレッド・アプリケーション(特に大規模なアプリケーション)が必要とするスレッドの総数を調べます。
- これらのアプリケーションのうちどのアプリケーションが同時に実行されるか、またその数を調べます。
- 上記の数を合計して、さらに他のユーザーまたはプロセスが実行する可能性のあるスレッド用に確保しておく適切なスレッド数を追加します(`nproc*2` の値を使用すると便利です)。
- 上記で算出した総スレッド数よりも十分に大きい値で、かつシステムの完全性に影響を与えない範囲の値を `nkthread` に設定します。

同時に実行されるスレッド・アプリケーションが大量にある大規模なシステムでは、システムのプロセス管理ツールやプログラムを使って、システムを試験的に動作させてみる必要があるでしょう。

付加情報

[プロセス管理パラメータの概要](#)

[プロセス管理に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nproc

同時に存在できるプロセスの数の制限。

指定可能な値:

フェールセーフ	4200
デフォルト	4200
許容値	100 - 30000

この値は許容値より高く設定できますが、このような値が必要となることはないでしょう。nproc を 110 未満に設定すると、システムのマルチユーザーモードでの実行に支障があります。構成によっては、最小値を高くすることができます。

nproc は、nkthread + 100 より大きくなければなりません。

nproc は、maxuprc + 100 より大きくなければなりません。

推奨値 プロセッサあたり 1000 プロセス

説明

nproc 調整可能パラメータは、任意の時点において、システム上に存在できるプロセスの絶対数を制御します。この値を増やすと、存在できるプロセスが増え、減らすと、プロセスの数が制限されます。

proc: table is full メッセージがメッセージバッファーに表示された場合は、nproc が低すぎるのがわかります。このメッセージバッファーは、dmesg または syslog から読み取ることができます。このメッセージは、アプリケーションが新しいプロセスを作成できなかったことを示します。nproc の設定値が低すぎると、新しいプロセスを fork できないことが原因で、アプリケーションエラーが発生する可能性があります。

pstat_dynamic を呼び出して psd_numprocsallocd を検査することによって、同時に使用されていたプロセスの数がわかります。このフィールドは、同時に使用されていたプロセスの数の「ハイウォーターマーク」を示しています。

この調整可能パラメータの変更を行う対象ユーザー

多数のプロセスを実行することが予想されるユーザー。

変更に関する制限事項

ありません。この調整可能パラメータは動的です。

この調整可能パラメータの値を大きくする場合

多数のプロセスを同時に実行するときに、この調整可能パラメータを増やす必要があります。

この値を大きくした場合の影響

ありません。

この調整可能パラメータの値を小さくする場合

システム上のプロセスの数を制限する場合、またはメモリーが圧迫されていて、かつ nproc の値が予想をはるかに超える大きさである場合に限り、この値を減らす必要があります。

この値を小さくした場合の影響

新しいプロセスを fork できないことが原因で、アプリケーションエラーが発生するリスクが増えます。

同時に変更する必要がある他の調整可能パラメータ

nkthread は、絶対に nproc より大きくなければなりません。デフォルトの式はこれを徹底するものであり、また、ランタイムカーネルもこの点をチェックします。

アプリケーションからファイルを操作できるようにする場合は、nfile と ninode を増やす必要があります。(デフォルトではありません)

デフォルトの式が nproc の倍数であるため、ksi_alloc_max を増やす必要があります。

maxuprc は、nproc より小さくなければなりません。

これらの関連する調整可能パラメータのいくつかは、nproc の変更に応じて自動的に調整されていましたが、今後は明示的に調整を行う必要があります。

また、調整可能パラメータには、リポートしないと有効にならないものがあります。したがって、リポートせずに nproc を大幅に増やす場合は注意が必要です。

付加情報

[プロセス管理パラメータの概要](#)

[プロセス管理に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

timeslice

timeslice は、スケジューリング・タイムスライスの間隔を定義します。

指定可能な値:

最小値	-1
最大値	2147483647 (約 8 か月)
デフォルト値	10 (単位は 10 ミリ秒)

整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

timeslice 間隔は、1 つのプロセスが CPU を連続して使用できる時間を示しており、この時間が経過すると、同じ優先順位の別のプロセスに処理が移されます。timeslice の値は、10 ミリ秒単位で指定します。以下に示す 2 つの値は、特別な意味を持っています。

0 システム・デフォルト値が使われます。現時点では 10 × 10 ミリ秒、つまり 100 ミリ秒です。

-1 総当たり式のスケジューリングが完全に無効になります。

システムに与える影響

timeslice で指定した時間が経過すると、保留中のシグナルをチェックするようプロセスに指示が出されます。これにより、システム・コールを実行しないプロセスを中止させることができます (無限ループに入った暴走プロセスなど)。timeslice に非常に大きい値や-1 を指定した場合は、このようなプロセスがシグナルのチェックなしに続行されるため、システム性能のボトルネックとなったり、システムのロックアップが引き起こされる可能性があります。

特殊なリアルタイム要件を持つシステム・アプリケーションなどで特に必要となる場合を除き、timeslice はデフォルト値のままにしておいてください。

このパラメータに関連するメモリ割り当てはありません。各タイムスライス間隔では多少の CPU 時間が消費されますが、この時間は精密には測定されていません。

関連するパラメータ

なし

付加情報

[プロセス管理パラメータの概要](#)

[プロセス管理に関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

キャラクタ・モード I/O ストリームに関するパラメータの概要

HP-UX オペレーティング・システムは、raw (キャラクタ・モード) デバイス・ファイルおよびそれに対応する端末や pty などのデバイスに関連付けられたカーネル・ドライバとの間でシリアル・データをやり取りするときに、ストリームと呼ばれる I/O パイプラインを使用します。ストリームには、1 つまたは複数のストリーム・モジュールを挿入 (プラグイン) できます。これにより、データの暗号化または圧縮、文字またはメッセージの変換、データ・パケットに対する情報のアタッチ、メッセージに対するプロトコル・データの追加など、さまざまな処理を実行することができます。

ストリーム内の各モジュールは、パイプを通じて転送されるデータに応じて特定のタスクを実行します。任意のストリームに対して、複数のモジュールをプッシュすることができます。ストリームに関する構成可能カーネル・パラメータを使うと、キャラクタ・モード I/O ストリームに割り当てられたリソースやキャラクタ・モード I/O ストリームによって消費されるリソースを管理できます。また、不適切なプッシュ処理や、その他ストリームの不適切な動作を防止するのにも役立ちます。

ストリームの詳細およびストリーム・プログラミングの技法は、STREAMS/UX for the HP 9000 Reference Manual を参照してください。

なお、ストリーム I/O パイプは、HP-UX および UNIX で従来からサポートされてきたファイル・システム・パイプラインとは異なることに注意してください。

付加情報

[プロセス・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ストリームに関する構成可能カーネル・パラメータ

キャラクタ・モード I/O ストリームに使用するリソースの管理用の構成可能カーネル・パラメータは、以下に示すとおりです。

NSTRBLKSCHED

これは、出荷前の設定専用のパラメータです。当社が認定したサポート・スタッフから特に指示がない限り変更しないでください。このパラメータに関する技術情報は公開されていません。このパラメータを不適切に使用すると、システムのロックアップなどの問題が発生する可能性があります。

NSTREVENT

システム上に同時に存在する未処理のストリーム bufcall の最大許容数。このパラメータには、ストリーム上にプッシュされたすべての結合モジュールで最大生成される bufcall と等しいか大きな値を設定する必要があります。また、これによって bufcall によるリソースを制限することができます。

NSTRPUSH

システム上の 1 つのストリーム内に同時に存在するストリーム・モジュールの最大許容数。このパラメータは、ソフトウェアに欠陥がある場合に過度に多数のモジュールがストリームにプッシュされるのを防ぐのに役立ちます。しかし、ストリームの不適切な使用に対する保護手段としては適していません。

NSTRSCHED

システム上で同時に実行されるストリーム・スケジューラ・デーモンの最大許容数。この値は、システム内にインストールされているプロセッサの数と関連します。

STRCTLSZ

システム上の任意のストリーム・メッセージの制御部分に格納される制御バイトの最大許容数。

STRMSGSZ

プロセス・テキスト・セグメントの最大許容サイズ (64 ビット・プロセスの場合)。

nstrpty

システム上に同時に存在するストリーム・ベースの PTY の最大許容数。これは、システム全体に適用される値です。

streampipes

すべてのパイプを強制的にストリーム・ベースにします。

付加情報

[ストリーム・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

STREAMS/UX for the HP 9000 Reference Manual

NSTREVENT

NSTREVENT は、システム上に同時に存在可能な未処理のストリーム bufcall の最大数を指定します。

指定可能な値:

最小値	なし
最大値	なし
デフォルト値	50

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは、ストリーム内に存在可能な未処理 bufcall の最大数を設定します。ストリーム内に存在する bufcall の数は、このストリームにプッシュされたストリーム・モジュールの数と性質により決まります。

このパラメータは、すべてのストリーム内で実行されるさまざまなモジュールが過剰な数の bufcall を発行した場合に発生するリソースの過負荷からシステムを保護することを目的としています。パラメータの値には、システム上の全ストリームを対象とし、通常の動作範囲内で発行が予測される bufcall の最大数以上の値を設定することが必要です。この値は、使用可能な各ストリーム・モジュールの動作や構造、およびシステム内のすべてのストリームに同時にプッシュされるモジュールの数や組合せにより決まります。

付加情報

[ストリームに関するカーネル・パラメータの概要](#)

[ストリームに関する構成可能カーネル・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

STREAMS/UX for the HP 9000 Reference Manual

NSTRPUSH

NSTRPUSH は、システム上の 1 つのストリーム内に同時に存在可能なストリーム・モジュールの最大数を指定します。

指定可能な値:

最小値	なし
最大値	なし
デフォルト値	16

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

このパラメータは、任意のストリームへプッシュ可能なストリーム・モジュールの最大数を指定します。これにより、暴走プロセスがストリームにプッシュするモジュールを自動的に選択するのを防止できます。ただしこのパラメータは、システム・ユーザーによるストリーム・モジュールの悪用を防止することを目的としていません。

大部分のシステムでは、ストリーム内で 3 つまたは 4 つ以上のモジュールは必要ありません。ただし稀に、これより多くのモジュールが必要になる場合があります。このパラメータのデフォルト値では、ストリーム内に 16 個のモジュールを設定できます。この値は、必要なモジュール数が最も多いインストール環境やアプリケーションにも十分対応できます。

お使いのシステムがストリーム内で必要とするモジュール数が 16 個を超える場合、必要とされるモジュール数や、他のシステム・リソースに関する要求事項(未処理の bufcall)や他の要因について慎重に検討することが必要です。

付加情報

[ストリームに関するカーネル・パラメータの概要](#)

[ストリームに関する構成可能カーネル・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

STREAMS/UX for the HP 9000 Reference Manual

NSTRSCHED

NSTRSCHED は、システム上で同時に実行可能なストリーム・スケジューラ・デーモン(smprocsched)の最大数を指定します。

指定可能な値:

最小値	0
最大値	32

デフォルト値 0

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

このパラメータは、複数のプロセッサを格納するシステム上で実行されるマルチ・プロセッサ(MP)ストリーム・スケジューラ・デーモンの最大数を指定します。単一プロセッサ(UP)システムは MP スケジューラ・デーモンを使用しませんが、MP システムと UP システム共に、必ず 1 つの UP ストリーム・スケジューラ(supsched)を備えていることに注意してください。

パラメータ値が 0 に設定されている場合、システムは、システム内のプロセッサの数に基づいて、実行するデーモンの数を決定します。プロセッサが 2~4 個の場合は 1、5~8 個の場合は 2、9~16 個の場合は 3、16 個を超える場合は 4 に設定されます。

パラメータ値が 0 でない正数に設定されている場合、smpsched で設定されている個数のデーモンが MP システム上に生成されます。

このパラメータは HP システムのみで使用され、将来の HP-UX リリースでは削除される予定です。

付加情報

[ストリームに関するカーネル・パラメータの概要](#)

[ストリームに関する構成可能カーネル・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

STREAMS/UX for the HP 9000 Reference Manual

STRCTLSZ

STRCTLSZ は、システム上の各ストリーム・メッセージの制御部分に格納できる制御バイトの最大数を設定します。

指定可能な値:

最小値 0
最大値 メモリ・サイズに依存
デフォルト値 1024 バイト

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

STRCTLSZ は、putmsg()によりシステム上の各ストリーム・メッセージの制御部分に挿入できる制御データのバイト数を制限します。パラメータを 0 に設定した場合は、メッセージの制御セグメントに格納できるバイト数に制限はありません。

送信しようとするバッファのサイズが STRCTLSZ の値を超える場合は、putmsg()から ERANGE エラーが返されます。

付加情報

[ストリームに関するカーネル・パラメータの概要](#)

[ストリームに関する構成可能カーネル・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

STREAMS/UX for the HP 9000 Reference Manual

STRMSGSZ

STRMSGSZ は、システム上の各ストリーム・メッセージに格納できるデータの最大バイト数を設定します。

指定可能な値:

最小値	0
最大値	メモリ・サイズに依存
デフォルト値	8192 バイト

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

このパラメータは、putmsg()または write()によりシステム上の各ストリーム・メッセージのデータ部分に挿入できるデータのバイト数を制限します。パラメータを 0 に設定した場合は、メッセージのデータ・セグメントに格納できるバイト数に制限はありません。

送信しようとするバッファのサイズが STRMSGSZ の値を超える場合、putmsg()からは ERANGE エラーが返され、write()ではデータが複数のメッセージに分割されます。

付加情報

[ストリームに関するカーネル・パラメータの概要](#)

[ストリームに関する構成可能カーネル・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

STREAMS/UX for the HP 9000 Reference Manual

nstrpty

nstrpty は、システム上に同時に存在可能なストリーム・ベースの PTY の最大数を指定します。

指定可能な値:

最小値	0
最大値	メモリ・サイズに依存
デフォルト値	0

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

nstrpty は、システム全体で存在可能なストリーム・ベースの PTY 数を制限します。データを PTY デバイス(ウィンドウなど)に送信する際、オープンしているすべてのウィンドウに対して PTY デバイスが常に設定されていることが必要です。

nstrpty は、ストリーム・ベースの I/O パイプを使用するシステム上の PTY デバイスの数以上の値に設定することが必要です。PTY 数よりはるかに大きい数をこのパラメータ値に設定することはお勧めできません。nstrpty は、ストリーム・ベースの PTY をサポートするデータ構造をカーネル内に作成する際に使われます。この値を大きく設定しすぎた場合、カーネルのメモリ・スペースが浪費されます。

付加情報

[ストリームに関するカーネル・パラメータの概要](#)

[ストリームに関する構成可能カーネル・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

STREAMS/UX for the HP 9000 Reference Manual

streampipes

streampipes は、すべてのパイプを強制的にストリーム・ベースにします。

指定可能な値:

最小値	0
最大値	1
デフォルト値	0

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

このパラメータは、pipe()システム・コールで作成されるパイプのタイプを定義します。デフォルト値である0の場合は、pipe()で作成されるパイプはすべて、通常のHP-UXファイル・システム・パイプになります。このパラメータを1に設定した場合は、pipe()で作成されるパイプはストリーム・ベースになり、作成されるストリーム上にモジュールをプッシュできるようになります。

このパラメータに0でない値を設定する場合は、pipemodおよびpipedevのモジュールとドライバを、/stand/systemファイル内に構成しなければなりません。

付加情報

[ストリームに関するカーネル・パラメータの概要](#)

[ストリームに関する構成可能カーネル・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

STREAMS/UX for the HP 9000 Reference Manual

System V のプロセス間通信機構

HP-UX オペレーティング・システムでは、共有メモリを使用して、連携動作するプログラムやプロセスの間で互いに情報をやり取りすることができます。以下の3種類のプロセス間通信機構が用意されています。

メッセージ

メッセージ・データは、共有メモリの特定の領域に格納されます。受信側プログラムは、そのメモリ領域からメッセージ・データを取得します。メッセージ用の共有メモリ・スペースの割り当てと管理の詳細は、[IPCメッセージ待ち行列の処理](#)の概要を参照してください。

セマフォ

連携して動作するプロセスの現在のステータスを通知するための増減カウンタを格納するための共有記憶領域。セマフォ用の共有メモリ・スペースの割り当てと管理の詳細は、[IPCセマフォの処理の概要](#)を参照してください。

共有メモリ

複数のプロセスによって共有される予約済みデータ記憶領域。この領域は、各連携プロセスにおいて同一となるように定義されたデータ構造を通じて共有されます。共有データ構造用の共有メモリ・スペースの割り当てと管理の詳細は、[共有メモリの処理の概要](#)を参照してください。

付加情報

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

メッセージ待ち行列の処理の概要

メッセージ

メッセージは、互いに連携して動作するプログラムの中でやり取りされる小さいサイズ (400 バイトなど) のデータです。メッセージの受け渡しは、メッセージ待ち行列を通じて行われます。

待ち行列は、異なるタイプのメッセージを格納できます。適切なパーミッションを持つプロセスであれば、どのプロセスも待ち行列内のメッセージを受け取ることができます。受信側のプロセスは、最初のメッセージ、特定のタイプの最初のメッセージ、または特定のタイプグループの最初のメッセージを取得できます。詳細は、HP-UX Reference の `msgop(2)` を参照してください。

メッセージ待ち行列

メッセージ待ち行列は、共有メモリに格納されたデータのリンク・リストとして実装されます。メッセージ待ち行列自体には、一連のデータ構造が格納されます。これらのデータ構造は、メッセージに 1 対 1 で関連付けられており、それぞれがメッセージのアドレス、タイプ、およびサイズ、さらに、待ち行列内の次のメッセージへのポインタを識別します。

プログラムから待ち行列を割り当てるには、`msgget()` システム・コールを使用します (HP-UX Reference の `msgget(2)` を参照)。メッセージを待ち行列に格納するには `msgsnd()` システム・コールを使用し、メッセージを取得するには `msgrcv()` を使用します (`msgop(2)` を参照)。メッセージ待ち行列の管理に関するその他の処理を実行するには、`msgctl()` システム・コールを使用します (`msgctl(2)` を参照)。

プログラム内でメッセージを使用する方法の詳細は、Advanced UNIX Programming (Marc J. Rochkind 著、Prentice-Hall, Inc、ISBN 0-13-011800-1) など、高度な UNIX プログラミングに関する専門書を参照してください。

メッセージおよびメッセージ待ち行列の管理

メッセージ待ち行列およびメッセージ・ヘッダ配列は、スワップ可能な共有メモリ・スペースに格納されます。これらの管理に必要なその他のデータ構造配列は、スワップ禁止のカーネル・スペースに格納されます。

メッセージ制御用のカーネル構成パラメータ

- システム上のメッセージ待ち行列の最大許容数
- メッセージ待ち行列の最大許容サイズ
- メッセージの最大許容サイズ
- 待ち行列内のすべてのメッセージのサイズの合計の最大許容値
- 待ち行列あたりのメッセージの最大許容数
- システム全体の同時メッセージの最大許容数
- メッセージ・ヘッダ・リストの最大許容サイズ
- 新しいメッセージに割り当てる空きスペースを識別するための空きスペース・マップの最大許容サイズ

メッセージ用のメモリ・スペースの管理の詳細は、[チュートリアル: メッセージ用のメモリ・スペースの割り当て](#) を参照してください。

付加情報

[メッセージ・パラメータの概要](#)

[メッセージ・パラメータの一覧](#)

[チュートリアル: プロセス間通信メッセージ用のメモリ・スペースの割り当て](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

メッセージ用メモリ・スペースの割り当て

IPC メッセージには、以下のメモリ・スペースを割り当てる必要があります。

- カーネル領域内でメッセージ識別子に使用するスペース
- メッセージ待ち行列用の共有メモリ・スペース
- メッセージ・ヘッダ用の共有メモリ・スペース

メッセージ待ち行列

待ち行列サイズ

各メッセージ待ち行列は、`msgmnb` (メッセージ待ち行列内の最大バイト数) バイトのメッセージを格納するのに十分なスペースを使って作成されます。各メッセージには `msgssz` (メッセージ・セグメント・サイズを示すパラメータ) バイトのメッセージ・セグメントが 1 つまたは複数格納されます。

メッセージ・サイズ

悪意のあるプログラムや品質に問題のあるプログラムによってメッセージ・スペースが過剰に消費されるのを避けるため、個々のメッセージのサイズは、それぞれ `msgmax` バイト以内に制限されます。各メッセージは、1 セグメントあたり `msgssz` バイトが格納されている 1 つまたは複数のセグメント群として待ち行列に保存されます。メッセージに使用されるセグメントの数は、`msgssz` を掛けた場合に、メッセージ内の合計バイト数より大きい値のうち、最も小さい整数値になります。

待ち行列スペース

待ち行列内のすべてのメッセージによって消費されるスペースの総量は、`msgmnb` (メッセージ待ち行列の最大バイト数を示すパラメータ) バイト以内に制限されます。この値を `msgmax` バイトより小さく設定してはいけません。

メッセージの総数

1 つの待ち行列内に格納されるメッセージの最大許容数は、個々のメッセージの長さ、上記の待ち行列サイズなどの制限条件によって異なります。そのため状況により変化します。ただし、これらに優先して適用されるシステム全体の制限があります。システム内のすべての待ち行列内に存在するメッセージの総数は、`msgtql` 以内に制限されます (下記の「メッセージ・ヘッダ配列」を参照してください)。

システムのグローバル制限

メッセージ待ち行列と個々のメッセージに適用される制限条件は上記のとおりですが、これらの他に、カーネルによって IPC メッセージ管理に適用される制限条件があります。ここでは、このような制限条件について述べます。

メッセージ待ち行列識別子

各メッセージ待ち行列には、メッセージ待ち行列識別子が関連付けられています。この識別子は、スワップ禁止のカーネル領域に格納されます。システム上に存在するメッセージ待ち行列の総数は、`msgmni` (識別子の最大数) によって制限されます。

メッセージ・ヘッダ配列

各メッセージ待ち行列には、メッセージ・ヘッダが関連付けられます。このヘッダは、スワップ可能な共有メモリ領域に格納されます。システム全体に存在するメッセージの総数は、`msgtql` (メッセージの総数制限) によって制限されます。

空きスペース管理

メッセージを格納するスペースは、メッセージを送信するときに割り当てられます。受信が完了すると、このスペースは他のメッセージを格納できるように解放されます。カーネルは、新しいメッセージに割り当てる空きスペースを識別するためのリソース・マップを保持しています。このマップのサイズは、`msgmap` パラメータによって制御されます。空きスペースのフラグメント化 (断片化) の詳細およびフラグメント化がリソース・マップに及ぼす影響は、`msgmap` を参照してください。

付加情報

[メッセージ・パラメータの概要](#)

[メッセージ・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

IPC メッセージに関する構成可能パラメータ

メッセージ用のスペースの割り当ては、以下のパラメータで制御できます。

`mesg`

(ワークステーションのみ)。システム・ブート時にメッセージを使用できるようにするかどうかを指定します。

`msgmap`

メッセージに共有メモリ・スペースを割り当てるための空きスペース・リソース・マップのサイズ。

`msgmax`

システム全体における個々のメッセージの最大サイズ (バイト単位)。

`msgmnb`

1 つのメッセージ待ち行列に格納されるすべてのメッセージの長さの合計の最大値 (バイト単位)

`msgmni`

システム上の任意のストリーム・メッセージの制御部分に格納される制御バイトの最大許容数。

msgseg

システム上に存在するメッセージ・セグメントの最大許容数。

msgssz

メッセージ・セグメントのサイズ(バイト単位)。

msgtql

すべてのパイプを強制的にストリーム・ベースにします。

付加情報

メッセージ・パラメータの概要

チュートリアル: プロセス間通信メッセージ用のメモリ・スペースの割り当て

構成可能カーネル・パラメータ・ヘルプのブラウズ

mesg

mesg は、システム・ブート時のカーネルでの System V IPC メッセージを使用可能/不可能に設定します (ワークステーションのみ)。

指定可能な値:

最小値	0 (カーネルから System V IPC メッセージ・パラメータを除く)
最大値	1 (カーネルに System V IPC メッセージ・パラメータを含める)
デフォルト値	1

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

mesg は、システム・ブート時にカーネルに System V IPC メッセージ・パラメータ用コードを含めるかどうかを指定します(ワークステーション・システムのみ)。

mesg = 1 コードはカーネルに含まれます(IPC メッセージは使用可能)。

mesg = 0 コードはカーネルに含まれません(IPC メッセージは使用不可能)。

サーバー・システムの場合: IPC メッセージは常にカーネル内で使用できます。

ワークステーション・システムの場合: mesg を 0 に設定すると、他のすべての IPC メッセージ・パラメータが無視されます。

関連するパラメータ

なし

付加情報

[メッセージ待ち行列の処理の概要](#)

[IPC メッセージに関する構成可能パラメータの一覧](#)

[チュートリアル: メッセージ用のメモリ・スペースの割り当て](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

msgmap

msgmap は、新しいメッセージに共有メモリ・スペースを割り当てるための空きスペース・リソース・マップのサイズを指定します。

指定可能な値:

最小値	3
最大値	メモリ・サイズに依存
デフォルト値	msgtql+2

このパラメータには、整数値を直接指定するか、または整数値を返す数式を指定してください。

詳細は、[パラメータ値の指定](#) を参照してください。

説明

メッセージ待ち行列は、共有メモリに格納されたデータのリンク・リストとして実装されます。各メッセージは、メッセージ待ち行列内の 1 つのロットまたは隣接する複数のロットで構成されます。メッセージの割り当て/割り当て解除が行われると、メッセージ用に予約されていた共有メモリ領域がフラグメント化(断片化)されます。

msgmap は、新しいメッセージへのスペース割り当てに使用するリソース・マップのサイズを指定します。リソース・マップには、すべてのメッセージ待ち行列が使用する共有メモリ・メッセージ・スペース内の空きスペースが示されます。マップの各エントリには、隣接する未割り当てロット・セットへのポインタと、このセットのサイズ(セグメント数)が格納されます。

さまざまなサイズのメッセージが格納されると、空きスペースのフラグメント化が進行します。リソース・マップは、空きスペースの各フラグメントに対するエントリを必要とするため、フラグメント化が過度に進行した場合、空きスペースのマップ配列がいっぱいになり、オーバーフローするおそれがあります。カーネルが新しいメッセージ用のスペースを要求していたり、受信メッセージが使用するスペースを解放している際にオーバーフローが発生した場合、以下のメッセージが出力されます。

DANGER: mfree map overflow

上記のエラー・メッセージが表示された場合は、msgmap の値を増やしてカーネルを再構築してください。

空きスペースのマップ・サイズ要求の評価

必要なリソース・マップのサイズは、システム上のアクティブなメッセージ待ち行列数に比例して増加します。また、待ち行列内のメッセージ・サイズの種類が増えた場合にも増加します。これにより、フラグメント化がさらに進行します。

通常、待ち行列内のすべてまたは大部分のメッセージがほぼ同じサイズの場合は、メッセージ待ち行列の空きスペースのフラグメント化が発生する割合は非常に低くなります。

関連するパラメータ

- msgmap には、(`msgtql` + 2)以下の値を指定します。
- msgmap には、(`msgseg` + 2)以下の値を指定します。
- msgmap が `msgtql` + 2 を超えた場合、msgmap に割り当てられたスペースの一部は使用できません。

付加情報

メッセージ待ち行列の処理の概要

IPC メッセージに関する構成可能パラメータの一覧

チュートリアル：メッセージ用のメモリ・スペースの割り当て

構成可能カーネル・パラメータ・ヘルプのブラウズ

msgmax

個々のメッセージの最大サイズをバイト単位で指定します。

指定可能な値:

最小値	0
最大値	<code>min(msgmnb, msgseg * msgssz, 65535)</code> バイト
デフォルト値	8192 バイト

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

msgmax は、待ち行列に入る個々のメッセージの最大サイズをバイト単位で指定します。

msgmax の値は、システム上で使用中のアプリケーションが使用するメッセージの容量が大きい場合のみ増やしてください。このパラメータにより、動作やコードに問題のあるプログラムによってメッセージ・バッファ・スペースが過剰に消費されるのを防止できます。

msgmax バイトを超えるメッセージを送信しようとする msgsnd() システム・コールは、EINVAL エラーを返します (msgop(2) を参照)

msgmax は、カーネルを再構築するか、または実行中のカーネル内で settune() を使用することにより設定できます。SAM および kmtune は settune() を使用します。msgmax は、新しいメッセージの送信時にのみチェックされます。このパラメータを変更すると、システム内で以降に実行されるすべての送信処理に影響が及びます。

関連するパラメータ

msgmax <= msgmnb

msgmax <= (msgssz × msgseg)

付加情報

[メッセージ待ち行列の処理の概要](#)

[IPC メッセージに関する構成可能パラメータの一覧](#)

[チュートリアル: メッセージ用のメモリ・スペースの割り当て](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

msgmnb

1 つのメッセージ待ち行列に同時に格納できるすべてのメッセージの合計サイズの最大値をバイト単位で指定します。

指定可能な値:

最小値	0
最大値	min(msgseg * msgssz, 65535) バイト
デフォルト値	16384 バイト

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

msgmnb は、1 つのメッセージ待ち行列に同時に格納されるすべてのメッセージの合計サイズの最大値をバイト単位で指定します。

この最大値を超えるメッセージを送信しようとする msgsnd() システム・コールは、以下のエラーを返します。

EAGAIN IPC_NOWAIT が設定されている場合

EINTR IPC_NOWAIT が設定されていない場合

msgmnb は、カーネルを再構築するか、または実行中のカーネル内で settune() を使用することにより設定できます。SAM および kmtune は settune() を使用します。このパラメータに対する動的な変更は、新しいメッセージ待ち行列の作成時にのみ影響を与えます。既存のメッセージ待ち行列への影響はありません。

関連するパラメータ

msgmnb >= msgmax

msgmnb <= (msgssz × msgseg)

付加情報

[メッセージ待ち行列の処理の概要](#)

[IPC メッセージに関する構成可能パラメータの一覧](#)

[チュートリアル: メッセージ用のメモリ・スペースの割り当て](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

msgmni

msgmni は、システム上に同時に存在可能なメッセージ待ち行列の最大数を指定します。

指定可能な値:

最小値	1
最大値	メモリ・サイズに依存
デフォルト値	50

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

msgmni は、システム上に存在可能なメッセージ待ち行列識別子の最大数を定義します。

システム上で作成されるメッセージ待ち行列 1 つにつき、1 つの識別子が必要です。

新しい待ち行列を使用できない場合

msgmni で指定されている最大数のメッセージ待ち行列がすでに存在する場合、新しいメッセージ待ち行列を割り当てようとすると、msgget()システム・コールにより ENOSPC エラーが返されます。HP-UX Reference の msgget(2)を参照してください。

親なしメッセージ待ち行列

プロセスがメッセージ待ち行列を割り当てた後、プロセス終了時に待ち行列の割り当て解除に失敗した場合、このメッセージ待ち行列はシステム上に残ります。破棄された待ち行列の割り当てを解除するには、ipcrm コマンドを使用します。ipcrm(1)を参照してください。

msgget()システム・コールを使ってメッセージ待ち行列を割り当てるプロセスは必ず、msgctl()システム・コールに対して IPC_RMID コマンドを使って、メッセージの割り当てを解除しなければなりません。msgctl(2)を参照してください。

メッセージ待ち行列のステータス

システム上でアクティブなメッセージ待ち行列のステータスを調べるには、ipcs コマンドを使用します。ipcs(1)を参照してください。

関連するパラメータ

なし

付加情報

[メッセージ待ち行列の処理の概要](#)

[IPC メッセージに関する構成可能パラメータの一覧](#)

[チュートリアル：メッセージ用のメモリ・スペースの割り当て](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

msgseg

msgseg は、システム上のすべてのメッセージ待ち行列内で同時に存在可能なメッセージ・セグメントのシステム全体における最大総数を指定します。

指定可能な値:

最小値	1
最大値	32767
デフォルト値	2048

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

msgseg に `msgssz` を掛けた値が、システム全体のすべてのメッセージ待ち行列に対して存在可能な共有メモリのメッセージ・スペースの合計容量になります(メッセージ・ヘッダのスペースは含まれません)。

関連するパラメータ `msgssz` (メッセージ・セグメントのサイズ(バイト単位))は、待ち行列内の各メッセージ・セグメントに予約するバイト数を指定します。メッセージが待ち行列に入ると、そのメッセージの長さにより、そのメッセージを格納するためにサイズが `msgssz` のセグメントを何個使用するかが決まります。待ち行列内で各メッセージが使用するスペースは常に、`msgssz` の整数倍数となります。

`msgseg` (メッセージ・セグメント)は、上記のセグメントを単位とする、システム全体のすべての待ち行列に使用可能なメッセージ数を定義します。

待ち行列のサイズとフラグメント化

待ち行列内のメッセージに使用できるスペースは、`msgseg × msgssz` で定義されます。この 2 つの値の比率を変更すると、あるメッセージの使用パターンに対してメッセージ・スペースがフラグメント化される方法が変わります。

付加情報

[メッセージ待ち行列の処理の概要](#)

[IPC メッセージに関する構成可能パラメータの一覧](#)

[チュートリアル：メッセージ用のメモリ・スペースの割り当て](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

msgssz

メッセージ待ち行列内にメッセージ・スペースを割り当てる際に使用するメッセージ・セグメントのサイズを指定します。

指定可能な値:

最小値	1
最大値	メモリ・サイズに依存
デフォルト値	8 バイト

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

`msgssz` に `msgseg` を掛けた値が、システム全体のすべてのメッセージ待ち行列に対して存在可能な共有メモリのメッセージ・スペースの合計容量になります(メッセージ・ヘッダのスペースは含まれません)。

`msgssz` は、IPC メッセージの保存用に割り当てられるメモリ・スペースのセグメント・サイズをバイト単位で指定します。新しいメッセージ用のスペースは、そのメッセージ全体を格納するのに必要な `msgssz` バイトごとの 1 つまたは複数のセグメントを割り当てることにより作成されます。

`msgmnb` は、各メッセージ待ち行列に格納できるスペースのバイト数を指定します。

待ち行列のスペースとフラグメント化

すべての待ち行列に使用できるスペースの合計は、`msgseg × msgssz` で定義されます。この 2 つの値の比率を変更すると、あるメッセージの使用パターンに対してメッセージ・スペースがフラグメント化される方法が変わります。

関連するパラメータ

(`msgseg × msgssz`) >= `msgmax`

付加情報

[メッセージ待ち行列の処理の概要](#)

[IPC メッセージに関する構成可能パラメータの一覧](#)

[チュートリアル：メッセージ用のメモリ・スペースの割り当て](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

msgtql

メッセージ待ち行列内にメッセージ・スペースを割り当てる際に使用するメッセージ・セグメントのサイズを指定します。

指定可能な値:

最小値	1
最大値	メモリ・サイズに依存
デフォルト値	40

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

`msgtql` は、メッセージ・ヘッダの保存領域の大きさを指定します。システムの待ち行列に格納される各メッセージごとに 1 つのメッセージ・ヘッダが作成されます。したがって、メッセージ・ヘッダ・スペースのサイズにより、システム全体で待ち行列に格納できるメッセージの最大数が定義されます。メッセージ・ヘッダは(スワップ可能な)共有メモリに保存されます。

`msgsnd()` システム・コールが `msgtql` によって設定されている最大値を超えるメッセージを送信しようとした場合、システム・コールは以下のどちらかを行います。

- IPC_NOWAIT フラグが設定されていない場合は、処理を中断してヘッダが空くのを待ちます。または、
- IPC_NOWAIT フラグが設定されている場合は、EAGAIN を返します。

関連するパラメータ

[msgmap](#) には、(msgtql + 2)以下の値を指定します。

msgmap が(msgtql + 2)を超えた場合、割り当てられているスペースの一部は無駄になります。

付加情報

[メッセージ待ち行列の処理の概要](#)

[IPC メッセージに関する構成可能パラメータの一覧](#)

[チュートリアル：メッセージ用のメモリ・スペースの割り当て](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

セマフォに関するパラメータの概要

System V IPC セマフォは、主に、プロセスを適切に同期した状態に維持し、共有データ構造へのアクセス時の衝突を防止する目的で使用されます。これらのセマフォをソフトウェア内で使用するには、かなり複雑なプログラミングが必要になるため、多くのプログラマは、セマフォ以外の有効な手段を使用できる場合には、セマフォを使用せずに済ませています。

セマフォの動作

semget()システム・コールは、指定された数のセマフォを格納する配列を割り当てます (HP-UX Reference の semget(2) を参照)。プログラムをできるだけシンプルにするために、セマフォを 1 つだけ格納する配列を割り当てるという方法がよく採られています。配列を割り当てた後のセマフォ処理は配列全体に対してアトミックに行われます。そこでは、配列内の個々のセマフォに対する操作は、1 つ以上のセマフォ操作を含んだ 1 つの配列を用いて行われます (semop(2)を参照)。semctl()を使うと、セマフォのパーミッション、変更時刻、またはオーナーを確認したり変更することができます。

セマフォは、一般に、ほかのプロセスをブロックして重要度の高い処理を行ったり、共有リソースを使用するプロセスによってインクリメントされます。このようなプロセスは、処理の完了時に値をデクリメントし、ブロックしたほかのプロセスがリソースにアクセスできるようにします。セマフォは、バイナリ・セマフォまたは汎用セマフォとして構成できます。バイナリ・セマフォは、0 および 1 の 2 つの値だけをとりま。一方、汎用セマフォ (カウンタ) は、あるプロセスによってインクリメントされ、そのプロセスに連携して動作するほかの 1 つ以上のプロセスによってデクリメントされます。検出できないオーバーフロー条件が発生するのを防止するために、カーネルは、セマフォのインクリメントを特定の最大許容値内に制限します。この制限値を設定するには、カーネル・パラメータの semvmx を使用します。ただし、このパラメータの値は、符号なし整数の上限内に制限されま。セマフォに負の値 (0 より小さい値) を与えることはできません。

セマフォのアンドゥ処理

エラーが発生したときや、プロセスを中止する必要があるとき、プロセスが異常終了したときなどには、1 つ以上のセマフォを新しい値か、または以前の値に変更しなければならなくなることがあります。これをセマフォのアンドゥと呼びます。このような条件が発生したときのセマフォの値は予測不能なので、アンドゥ処理によってセマフォの値を変更できる範囲は制限されています。この制限値を設定するには、カーネル・パラメータの [semaem](#) を使います。

System V IPC セマフォの処理の詳細は、Advanced UNIX Programming (Marc J. Rochkind 著、Prentice-Hall, Inc. ISBN 0-13-011800-1) など、高度な UNIX プログラミングに関する専門書を参照してください。

セマフォに関する制限

構成可能カーネル・パラメータには、システム上に同時に存在するセマフォ・セットの最大許容数を指定するパラメータと、ユーザーにとって利用可能な個々のセマフォがシステム上に同時に何個まで存在できるかを指定するパラメータが用意されています。なお、1 つのセマフォ・セットに格納できるセマフォの数は 500 に固定されており、この制限値を変更することはできません。

付加情報

[セマフォ・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

IPC セマフォに関する構成可能パラメータ

System V IPC セマフォの動作制限に関するカーネル・パラメータには、以下の 8 つのパラメータがあります。

[sema](#)

(ワークステーションのみ)。システム・ブート時に IPC セマフォを使用可能/使用不可能にします。

[semaem](#)

セマフォの「アンドゥ」処理でセマフォの値を変更するときの差分の最大許容値。

[semmap](#)

要求されたセマフォ・セットを割り当てるための空きセマフォ・リソース・マップのサイズ。

[semjni](#)

システム上に同時に存在する IPC セマフォ・セットの最大許容数。

[semnns](#)

システム全体において、システム・ユーザーが利用できる個々の IPC セマフォの総数。

[semnnu](#)

システム上の任意の IPC セマフォに対してアンドゥ処理を保留にするプロセスの最大許容数。

semume

任意のプロセスがアンドゥ処理を保留にする IPC セマフォの最大許容数。

semvmx

任意の IPC セマフォに設定できる値の上限。これは、検出できないオーバーフロー条件が発生するのを防止するためのパラメータです。

付加情報

[セマフォ・パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

sema

sema はシステム・ブート時に、カーネル内での System V IPC セマフォのサポートを有効/無効にします (ワークステーションのみ)。

指定可能な値:

最小値	0 (カーネルから System V IPC セマフォ・コードを除く)
最大値	1 (カーネルに System V IPC セマフォ・コードを含める)
デフォルト値	1

このパラメータには、**整数値 0 または 1** を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

sema は、システム・ブート時に System V IPC セマフォ用のコードをカーネルに含めるかどうかを決定します (ワークステーション・システムのみ)。

sema = 1 カーネル内にコードが含まれます (IPC セマフォは使用可能)。

sema = 0 カーネル内にコードが含まれません (IPC セマフォは使用不可能)。

サーバー・システム: IPC セマフォは、カーネル内で常に使用可能です。

ワークステーション・システム: shmем を 0 に設定した場合は、その他の IPC セマフォ・パラメータはすべて無視されます。

Starbase グラフィックス・ライブラリなどの一部の HP-UX サブシステムでは、セマフォを使用します。System V IPC セマフォを必要とするアプリケーションがシステム上に存在しないことが確実でない限り、セマフォは無効化しないでください。

sema を 0 に設定した場合は、システム・コール `semget()` または `semop()` を使用するすべてのプログラムから、SIGSYS シグナルが返されます。

関連するパラメータ

sema を設定しなければ、その他のセマフォ・カーネル・パラメータはすべて無視されます。

付加情報

[セマフォ・パラメータの概要](#)

[IPC セマフォに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

semaem

semaem は、セマフォの「アンドゥ」処理で変更できるセマフォの最大値を定義します。

指定可能な値:

最小値	0
最大値	SEMVMX または 32767 のうち小さい方の値
デフォルト値	16384

整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

アンドゥはセマフォ処理におけるオプションのフラグであり、起動元プロセスが中止されたような場合に、処理内容を元に戻すために使われます。

semaem は、1 回のアンドゥ処理で変更できるセマフォの最大値を指定します。

アンドゥ値はプロセスごとに累積されるため、あるプロセスでセマフォに対して複数のアンドゥ処理がある場合は、各アンドゥ処理の値が加算されてその合計値が `semadj` という名前の変数に保存されます。プロセスが中止された場合は、この `semadj` の値に基づいて、セマフォが増分または減分されます。

セマフォ・アンドゥ処理の詳細は、HP-UX Reference の `semop(2)` を参照してください。

`semop()` コールにより、`semadj` の絶対値が `semaem` を超える値になる場合は、ERANGE エラーが返されます。

関連するパラメータ

semaem には、SEMVMX 以下の値を指定してください ([semvmx](#) を参照)。

付加情報

[セマフォ・パラメータの概要](#)

[IPC セマフォに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

semmap

semmap は、共有メモリ内に新しい System V IPC セマフォを割り当てるときに使われる空きスペース・リソース・マップのサイズを指定します。

指定可能な値:

最小値	4
最大値	メモリ・サイズに依存
デフォルト値	SemMNI+2

整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

各識別子に割り当てられる個々のセマフォ・セットは、sem 配列内の 1 つまたは複数の連続スロットを占有します。割り当てられたセマフォが解除される都度、sem 配列はフラグメント化 (断片化) されていきます。

semmap は、sem 配列内の空き領域を示すリソース・マップのサイズを定義します。このマップ内のエントリは、連続した未割り当てスロットのセットを指しています。各エントリは、セットへのポインタと、セットのサイズを示す情報から構成されています。

セマフォ使用率が非常に高く、セマフォ・セットに対する要求に対応できない場合は、以下のメッセージが返されます。

```
danger: mfree map overflow
```

この場合は、semmap により大きい値を指定して、新しいカーネルを構成してください。

各セマフォ識別子が持つセマフォ数をすべて同一にすると、sem 配列のフラグメント化が軽減されるため、semmap により小さい値を指定できます。

最小値は 4 です。マップ用のオーバーヘッドとして 1 つ、システム初期化時に sem 配列が空いていることを示すためにもう 1 つのスロットが常に必要になります。

関連するパラメータ

- (semmap-2) が、sem 配列の連続した未割り当て領域の最大数になります。
- semmap には、(semmni + 2) 以下の値を指定します。
- semmap にこれより大きい値を指定すると、割り当てスペースの一部が無駄になります。

付加情報

[セマフォ・パラメータの概要](#)

[IPC セマフォに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

semmni

semmni は、システム上に同時に存在できる IPC セマフォ・セットの最大数を指定します。

指定可能な値:

最小値	2
最大値	メモリ・サイズに依存
デフォルト値	64

整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

semmni は、システム・ユーザーが使用できるセマフォのセット (識別子) の最大数を定義します。

セマフォ・セットを使い切った場合は、semget()システム・コールから ENOSPC エラー・メッセージが返されます。

関連するパラメータ

- semmni には、semmns 以下の値を指定します。
- semmns には、(semmni * semmsl) 以下の値を指定します。
- semmap には、(semmni + 2) 以下の値を指定します。

semmsl は、1 つのセマフォ ID に関連付けられるセマフォの最大数を示します。semmsl は 500 に設定されており、構成は不可能です。

付加情報

[セマフォ・パラメータの概要](#)

[IPC セマフォに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

semmns

semmns は、ユーザーに割り当てられる個々の IPC セマフォの、システム全体での最大数を定義します。

指定可能な値:

最小値	2
最大値	メモリ・サイズに依存
デフォルト値	128

整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

semmns は、システム・ユーザーが使用できる個々のセマフォの、システム全体での合計最大数を定義します。

空きスペース・マップにより、共有メモリのセマフォ領域に semget() 要求を満たすだけの連続セマフォ・スロットがないことが判明した場合は、semget() から ENOSPC エラーが返されます。十分な空きセマフォ・スロットはあっても、これらが連続していなければ、このエラーが返されます。

関連するパラメータ

- [semmni](#) には、semmns 以下の値を指定します。
- semmns には、(semmni * semmsl) 以下の値を指定します。

semmsl は、1 つのセマフォ ID に関連付けられるセマフォの最大数を示します。semmsl は 500 に設定されており、構成は不可能です。

付加情報

[セマフォ・パラメータの概要](#)

[IPC セマフォに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

semmnu

semmnu は、システム上の任意の IPC セマフォに対してアンドゥ処理を保留できるプロセスの最大数を定義します。

指定可能な値:

最小値	1
最大値	nproc-4
デフォルト値	30

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

アンドゥはセマフォ処理における特別なオプションのフラグであり、起動元プロセスが中止されたような場合に、処理内容を元に戻すために使われます。セマフォ・アンドゥ処理の詳細は、HP-UX Reference の semop(2) を参照してください。

semmnu は、任意のセマフォに対してアンドゥ処理を保留できるプロセスの最大数を指定するものであり、sem_undo 構造体のサイズを定義します。

この制限値を超えた場合は、SEM_UNDO フラグを使用する semop()システム・コールから ENOSPC エラーが返されます。

関連するパラメータ

- semmnu は sem_undo 構造体のサイズを定義します。この構造体には、semume でサイズ指定されるサブ構造体が含まれています。
- semmnu に (nproc -4) に等しい値を設定しても、この値はシステム上でセマフォを同時に使用できるプロセスの最大数であるため、意味がありません。

付加情報

[セマフォ・パラメータの概要](#)

[IPC セマフォに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

semume

semume は、任意のプロセスがアンドゥ処理を保留できる IPC セマフォの最大数を定義します。

指定可能な値:

最小値	1
最大値	SemMNS
デフォルト値	10

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

アンドゥはセマフォ処理における特別なオプションのフラグであり、起動元プロセスが中止されたような場合に、処理内容を元に戻すために使われます。セマフォ・アンドゥ処理の詳細は、HP-UX Reference の `semop(2)` を参照してください。

`semume` は、任意のプロセスがアンドゥ処理を保留できるセマフォの最大数を指定します。

SEMOP は、1 回のシステム・コールで変更できるセマフォの最大数です。この値は、`/usr/include/sys/sem.h` ファイル内に指定されています。

`semume` の値を超えた場合は、SEM_UNDO フラグを使用する `semop()` システム・コールから EINVAL エラーが返されます。

関連するパラメータ

- `semume` には、`semmns` 以下の値を指定します。
- `semume` は、`sem_undo` 構造体に含まれる `undo` サブ構造体のサイズを指定します。`sem_undo` のサイズは、`semmnu` で定義されています。

付加情報

[セマフォ・パラメータの概要](#)

[IPC セマフォに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

semvmx

`semvmx` は、使用可能な最大セマフォ値を指定します。

指定可能な値:

最小値	1
最大値	65535
デフォルト値	32767

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

`semvmx` は、セマフォの最大値を指定します。検出不能なセマフォ・オーバーフローの発生を防止するために、この値は 16 ビットの符号なし整数の最大値 (65535) 以下でなければなりません。

semop()システム・コールで semvmx の値を超えてセマフォ値を増分しようとする、ERANGE エラーが返されます。semvmx が 65535 を超えると、検出されない形でセマフォ値のオーバーフローが発生する可能性があります。

semopm は、1 回のシステム・コールで変更できるセマフォの最大数です。この値は、/usr/include/sys/sem.h ファイル内に指定されています。

関連するパラメータ

- [semaem](#) には、semvmx 以下の値を指定してください。

付加情報

[セマフォ・パラメータの概要](#)

[IPC セマフォに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

共有メモリの処理の概要

共有メモリ

共有メモリは、互いに連携して動作する複数のプロセスの間で共有されるデータ構造およびデータを記憶するための予約メモリ・スペースです。同じメモリ・スペースを共有すれば、あるプロセスが使用したデータをほかのプロセスが使用するときにデータを別の場所にコピーしたり移動する必要がなくなるので、メモリ消費、プロセッサ時間、およびオーバーヘッドを抑えることができます。

共有メモリのアクセスと使用

共有メモリの管理は、メッセージの管理およびセマフォの管理と多くの点で共通しています。プロセスが共有メモリ・セグメントの割り当てを要求するときには、shmget()システム・コールが使用され、この関数のパラメータの一つを経由してセグメント・サイズが指定されます。(HP-UX Reference の shmget(2) を参照)。いったん割り当てられたセグメントには、1 つまたは複数のプロセスを結び付けることができます。プロセスを結び付けるには、shmat()システム・コールを使います。また、結び付けられたプロセスは、終了時に shmdt()システム・コールによって切り離すことができます (shmop(2) を参照)。shmctl()を使用すると、セグメントの情報を取得したり、不要になったセグメントを削除することができます (shmctl(2) を参照)。

セマフォを使用すると、共有メモリへの読み取り/書き込みアクセスが衝突するのを防止できます。しかし、共有メモリ・セグメントへの書き込みを 1 つのプロセスで行い、そのセグメントからの読み取りを 1 つ以上のプロセスで行う方式 (write once/read many 方式) の方がよく使われます。この方式では、新たな書き込みを行うときには、書き込みプロセスが新しいセグメントを割り当てます。そして、新しいセグメントが利用可能になった時点で、読み取りプロセスがそのセグメントに結び付けられます。

共有メモリの管理

共有メモリは、スワップ可能な共有メモリ・スペース内で割り当てられます。共有メモリ管理用のデータ構造は、カーネル内に格納されます。共有メモリの制御に関するカーネル構成パラメータには、以下のパラメータがあります。

- 共有メモリ・セグメントの最大許容サイズ。
- システム全体において同時に割り当てられる共有メモリ・セグメントの最大許容数。
- 単一のプロセスに結び付けられる共有メモリ・セグメントの最大許容数。

付加情報

[共有メモリに関するカーネル・パラメータの説明一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

IPC 共有メモリに関する構成可能パラメータ

共有メモリのスペース割り当てを制御するパラメータは、以下に示すとおりです。

shmem

(ワークステーションのみ)。システム・ブート時に共有メモリを使用可能/使用不可能にします。

shmmax

共有メモリ・セグメントの最大許容サイズ (バイト単位)。

shmmni

システム上に同時に存在する共有メモリ・セグメントの最大許容数。

shmseg

1つのプロセスに同時にアタッチされる共有メモリ・セグメントの最大許容数。

付加情報

[セマフォ・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

shmem

shmem はシステム・ブート時に、カーネル内での System V IPC 共有メモリのサポートを有効/無効にします (ワークステーションのみ)。

指定可能な値:

最小値	0 (カーネルから System V IPC 共有メモリ・コードを除く)
最大値	1 (カーネルに System V IPC 共有メモリ・コードを含める)
デフォルト値	1

このパラメータには、整数値 0 または 1 を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

shmем は、システム・ブート時に System V IPC 共有メモリ用のコードをカーネルに含めるかどうかを定義します (ワークステーション・システムのみ)。

shmем = 1 カーネル内にコードが含まれます (IPC 共有メモリを使用可能)。

shmем = 0 カーネル内にコードが含まれません (IPC 共有メモリは使用不可能)。

サーバー・システム: IPC 共有メモリは、カーネル内で常に使用可能です。

ワークステーション・システム: shmем を 0 に設定した場合は、その他の IPC 共有メモリ・パラメータはすべて無視されます。

共有メモリを使用不可能にしてもよい場合

Starbase グラフィックスなどの一部のサブシステムは、共有メモリを必要とします。また X ウィンドウなどのように、共有メモリが使える場合はサーバーとクライアント間の通信に (通常大量の) 共有メモリを使用し、使用できない場合はソケットを使用するものもあります。メモリ・スペースが不足気味で、共有メモリがなくても (性能は低下するにしても) アプリケーションの実行に問題がない場合は、共有メモリを使用不可能にしても構いません。

関連するパラメータ

なし

付加情報

[共有メモリに関するカーネル・パラメータの説明一覧](#)

[IPC 共有メモリに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

shmmax

システム全体で適用される共有メモリ・セグメントの最大サイズを指定します。

指定可能な値:

最小値	2048 (2 KB)
最大値	1 GB (32 ビット・システムの場合)
デフォルト値	0x04000000 (64 MB)

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

shmmax は、システム全体で適用される共有メモリ・セグメントの最大サイズをバイト単位で指定します。shmget()システム・コールでこの値より大きいセグメントを要求した場合は、エラーが返されます (shmget(2) を参照)。

指定された値は、スワップスペースで利用可能な最大値を超えることはできません。指定可能な最小値と最大値、およびそのシステムのデフォルト値については、`/etc/conf/master.d/*` ファイルを参照してください。

shmmax は、カーネルを再構築するか、または実行中のカーネル内で `settune()` を使用することにより設定できます。SAM および `kmtune` は `settune()` を使用します。shmmax は、新しい共有メモリ・セグメントの作成時にのみチェックされます。このパラメータに対する動的な変更は、`settune()` コール以降に作成される共有メモリ・セグメントのサイズにのみ影響を与えます。

関連するパラメータ

なし

付加情報

[共有メモリの処理の概要](#)

[IPC 共有メモリに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

shmmni

shmmni は、システム全体で使用できる共有メモリ・セグメントの最大数を指定します (セグメント識別子の数を制限)。

指定可能な値:

最小値	3
最大値	メモリ・サイズに依存
デフォルト値	200

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

shmmni は、システム全体で同時に存在できる共有メモリ・セグメントの最大数を指定します。shmmni で指定した数のセグメントがすでに存在している状態で、shmget()システム・コールから新しいセグメントを要求した場合は、エラーが返されます (HP-UX Reference の shmget(2) を参照)。このパラメータは、スワップ禁止のカーネル・スペースに保存される共有メモリ・セグメント識別子リストのエントリ数を制限します。識別子に割り当てられるスペースは、shmmni × 104 バイトになります。

shmmni に指定した値が大きすぎると、メモリが浪費され、システムの処理性能が低下する可能性があります。メモリ量の少ないシステムでこの値を大きく設定しすぎると、メモリが消費されてしまいシステムがブートできなくなる可能性もあります。メモリ使用率を最適化するには、実際のシステム要件にできるだけ近い値を選択することが大切です。システム要件で特に必要となる場合を除き、通常は 1024 以下の値を指定してください。

Starbase グラフィックスを使うには、shmmni に 4 以上の値を設定する必要があります。

関連するパラメータ

なし

付加情報

[共有メモリの処理の概要](#)

[IPC 共有メモリに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

shmseg

1 つのプロセスに同時にアタッチできる共有メモリ・セグメントの最大数を定義します。

指定可能な値:

最小値	1
最大値	shmmni
デフォルト値	120

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

shmseg は、1 つのプロセスに同時にアタッチできる共有メモリ・セグメントの最大数を指定します。shmat()に対する呼び出しでこの制限値を超えた場合は、エラーが返されます (shmop(2) を参照)。

shmseg は、カーネルを再構築するか、または実行中のカーネル内で settune() を使用することにより設定できます。SAM および kmtune は settune() を使用します。shmseg は、セグメントがプロセスにアタッチされるたびに、shmat() 内でのみチェックされます。このパラメータに対する動的な変更は、shmat() に対する以降の呼び出しにのみ影響を与えます。既存の共有メモリ・セグメントへの影響はありません。

関連するパラメータ

なし

付加情報

[共有メモリの処理の概要](#)

[IPC 共有メモリに関する構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

VME I/O サブシステムに関するパラメータ(オプション製品)

ここでは、VME I/O サブシステムに関連する構成可能カーネル・パラメータのリストを示します。各パラメータの詳細は、次の一覧から目的のパラメータを選択し、説明を参照してください。また、これらのパラメータの適切な設定値の詳細は、VME のドキュメントを参照してください。

[vmebpn_public_pages](#)

[vmebpn_sockets](#)

[vmebpn_tcp_ip](#)

[vmebpn_tcp_ip_mtu](#)

[vmebpn_total_jobs](#)

[vme_io_estimate](#)

付加情報

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vmebpn_public_pages

vmebpn_public_pages は、VME スレーブ I/O メモリ・マッパー用に確保しておく 4 KB ページの数を指定します。

指定可能な値:

最小値	0
最大値	32
デフォルト値	1

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

vmebpn_public_pages は、VME スレーブ I/O メモリ・マップ用確保しておく 4 KB ページの数を指定します。

VME I/O サブシステム(オプション製品)に関するカーネル・パラメータを設定する方法の詳細は、VME ドキュメントを参照してください。

関連するパラメータ

なし

付加情報

[VME に関するカーネル・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vmebpn_sockets

vmebpn_sockets は、VME ソケット・ドメイン AF_VME_LINK をアクティブにするかどうかを指定します。

指定可能な値:

最小値	0 (AF_VME_LINK は非アクティブ)
最大値	1 (AF_VME_LINK はアクティブ)
デフォルト値	1 (AF_VME_LINK はアクティブ)

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

vmebpn_sockets は、VME ソケット・ドメイン AF_VME_LINK を使用可能/使用不可能にします。

VME I/O サブシステム(オプション製品)に関するカーネル・パラメータを設定する方法の詳細は、VME ドキュメントを参照してください。

付加情報

[VME に関するカーネル・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vmebpn_tcp_ip

vmebpn_tcp_ip は、システム上で使用できる DLPI PPA の最大数を指定します。

指定可能な値:

最小値	0
最大値	1
デフォルト値	1

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

vmebpn_tcp_ip は、システム上で使用できる DLPI PPA の最大数を指定します。0 に設定した場合は、TCP-IP は使用できません。

最大値は 1 です。

VME I/O サブシステム(オプション製品)に関するカーネル・パラメータを設定する方法の詳細は、VME ドキュメントを参照してください。

付加情報

[VME に関するカーネル・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vmebpn_tcp_ip_mtu

vmebpn_tcp_ip_mtu は、PPA 転送単位の最大許容サイズを KB 単位で指定します。

指定可能な値:

最小値	0
最大値	64
デフォルト値	8

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

vmebpn_tcp_ip_mtu は、PPA 転送単位の最大許容サイズを KB 単位で指定します。

VME I/O サブシステム(オプション製品)に関するカーネル・パラメータを設定する方法の詳細は、VME ドキュメントを参照してください。

付加情報

[VME に関するカーネル・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vmebpn_total_jobs

vmebpn_total_jobs は、システム全体で同時にオープンできる VME ポートの最大数を指定します。

指定可能な値:

最小値	0
最大値	8096
デフォルト値	16

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

vmebpn_total_jobs は、システム全体で同時にオープンできる VME ポートの最大数を指定します。

VME I/O サブシステム(オプション製品)に関するカーネル・パラメータを設定する方法の詳細は、VME ドキュメントを参照してください。

付加情報

[VME に関するカーネル・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

vme_io_estimate

vme_io_estimate は、カーネル I/O スペース内で VME サブシステム用に割り当てられる 4 KB ページの数を指定します。

指定可能な値:

最小値	0
最大値	0x800
デフォルト値	0x800

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

vme_io_estimate は、カーネル I/O スペース内で VME サブシステム用に割り当てられる 4 KB ページの数を指定します。

VME I/O サブシステム(オプション製品)に関するカーネル・パラメータを設定する方法の詳細は、VME ドキュメントを参照してください。

付加情報

[VME に関するカーネル・パラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

その他の各種パラメータの概要

ここでは、通常は互いに関連性のない各種の HP-UX サブシステムに影響を及ぼす構成可能パラメータについて説明します。これらのパラメータの一覧は、[その他の各種構成可能パラメータ](#)を参照してください。

非同期ディスク I/O ドライバ

[max_async_ports](#)

システム全体において、プロセスが非同期ディスク I/O ドライバにアクセスするためにオープンするポートの最大許容数を指定します。

システム・クロック

システム・クロックに関するパラメータには、[dst](#) と [timezone](#) の 2 つがあります。dst は、夏時間を使用する場合に、夏時間のスタイルを指定します。timezone は、ローカル・タイム・ゾーンと世界協定時刻 (UTC) の間の時差を指定します。UTC は、グリニッチ標準時刻 (GMT) とも呼ばれます。

高速シンボリック・リンク・トラバーサル

create_fastlinks

パス名の検索中でのディスクへのアクセス頻度を減らすため、効率的な形でシンボリック・リンクを作成するようにシステムに指示するバイナリ変数です。

ディスク I/O 用の関数の制御

default_disk_ir

write()でデータを書き込んだときに、データがディスクの書き込みバッファに格納された時点で制御を戻すか、それともデータがディスク・メディア上に物理的に格納されるまで待機するかを決定します。

o_sync_is_o_dsync

O_SYNC フラグを指定して open()または fcntl() を呼び出す場合、O_DSYNC フラグを指定したのと同じ処理をするかどうかを決定します。このパラメータは、ファイル・データをディスクに書き込んだ後、ファイルのアクセス時刻に関する属性を更新する前に、関数の呼び出し元に戻るかどうかを制御します。

カーネル・パニック・ダンプ

alwaydump

カーネル・パニックの発生時にどのクラスのシステム・メモリをディスクにダンプするかを指定します。

dontdump

カーネル・パニックの発生時にどのクラスのシステム・メモリをディスクへのダンプから除外するかを指定します。

initmodmax

カーネル・パニック (システム・クラッシュ) ダンプ処理によって保存されるカーネル・モジュールの最大数を指定します。

マルチ・プロセッサ・スピンロック

これらのパラメータは、複数のプロセッサを搭載したシステムで使用するスピンロックの割り当てを制御します。これらのパラメータは、十分な知識のあるユーザーを想定して用意されています。当社が認定したフィールド・サポート・スタッフから直接の指示がない限り、これらのパラメータを変更することは避けてください。これらのパラメータに不適切な値を設定すると、システムの性能が大幅に低下する可能性があります。

bufcache_hash_locks	バッファ・キャッシュ・スピンロック・プール
chanq_hash_locks	チャンネル待ち行列スピンロック・プール
ftable_hash_locks	ファイル・テーブル・スピンロック・プール
io_ports_hash_locks	I/O ポート・スピンロック・プール
pfdat_hash_locks	Pfdat スピンロック・プール
region_hash_locks	プロセス・リージョン・スピンロック・プール
sysv_hash_locks	System V プロセス間通信スピンロック・プール
vnode_cd_hash_locks	vnode クリーン/ダーティ・スピンロック・プール
vnode_hash_locks	vnode スピンロック・プール

詳細は、[スピンロック・パラメータ](#) を参照してください。

分散サーバー・メモリ・マップ I/O

[clireservedmem](#)

高速分散サーバー環境においてユーザー・プロセスによる I/O マップ用に予約するシステム・メモリのバイト数を指定します。例えば、大規模なデータベース処理プロセスなどに使用されます。

等価マップ・メモリ

[eqmemsize](#)

等価マップ・メモリとして使用できるように予約するメモリ・ページの数を指定します。主に、DMA 転送に使用されます。

システム上のユーザーの最大数

[maxusers](#)

システムにログインするユーザーの最大想定数を指定します。この値は、ほかのシステム・パラメータ定義でシステム・リソースを割り当てるときの基準となります。

待ち行列シグナル

待ち行列シグナルの動作を制御するパラメータには、[ksi_alloc_max](#) と [ksi_send_max](#) の 2 つがあります。[ksi_alloc_max](#) は、システム全体において割り当て可能な待ち行列シグナルの最大数を指定します。[ksi_send_max](#) は、プロセスが 1 つまたは複数の受信プロセスに送信して保留にできる待ち行列シグナルの最大数を指定します。

カーネル・タイムアウト・スケジュールの制限

[ncallout](#)

カーネルがスケジュールするタイムアウトの最大許容数を指定します。

telnet セッションに関するパラメータ

[nstrtel](#)

システム上で利用できる telnet セッション・デバイス・ファイルの数を指定します。

Web サーバーに関するパラメータ

[sendfile_max](#)

HP-UX Web サーバー上の `sendfile()` システム・コールが排他的に使用するバッファ・キャッシュの最大サイズを指定します。

CD-ROM ファイル・システムの vnode テーブルに関する制限

[ncdnode](#)

CD-ROM ファイル・システムの vnode テーブルのエントリの最大数を指定します。これにより、CD-ROM ファイル・システム・ノードをメモリ内に何個まで開くことができるかが決まります。

tty 転送および pty 転送のブロック数

nclist

tty デバイスおよび pty デバイスによるデータ転送用に割り当てる cblock の最大許容数を指定します。

デバイス I/O ライブラリ・バッファ

ndilbuffers

同時にオープンされるデバイス I/O ライブラリ・デバイスの最大許容数を指定します。

pty エントリの数

npty

システム上に存在する擬似 tty エントリの最大許容数を指定します。

電源異常処理に関するパラメータ

pfail_enabled

AC 電源の損失が発生したときに電源異常ルーチンを使用するかどうかを指定します。バックアップ電源ハードウェアを備えたサーバー・システムの場合にのみ適用されます。

共有ライブラリ上のパブリック保護 ID に関するパラメータ

public_shlibs

システム・性能を向上するために共有ライブラリ上でパブリック保護 ID を使用できるようにするかどうかを指定します。

リアルタイム優先順位レベル

rtsched_numpri

システム上で利用できる POSIX リアルタイム・プロセスの優先順位として指定できる数です。

ITE コンソールのスクロール・サイズ

scroll_lines

内部端末エミュレータ (ITE) システム・コンソール上でスクロールする行数を定義します。

システム・メモリの予約領域

unlockable_mem

システム・オーバーヘッドおよび仮想メモリの管理用に予約するシステム・メモリ (ユーザー・プロセスでロックできないメモリ) の量を指定します。

付加情報

その他の各種構成可能パラメータの一覧

構成可能カーネル・パラメータ・ヘルプのブラウズ

その他の各種構成可能パラメータ

ここでは、通常は互いに関連性のない各種の HP-UX サブシステムに影響を及ぼす構成可能パラメータについて説明します。

ディスク I/O

default_disk_ir

write()でデータを書き込んだときに、データがディスクの書き込みバッファに格納された時点で制御を戻すか、それともデータがディスク・メディア上に物理的に格納されるまで待機するかを指定します。

o_sync_is_o_dsync

open()またはfcntl()に O_SYNC フラグが設定されている場合でも、O_DSYNC フラグを指定したのと同じ処理をするかどうかを指定します。これにより、ファイルのアクセス時刻に関する属性を更新する前に、関数の呼び出し元に戻るかどうかを制御します。

カーネル・パニック・ダンプ

alwaydump

カーネル・パニックの発生時にどのクラスのシステム・メモリをダンプするかを指定します。

dontdump

カーネル・パニックの発生時にどのクラスのシステム・メモリをダンプから除外するかを指定します。

initmodmax

カーネル・パニック (システム・クラッシュ) ダンプ処理によって保存されるカーネル・モジュールの最大数。

マルチ・プロセッサ・スピンロック

これらのパラメータは、複数のプロセッサを搭載したシステムで使用するスピンロックの割り当てを制御します。

bufcache_hash_locks

バッファ・キャッシュ・スピンロック・プール

chanq_hash_locks

チャンネル待ち行列スピンロック・プール

ftable_hash_locks

ファイル・テーブル・スピンロック・プール

io_ports_hash_locks

I/O ポート・スピンロック・プール

pfdat_hash_locks

Pfdat スピンロック・プール

region_hash_locks

プロセス・リージョン・スピンロック・プール

sysv_hash_locks

System V プロセス間通信スピンロック・プール
vnode_cd_hash_locks
vnode クリーン/ダーティ・スピンロック・プール
vnode_hash_locks
vnode スピンロック・プール

詳細は、[スピンロック・パラメータ](#) を参照してください。

分散サーバー・メモリ・マップ I/O

[clircreservedmem](#)

高速分散サーバー環境においてユーザー・プロセスによる I/O マップ用に予約するシステム・メモリのバイト数を指定します。例えば、大規模なデータベース処理プロセスなどに使用されます。

等価マップ・メモリ

[eqmemsize](#)

等価マップ・メモリとして使用できるように予約するメモリ・ページの数指定します。主に、DMA 転送に使用されます。

システム上のユーザーの最大数

[maxusers](#)

システムにログインするユーザーの最大想定数を指定します。この値は、ほかのシステム・パラメータ定義でシステム・リソースを割り当てるときの基準となります。

待ち行列シグナル

待ち行列シグナルの動作を制御するパラメータには、[ksi_alloc_max](#) と [ksi_send_max](#) の 2 つがあります。ksi_alloc_max は、システム全体において割り当て可能な待ち行列シグナルの最大数を指定します。ksi_send_max は、プロセスが 1 つまたは複数の受信プロセスに送信して保留にできる待ち行列シグナルの最大数を指定します。

カーネル・タイムアウト・スケジュールの制限

[ncallout](#)

カーネルがスケジュールするタイムアウトの最大許容数を指定します。

telnet セッションに関するパラメータ

[nstrtel](#)

システム上で利用できる telnet セッション・デバイス・ファイルの数を指定します。

Web サーバーに関するパラメータ

[sendfile_max](#)

HP-UX Web サーバー上の sendfile() システム・コールが排他的に使用するバッファ・キャッシュの最大サイズを指定します。

CD-ROM ファイル・システムの vnode テーブルに関する制限

[ncdnode](#)

HP-UX 11i 構成可能カーネル・パラメータ (HP-UX 11.0 用)

CD-ROM ファイル・システムの vnode テーブルのエントリの最大数を指定します。これにより、CD-ROM ファイル・システム・ノードをメモリ内に何個まで開くことができるかが決まります。

tty 転送および pty 転送のブロック数

nclist

tty デバイスおよび pty デバイスによるデータ転送用に割り当てる cblock の最大許容数を指定します。

デバイス I/O ライブラリ・バッファ

ndilbuffers

同時にオープンされるデバイス I/O ライブラリ・デバイスの最大許容数を指定します。

pty エントリの数

npty

システム上に存在する擬似 tty エントリの最大許容数を指定します。

電源異常処理に関するパラメータ

pfail_enabled

AC 電源の損失が発生したときに電源異常ルーチンを使用するかどうかを指定します。バックアップ電源ハードウェアを備えたサーバー・システムの場合にのみ適用されます。

共有ライブラリ上のパブリック保護 ID に関するパラメータ

public_shlibs

システム・性能を向上するために共有ライブラリ上でパブリック保護 ID を使用できるようにするかどうかを指定します。

リアルタイム優先順位レベル

rtsched_numpri

システム上で利用できる POSIX リアルタイム・プロセスの優先順位として指定できる数です。

ITE コンソールのスクロール・サイズ

scroll_lines

内部端末エミュレータ (ITE) システム・コンソール上でスクロールする行数を定義します。

システム・メモリの予約領域

unlockable_mem

システム・オーバーヘッドおよび仮想メモリの管理用に予約するシステム・メモリ (ユーザー・プロセスでロックできないメモリ) の量を指定します。

システム・メモリの予約領域

その他の各種構成可能パラメータの一覧

構成可能カーネル・パラメータ・ヘルプのブラウズ

clicreservedmem

clicreservedmem は、高速分散サーバー環境においてユーザー・プロセスによる I/O マップ用に予約するシステム・メモリをバイト単位で指定します。例えば、大規模なデータベース処理プログラムなどに使用されます。

指定可能な値:

これらのパラメータの下限値と上限値はすべて同一です。

以下に示すように、デフォルト値のみが異なります。

最小値	0
最大値	なし
デフォルト値	0

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

通常の HP-UX システムは、I/O マップ用に比較的少ない容量のシステム・メモリを予約します。ただし、一部の特殊アプリケーション(大規模なデータベース処理プログラムなど)はしばしば、高速高帯域通信ネットワークにより相互接続されている高速サーバーで構成されるクラスター上で実行されます。これらのアプリケーションは大量のシステム・リソースが必要とするため、多くの場合メモリ・マップ I/O を使って通信を行います。この場合、アプリケーションおよび対応する I/O またはネットワーク・ソフトウェアによって大容量のメモリ・ブロックが共有されます。

構成可能なパラメータ clicreservedmem により、大容量のネットワーク用メモリ・マップ I/O を実行するアプリケーション用に、全システム・メモリの 16 分の 15(約 93%)を確保しておくことができます。このパラメータ値を 512 GB または 1 TB に設定することもできますが、通常は約 1 GB から 64 GB の範囲で設定します。SAM で選択した値と関係なく、システムが予約する実際のメモリを全システム・メモリの 16 分の 15 を超えて設定することはできません。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

create_fastlinks

create_fastlinks は、システムが高速シンボリック・リンクを使用するよう構成します。

指定可能な値:

最小値	0
最大値	1
デフォルト値	0

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

create_fastlinks が 0 でない値の場合、システムは、パス名のルックアップ時に各シンボリック・リンクに対してディスク・ブロックへのアクセス回数を 1 回ずつ減らすように、HFS シンボリック・リンクを作成します。このとき、HFS ディスク・フォーマットが一部変更されるため、高速シンボリック・リンク用にフォーマットされたディスクはすべて、HP-UX リリース 9.0 以前のワークステーション・システムまたは HP-UX リリース 10.0 以前のサーバー・システムで使用できなくなります(この構成可能パラメータは HP-UX リリース 9.0 のワークステーション・システムには存在しますが、HP-UX リリース 9.0 のサーバー・システムには存在しません)。

以前のバージョンとの互換性を提供するため、create_fastlinks のデフォルト値は 0 に設定されており、より高速の新しいフォーマットが作成されないようになっています。ただし、すべての HP-UX 10.0 カーネル(およびすべてのワークステーションの HP-UX 9.0 カーネル)では、create_fastlinks が 0 または 0 でない値のどちらに設定されている場合でも、両方のディスク・フォーマットを認識します。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

default_disk_ir

default_disk_ir は、ディスク I/O 時にすぐに制御を戻すかどうかを指定します。

指定可能な値:

最小値	0(オフ)
最大値	1(オン)
デフォルト値	0(オフ)

0 (すぐに制御を戻すことを不可能にする)、または 1 (すぐに制御を戻すことを可能にする)に設定します。

説明

default_disk_ir は、ディスク I/O 時にすぐに制御を戻すかどうかを指定します。

すぐに制御を戻すことを可能に設定すると、データ・キャッシュを持つディスク・ドライブは、メディアへのデータの書き込みが終了した後ではなく、データがキャッシュに入るとすぐ write() システム・コールから戻ります。これにより、場合によっては書き込み性能が向上します(特に順次転送の場合)。ただし、デバイスがキャッシュに入ったデータをメディアに書き込む前にデバイスで電源障害が発生したりリセットが行われた場合、キャッシュに入ったデータが失われるおそれがあります。サーバー・システムの場合、このパラメータを 0(オフ)に設定することをお勧めします。

構成可能なパラメータ default_disk_ir は、mount コマンドのオプションではありませんが(HP-UX Reference の mount(1m)を参照)、ファイル・システムと(raw)ディスクの性能、逆に言えばシステム・リセット発生時のデータの完全性に大きな影響を与えます。このパラメータはオン(1)またはオフ(0)のどちらかに設定します。

カーネル(/stand/system)の作成に使用するカーネル構成ファイルからこの構成可能パラメータを削除した場合、このパラメータ値はオフ(0)とみなされます。したがって、すぐに制御を戻す動作 (Write Cache Enable (WCE)とも呼ばれる)のデフォルトの設定はオフ(不可能)になります。

default_disk_ir は、delayed-write(遅延書き込み) やそれに対比される write-through(ライト・スルー)といったファイルシステムの振る舞いにも影響を与えます。

注記: HP-UX リリース 10.0 またはそれ以降が動作するサーバー・システムにマウントされているディスクは、電源投入時のデフォルト値として WCE=off (キャッシュへの書き込み=オフ)に設定されていることが必要です。ディスクを後で 10.0 以前のワークステーション・システムに移動した場合、ディスクは、キャッシュへの書き込みが不可能な状態で引き続き動作します。必要に応じてすぐに制御を戻す動作を再度使用可能にするには、scsictl コマンドを使用します(HP-UX Reference の scsictl(1M)を参照)。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

dst

dst は、夏時間への変換を可能/不可能にし、変換スケジュールを指定します。

指定可能な値:

以下の整数値のいずれかを指定してください。

- 0 夏時間を使用不可能にします。
- 1 夏時間を米国形式に設定します(デフォルト)。

- 2 夏時間をオーストラリア形式に設定します。
- 3 夏時間を西ヨーロッパ形式に設定します。
- 4 夏時間を中央ヨーロッパ形式に設定します。
- 5 夏時間を東ヨーロッパ形式に設定します。

値の入力方法については、[パラメータ値の指定](#) を参照してください。

説明

dst は、夏時間に変換するかどうか、また夏時間と標準時間の変換にどのスケジュールを使用するかを指定します。

このパラメータを 0 に設定すると、夏時間への変換が使用不可能になります。0 でない値に設定すると、夏時間への変換が可能になり、ファイル `usr/include/sys/time.h` での以下の定義に従って変換スケジュールを選択できます。

```
#define DST_NONE    0          /* not on dst */          (夏時間使用不可能)
#define DST_USA     1          /* USA style dst */      (米国形式)
#define DST_AUST    2          /* Australian style dst */ (オーストラリア形式)
#define DST_WET     3          /* Western European dst */ (西ヨーロッパ形式)
#define DST_MET     4          /* Middle European dst */ (中央ヨーロッパ形式)
#define DST_EET     5          /* Eastern European dst */ (東ヨーロッパ形式)
```

関連するパラメータ

timezone

dst と timezone は共に、システム・クロック・ルーチンが使用するパラメータで、環境変数 TZ との整合性が維持されていることが必要です。詳しくは、HP-UX Reference の `environ(5)` と `login(1)` を参照してください。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

eqmemsize

等価マップ・メモリ予約プールのサイズをページ数で指定します。

指定可能な値:

最小値 0
 最大値 メモリ・サイズに依存
 デフォルト値 15 ページ

このパラメータには、整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

eqmemsize では、実モードと仮想モードのアドレッシングが同じ場合に割り当てメモリを必要とするドライバとサブシステムが使用するスペースとして、予約するメモリ・スペースの最小値をページ数で指定します。システムはブート時に、インストールされている物理メモリ量に基づいて、実際に予約されているスペースの量を増やす場合があります。

ドライバは、このページを使ってハードウェア・インタフェースとの間で情報を転送します。ドライバは、仮想モードを使ってページ上にデータを配置します。

次に、ハードウェアは実モードでデータを DMA (Direct Memory Access (メモリへの直接アクセス)) 転送し、アドレッシング変換を回避します。仮想アドレスと実アドレスは同一のため、特別なアドレス処理は必要ありません。またこのスペースは、EXEC_MAGIC プロセスに代わって発行されるアドレス・エイリアシング要求をサポートする際にも使用されます。

システムは通常、システム全体のメモリ・プールから一致する仮想アドレスのある空きページを取得することにより、等価マップ・メモリに対する要求を動的に処理します。システムが等価マップ・メモリを動的に取得できない場合は、最終手段としてシステムの予約プールを使用します。システムは通常、等価マップ・ページを動的に割り当てることができるので、予約プールが使い果たされることはありません。ただし、負荷が比較的高く、物理メモリ構成が 1 GB を超えるシステムでは、この予約プールが使い果たされる可能性があります。

さらに、このシステム上で実行されるアプリケーションの性質やシステム負荷、メモリ、I/O 構成、その他の要因によっても、予約プールが使い果たされる可能性があります。このような状況が発生した場合、システムは、予約プールが使い果たされ、eqmemsize の値を増やす必要があることを示すメッセージをコンソールに表示します。

値を指定する

デフォルトの eqmemsize は、等価マップ・メモリ用のスペース不足によるシステム障害が発生しない限り、デフォルト値 15 より大きくしてはいけません。システム障害が発生して、値を大きくする必要が生じた場合は、システムの I/O 構成に基づいて値を大きくしてください。大規模な構成の場合は、小規模なシステムの場合より値をより大きく設定することが必要です。X.25 ユーザーの場合、システム内の X.25 カード 1 枚につき 1 ページ設定されていることが必要です。

等価マップ・メモリの作成

I/O サブシステムは、システムのブート・プロセス中に、I/O 構成に基づいて等価マップ・メモリ・ページのプールを構築します。

何らかの理由により要件が満たされない場合は、I/O サブシステムの他のオプションを使って、等価マップ・ページを取得できます。そのうちの 1 つのオプションとして、等価マップ・メモリ・プールを使用する方法があります。等価マップ・メモリ・プールのデフォルトのサイズは、eqmemsize のエントリで定義されています。現在のデフォルト値は 15 ページです。これらのページのいずれかを使用できますが、このページはまず既存の空きページを等価マップ・ページに変換しようとしています。マシンの規模や負荷レベルにより、変換が容易なページと容易でないページがあります。したがって、等価マップ・プールには eqmemsize で設定されているページ数が最低限含まれますが、変換によってより多くのスペースが確保される可能性があります。

等価マップ・メモリ・アロケータはページ変換を試みるので、通常システムの等価マップ・ページが使い果たされる可能性はありません。

特殊なシステムの場合

非常に大規模な I/O 構成または多数の X.25 カードを備えたシステムでは、eqmemsize の値を増やす必要がある場合があります。等価マップ・メモリ不足によるシステム・パニックが発生した場合は必ず、eqmemsize の値を増やしてください。ただし、予約プールはメモリ・ページが変換できない場合にしか使用されないため、このような障害が起こることはめったにありません。大規模なシステム、特に物理メモリが 1 GB を超えるシステムでは、等価マップ・メモリ不足による障害に遭遇する可能性が高くなります。

このプールに格納されているメモリ・ページは、等価マップ要求以外の目的に使用することはできません。つまり、これらのページを通常のシステム・プロセスに使用できないため、eqmemsize の値を、システム・プロセスの要件などを無視して大きく設定してはいけません。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

executable_stack

スタック領域でのプログラム実行を有効/無効にします (セキュリティ機能)。

指定可能な値:

最小値	0
最大値	2
デフォルト値	1

このパラメータには、整数値を指定してください。0 は無効、1 は有効、2 は警告付きで有効です。

パラメータの基礎知識については、[パラメータ値の指定](#)を参照してください。

説明

executable_stack は、通常試みられることが多いセキュリティ侵害に対する保護手段を提供します。このパラメータでは、システムのメモリマッピング・ハードウェアを使用するかどうかについて、システム全体のデフォルト値を設定できるため、最も一般的なセキュリティ侵害方法の 1 つである「スタック・バッファ・オーバーフロー攻撃」の防御に役立ちます。

executable_stack は、通常試みられることが多いセキュリティ侵害に対する保護手段を提供します。このパラメータでは、システムのメモリマッピング・ハードウェアを使用するかどうかについて、システム全体のデフォルト値を設定できるため、最も一般的なセキュリティ侵害方法の 1 つである「スタック・バッファ・オーバーフロー攻撃」の防御に役立ちます。

executable_stack は、通常試みられることが多いセキュリティ侵害に対する保護手段を提供します。このパラメータでは、システムのメモリマッピング・ハードウェアを使用するかどうかについて、システム全体のデフォルト値を設定できるため、最も一般的なセキュリティ侵害方法の 1 つである「スタック・バッファ・オーバーフロー攻撃」の防御に役立ちます。

このパラメータについては、`chattr(1)` マン・ページ内の '+es' オプションの説明も参照してください。このパラメータの影響、有効な設定値の意味、各自のシステム上で異なる設定が必要かどうかの判断方法、セキュリティと互換性の兼ね合いを考慮してシステム全体に対する設定とアプリケーションごとの設定を最適な形で組み合わせる方法などが示されています。

付加情報

[st_large_recs](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ksi_alloc_max

ksi_alloc_max は、システム全体において割り当て可能な待ち行列シグナルの最大数を指定します。

指定可能な値:

最小値	32
最大値	メモリ・サイズに依存
デフォルト値	nproc*8

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

カーネルは、`sigqueue()` システム・コールを使ってプロセスが送信する待ち行列シグナルを保持するのに必要な記憶スペースを割り当てます(HP-UX Reference の `sigqueue(2)` を参照)。割り当てる必要があるスペースの量を、`ksi_alloc_max` を使って決定します。通常のシステム動作時に、既存のプロセスにより送信される待ち行列シグナルと受信プロセス側で保留状態になっている待ち行列シグナルの合計によって使用可能なデータ構造のスペースがいっぱいになった場合は、新しい待ち行列シグナルは送信できません。

待ち行列シグナルは、従来の HP-UX/UNIX シグナルとは異なることに注意してください。従来のシグナル(抹消シグナルまたはハングアップ・シグナルなど)は、受信プロセスに送信されていました。複数の同一シグナルが 1 つのプロセスに送信された場合、プロセス側では複数のシグナルが送信されたことを判別できませんでした。これに対して待ち行列シグナルの場合、プロセス

は待ち行列シグナルを処理した後、再度待ち行列をチェックして、待ち行列に別のシグナルがないかを検索するので、このようなシグナルの判別に関する問題は発生しなくなりました。

`ksi_alloc_max` は、待ち行列シグナルの処理用にカーネル内で割り当てるデータ構造スペースの量を制御することにより、待ち行列に入れることのできるシステム全体の待ち行列シグナルの最大数を指定します。`/usr/include/limits.h` で定義されている `SIGQUEUE_MAX` と `_POSIX_SIGQUEUE_MAX` の値は、このパラメータ値の影響を受けます。

このパラメータのデフォルト値は `nproc*8` に設定されており、システムが最大能力で動作している場合、システム上で実行されている各プロセスに対して最大 8 個のシグナルを保留にすることができます。システムのソフトウェア要件で 9 個以上のシグナル数が要求されない限り、この値はほぼすべてのシステムに適しています。

関連するパラメータ

[ksi_send_max](#)

任意の 1 つのプロセスが送信可能で、かつ受信プロセス側で保留にできる待ち行列シグナルの最大数を設定します。これにより、カーネルのシグナル待ち行列用スペースが占有されて、シグナルを送信する必要のある他のプロセスに悪影響を及ぼすのを避けることができます。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ksi_send_max

`ksi_send_max` は、1 つのプロセスが 1 つまたは複数の受信プロセスに送信して保留にできる待ち行列シグナルの最大数を指定します。

指定可能な値:

最小値	32
最大値	メモリ・サイズに依存
デフォルト値	32

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

カーネルは、`sigqueue()` システム・コールを使ってプロセスが送信する待ち行列シグナルを保持するのに必要な記憶スペースを割り当てます(HP-UX Reference の `sigqueue(2)` を参照)。割り当てる必要があるスペースの量を、`ksi_alloc_max` を使って決定します。通常のシステム動作時に、既存のプロセスにより送信される待ち行列シグナルと受信プロセス側で保留状態になっている

待ち行列シグナルの合計によって使用可能なデータ構造のスペースがいっぱいになった場合は、新しい待ち行列シグナルは送信できません。

待ち行列シグナルは、従来の HP-UX/UNIX シグナルとは異なることに注意してください。従来のシグナル(抹消シグナルまたはハングアップ・シグナルなど)は、受信プロセスに送信されていました。複数の同一シグナルが 1 つのプロセスに送信された場合、プロセス側では複数のシグナルが送信されたことを判別できませんでした。これに対して待ち行列シグナルの場合、プロセスは待ち行列シグナルを処理した後、再度待ち行列をチェックして、待ち行列に別のシグナルがないかを検索するので、このようなシグナルの判別に関する問題は発生しなくなりました。

ksi_send_max は、1 つのプロセスが 1 つまたは複数の受信プロセスに送信して保留にできる待ち行列シグナルの最大数を指定します。このパラメータは、1 つのプロセスが大量のシグナルを発行することによりシグナルのデータ構造スペースを独占するのを防止する機構を備えています。これにより、他のプロセスがカーネルのリソース不足でシグナルを送受信できなくなるのを防ぐことができます。

デフォルト値 32 は、一般に使用されている大部分の HP-UX アプリケーションに適しています。この値より大きく設定する必要がある特殊なアプリケーションを使用している場合は(sigqueue()が EAGAIN を返す場合)、値を大きく設定して、エラーを回避してください。ただしこれは、sigqueue()が返す EAGAIN エラーが、不適切なタイミングでシグナルを生成する暴走プロセスによるものでない場合に限りです。

関連するパラメータ

ksi_alloc_max

待ち行列シグナルの処理用にカーネル内で割り当てるデータ構造スペースの量を制御することにより、待ち行列に入れることのできるシステム全体の待ち行列シグナルの最大数を指定します。/usr/include/limits.h で定義されている SIGQUEUE_MAX と _POSIX_SIGQUEUE_MAX の値は、このパラメータ値の影響を受けます。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

max_async_ports

max_async_ports は、システム全体において、プロセスが非同期ディスク I/O ドライバにアクセスするためにオープンするポートの最大数を指定します。

指定可能な値:

最小値 1

最大値

デフォルト値 50

このパラメータには、**整数値**を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

max_async_ports は、プロセスが非同期ディスク I/O ドライバにアクセスするためにオープンするポートの最大数を指定します(このパラメータは RS-232 非同期データ通信インタフェースに対しては作用しません)。システムは、プロセスと非同期ディスク・ドライバ間のすべての通信に使用されるポートがオープンされた時点で、各ポートに対してポート構造を配列として割り当てます。アプリケーションが必要とする非同期ポートの数は通常、そのアプリケーションに付属するドキュメントで指定されています(データベース・アプリケーション・ソフトウェア、ビデオ管理ソフトウェアなど)。

max_async_ports の適切な値を決定するには、以下の手順を行います。

1. 非同期ディスク I/O を使用する各アプリケーションとプロセスに必要なポート数を調べます。
2. 上記のアプリケーションのうちどのアプリケーションが別のプロセスとして同時に実行されるかを調べます。また、あるアプリケーションの複数のコピーが同時に別のプロセスとして実行されるかどうかを調べます。
3. 上記の数値に基づいて、すべてのプロセスが非同期ディスク I/O ドライバにアクセスするためにオープンするポートの最大数を決定して、適切な合計ポート数を算出します。
4. max_async_ports を上記の値以上に設定してください。

付加情報

[非同期 I/O に関するパラメータの概要](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

maxusers

システムに同時にログインするユーザーの最大想定数に基づいて、システム・リソースを割り当てます。

指定可能な値:

最小値	0
最大値	メモリ・サイズに依存
デフォルト値	32

このパラメータには、**整数値**を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

maxusers は、システム・リソース割り当ての上限を設定するもので、実際にシステムにログインするユーザー数を設定するものではありません。maxusers 自体がシステム内の構造のサイズを決定するのではなく、maxusers の値に基づいて、他のグローバル・システム・パラメータのデフォルト値が決まります。maxusers を基準に他の構成可能パラメータの値を定義する場

合、リソース割り当てのバランスが取れていない場合に発生する無駄なスペースを最小限に抑えることができるので、カーネルのサイズが小さくなり、効率が向上します。

maxusers は、C 言語のマクロ MaxUsers を定義します(#define MaxUsers 8 など)。このマクロにより、システム・テーブルのサイズが決定されます。実際のユーザー数の上限は、購入した HP-UX ライセンスのバージョンによって異なります。実際のユーザー数の上限を調べるには、uname -a コマンドを使用します。

構成可能パラメータをそれぞれ個別に変更するより、ユーザーの最大想定数を基準にして、数式でパラメータを指定の方が簡単です(例: nproc=(20+8*MaxUsers))。このようにすると、システム上のユーザーの最大数を増やす場合でも、maxusers パラメータを変更するだけで済みます。

関連するパラメータ

[nproc](#)

[ncallout](#)

[ninode](#)

[nfile](#)

デフォルト値は、カーネル定義ファイル/etc/conf/master.d/*の maxusers を基準に設定されています。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ncallout

カーネルによるスケジュール設定が可能なタイムアウトの最大数を指定します。

指定可能な値:

最小値	6
最大値	メモリ・サイズに依存
デフォルト値	16+nproc

このパラメータには、整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

ncallout は、カーネルによるスケジュール設定が可能なタイムアウトの最大数を指定します。タイムアウトを使用するものを以下に示します。

- alarm()システム・コール
- setitimer()システム・コール
- select()システム・コール
- ドライバ
- uucp プロセス
- プロセスのスケジュール設定

システムがこのタイムアウト値を超えた場合、システム・コンソールに以下の致命的エラーを示すメッセージが表示されます。

```
panic: timeout table overflow
```

関連するパラメータ

nproc の値を増やした場合、それに合わせて ncallout の値を増やすことが必要です。一般的な原則として、複数のコールアウトを使用するプロセスがない限り、1 つのプロセスにつき 1 つのコールアウトしか許可してはいけません。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ncdnode

ncdnode は、メモリ内でオープンできる CD-ROM ファイル・システム・ノードの最大数を定義します。

指定可能な値:

最小値	14
最大値	メモリ・サイズに依存
デフォルト値	150

このパラメータには、整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

ncdnode は、(vnode テーブル内の)メモリ内でオープンできる CD-ROM ファイル・システム・ノードの最大数を定義します。このパラメータの機能は ninode と同じですが、CD-ROM ファイル・システムだけにしか適用されない点が異なります。ncdnode の動作は、ワークステーション・システムとサーバー・システムのどちらの場合も同じです。

各ノードは 288 バイトを使用します。例えば ncdnodes が 10000 に設定されている場合は、CD-ROM ファイル・システムのノード・テーブル専用に約 3 MB のメモリが予約されます。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nclist

nclist は、pty デバイスおよび tty デバイスによるデータ転送用の cblock 数を指定します。

指定可能な値:

最小値	132
最大値	使用可能なメモリにより制限
デフォルト値	(100+16*MAXUSERS)

このパラメータには、整数値を直接指定するか、または整数式を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

nclist は、システム内で割り当てる cblock 数を指定します。転送データは、tty デバイスおよび pty デバイスの通過時に cblock に保存されます。

nclist のデフォルト値(100+16* MAXUSERS)の計算式では、システムがコンソールなどとやりとりする通信データを処理する際に使用する cblock 数を 100 とし、1 回のユーザー・セッションあたりの cblock 数平均 16 としています。cblock は、ログイン・セッションだけでなくシリアル接続(SLIP 接続、UUCP 転送、端末エミュレータなど)にも使用されることに注意してください。お使いのシステムが上記のシリアル接続を使用している場合は、接続数に応じて nclist の値を増やす必要があります。

cblock プールがすべて使用された場合は、cblock が必要となった時点で使用できないため、tty デバイスおよび pty デバイスを通過中のデータが失われる可能性があります。このような状況が発生した場合は、システム・メッセージ・バッファに以下のメッセージが書き込まれます。

WARNING: cblock exhaustion has occurred n times (see termio(7)).

上記のメッセージは、nclist の値を増やす必要があることを示しています。nclist の最小値は 132 です。最大値に制限はありませんが、各 cblock が 32 バイトの(スワップ不可能な)常駐マシン・メモリを使用することを考慮に入れて値を設定することが必要です。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ndilbuffers

ndilbuffers は、同時にオープン可能なデバイス I/O ライブラリ・デバイスの最大数を指定します。

指定可能な値:

最小値	1
最大値	メモリ・サイズに依存
デフォルト値	30

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

ndilbuffers は、システム全体で同時にオープン可能なデバイス I/O ライブラリ(DIL)デバイスファイルの最大数を指定します。

ndilbuffers は、デバイス I/O ライブラリ(DIL)が専用で使用します。DIL が使用されていない場合、DIL バッファは不要です。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

npty

npty は、システム上で使用可能な擬似 tty データ構造の最大数を指定します。

指定可能な値:

最小値	1
最大値	メモリ・サイズに依存
デフォルト値	60

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

npty は、擬似 tty ドライバが使用する以下の構造の数を制限します。

```
struct tty pt_tty[npty];  
struct tty *pt_line[npty];  
struct pty_info pty_info[npty];
```

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nstrtel

nstrtel は、受信した telnet セッションでカーネルがサポート可能な telnet デバイス・ファイルの数を指定します。

指定可能な値:

最小値	60
最大値	なし
デフォルト値	60

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

nstrtel は、サーバー上で受信する telnet セッションが使用するデバイス・ファイルのサポートに必要なカーネル・データ構造を、システムのブート時に何個作成するかを指定します。この数は、システム上に存在するデバイス・ファイルの数と一致していることが必要です。insf コマンドまたは SAM を使ってさらに多くの telnet デバイス・ファイルを作成する場合、nstrtel の値もそれに合わせて増やす必要があります。そうでなければ、システムとの通信に使用できるカーネル・データ構造がないためデバイス・ファイルを使用できなくなります。

nstrtel には、システム上の telnet デバイス・ファイルの数以上の値を指定します。システム上に実際に存在するデバイス・ファイル数を超える値を設定すると、余分なデータ構造により浪費されるメモリの量が増加します。ただし、後でデバイス・ファイルを追加する予定がある場合は、このように設定しても構いません。

付加情報

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

nsysmap、nsysmap64

カーネルの動的メモリ仮想アドレス空間リソース・マップ内のエントリ数を設定します。

指定可能な値:

最小値	800
最大値	メモリ・サイズに依存
デフォルト値	2 * nproc

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

nsysmap および 64 ビット版の nsysmap64 は、カーネルの動的メモリ・リソース・マップのサイズを設定します。このリソース・マップはアドレス/長さのペアの配列で、カーネルの動的アドレス空間内の空き仮想空間を示します。64 ビット版のカーネルではより多くの仮想アドレス空間を使用できるため、32 ビット版のカーネルと 64 ビット版のカーネル用にそれぞれ異なるパラメータが用意されています。

以前は、カーネルの動的メモリ・リソース・マップは、システムによってのみ設定され、簡単には変更できませんでした。特定の作業負荷により、カーネルのアドレス空間が著しくフラグメント化 (断片化) されたような場合は、リソース・マップ内に非常に多くのエントリが作成されてしまいます。こうした状況が発生すると、リソース・マップ内に新たなエントリを書き込めずに、「リークのある」カーネル仮想アドレス空間が発生することになります。そのため、このオーバーフローが頻繁に発生すると、仮想アドレス空間が使い尽くされてしまいます。

システムはアルゴリズムを使用して、ブート時にシステムの作業負荷に応じてマップ・サイズを自動的に調整します。この自動設定される値ではメモリ・リソース・マップ配列のオーバーフローを十分に防止できない場合は、特定の作業負荷に合わせてこのパラメータを調整してください。

デフォルト値以外の値を設定した場合であっても、システムのサイズによっては、指定した値よりも大きい値がカーネルにより設定されることがある点に注意してください。

目的

この調整可能パラメータは、システム・パニック `kalloc: out of virtual space` に対処するために追加されたものです。リソース・マップ・オーバーフローが発生していないシステムでは、このパラメータを変更する必要はありません。

リソース・マップ・オーバーフローが発生すると、コンソールに以下のメッセージが表示されます。

```
sysmap32: rmap ovflo, lost [X,Y]*
```

または

```
sysmap64: rmap ovflo, lost [X,Y]
```

(X および Y は 16 進数値)

この状況がまれにしか発生しない場合は、パラメータを変更する必要はありません。頻繁に発生する場合、例えば再ブートせずに長期間 (1 年以上など) 使用し続けるようなシステム上でこの状況が日に何度も発生するような場合は、設定値を大きくする必要があります。値を大きくしなければ、以下に示すシステム・パニックが発生する可能性があります。

```
kalloc: out of kernel virtual space
```

`nsysmap` または `nsysmap64` の値を大きくする場合は、大体の目安としてまず値を 2 倍にしてみてください。値を 2 倍にすることを何度か繰り返しても問題が解消しない場合は、カーネルに関して他の問題があるものと思われます。当社サポート担当までお問い合わせください。

副作用

このパラメータの値を大きくすると、カーネルのメモリ使用量がごくわずかだけ増加します。そのため作業負荷によっては、パラメータに大きい値を設定しすぎると、カーネルのメモリ割り当て性能が低下することがあります。

このパラメータの値をデフォルト値よりも小さくすることは危険です。リソース・マップ・オーバーフローが発生する可能性が高くなり、ついにはカーネル・パニックを引き起こす危険性があります。`nsysmap` の値を小さくする場合は、事前に当社サポート担当まで必ずご相談ください。

付加情報

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

`o_sync_is_o_dsync`

`o_sync_is_o_dsync` は、`open()` コールまたは `fcntl()` コールで使われている `O_SYNC` フラグを、`O_DSYNC` フラグに変換するかどうかを指定します。

指定可能な値:

最小値	0
最大値	1
デフォルト値	0

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

`open()` コールまたは `fcntl()` コールで、`O_SYNC` フラグまたは `O_DSYNC` フラグを設定すると、ディスクへのデータ書き込みが確実に終了してからコールが返されるようになります。これらのフラグを設定しなければ、システム・ソフトウェアおよびハードウェアにより書き込み処理が正常に終了するという想定の下で、ディスク・アクセス要求が開始された直後に関数から制御が戻されます。

`O_SYNC` フラグまたは `O_DSYNC` フラグを設定した場合は、要求したディスク I/O 処理が終了するまでは関数から呼び出し元プロセスに制御が戻されないため、ディスクへのデータ書き込みが正常に終了したことが保証されます。この点についてはどちらのフラグも同じ働きをしますが、1 つだけ大きな違いがあります。つまり `O_SYNC` フラグを設定した場合は、ディスク操作が終了し、さらにこの書き込み処理に伴うすべてのファイル属性 (アクセス時刻、変更時刻、状態変更時刻など) の更新が終了するまでは、関数から制御が戻されません。これらの処理がすべて終了して初めて、呼び出し元プロセスに制御が戻されます。

`o_sync_is_o_dsync` を 1 に設定すると、すべての `open()` コールまたは `fcntl()` コール内の `O_SYNC` フラグが、`O_DSYNC` に変換されます。フラグを変換すると、ディスク上のファイル属性の更新前に呼び出し元プロセスに制御が戻されるようになるため、システム障害が発生した場合には、ディスク上の属性情報が正しく更新されていない状態になる可能性があります。

このパラメータに 0 でない値を指定すると、ファイルのタイムスタンプ属性の更新を待たずに関数から制御が戻されるようになります。ただしこの場合も、呼び出し元プロセスの続行前に、実ファイル・データのディスク書き込みは確実に終了しています。この設定は、大量のディスク I/O を実行し、かつファイル・データの整合性が重要になる環境で有益であり、タイムスタンプの更新を待たずに呼び出し元処理を続行できるため処理性能が向上します。システムやディスクのクラッシュ時に正しいファイル・アクセス時刻が失われる危険性はあっても、性能面での利点がこれを上回ると判断できる場合は、パラメータを 1 に設定してください。それ以外の場合は、デフォルト値の 0 のままにしておいてください。

`O_SYNC` フラグまたは `O_DSYNC` フラグが使われていない場合は、このパラメータ設定によるディスク I/O 処理への影響はありません。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

pfail_enabled

システム電源異常関連ルーチンを使用可能/使用不可能にします (サーバーのみ)。

指定可能な値:

最小値	0
最大値	1
デフォルト値	0

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

pfail_enabled は、ワークステーション・システムによるローカルな電源異常(システムの中央バスに影響を及ぼしてコンピュータの停止に到る)への対応を可能にするかどうかを決定します。以下の説明に従って、0 または 1 を指定してください。

- 0 電源異常は検出されません。この設定では、電源異常からの復元を可能にする/sbin/powerfail コマンド (/etc/inittab から開始) は実行されません。電源異常の発生時に実行中であったプログラムを、電源復元後に再開することはできません。
- 1 電源異常を検出します。この設定では、システムにより電源異常が検出されて、/sbin/powerfail コマンド (/etc/inittab から開始) に関連した復元機構が実行されます。そのため電源異常が発生しても、そのとき実行中であったプログラムを電源復元後に再開できます。

電源異常に伴うシステムのシャットダウンと起動は、ガイドラインに従って適切に実行する必要があります。ガイドラインの詳細は、「システム/ワークグループの管理」を参照してください。

powerfail は、すべての/etc/inittab ファイルに記述されていますが、このエントリは powerfail をサポートするシステムでのみ認識されて使用される点に注意してください。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

public_shlibs

共有ライブラリに対する「パブリック」保護 ID を使用可能にします。

指定可能な値:

最小値	0
最大値	1 (または 0 でない値)
デフォルト値	0

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

public_shlibs は、共有ライブラリに対する「パブリック」保護 ID を使用可能にします。

共有ライブラリは mmap() を使って実装されており、個々の mmap() には一意の保護 ID が与えられています。各プロセスは 4 つの保護 ID レジスタを持ち、そのうち 2 つはテキストまたはデータにハードコーディングされています。残りの 2 つは、各ユーザー・プロセスがアクセスする共有ライブラリや共有メモリ・セグメントなどの間で共有されます。

共有ライブラリが導入されると、プロセス ID (pid) のスラッシングの増加に伴う処理性能の低下という問題が発生しました。この影響を最小限に抑えるために、すべての共有ライブラリ・マッピングにパブリック保護 ID が追加されて、2 つの保護 ID レジスタにアクセスする共有ライブラリを、オブジェクトのプールから効率よく削除できるようになりました。

public_shlibs を 1 に設定すると、共有ライブラリにパブリック保護 ID が割り当てられます。また 0 に設定すると、パブリック・アクセスが無効化されて、すべての共有ライブラリに一意の保護 ID が与えられます。デフォルト値は 1 であり、0 でない値はすべて 1 と解釈されます。

なんらかの「セキュリティ・ホール」があるなど、パブリック値を使用すべきでない特別な理由がある場合を除き、このパラメータは 0 に変更しないでください。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

rtsched_numpri

リアルタイム・プロセス・スケジューリング優先順位の最大数を指定します。

指定可能な値:

最小値	32
最大値	512

デフォルト値 32

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

rtsched_numpri は、リアルタイム・スケジューラの下で実行されるリアルタイム・プロセス用に使用できる優先順位の最大数を指定します (POSIX 標準 P1003.4)。

推奨値

通常の構成では、デフォルト値の 32 のままで十分です。各プロセスにさらに詳細な優先順位レベルを設定する必要がある場合は、この値を適宜増分してください。ただし、rtsched_numpri の値を大きくして優先順位の数を増やすと、実行すべきプロセスの判断に時間がかかるようになり、システム全体の処理性能が低下する点にも注意してください。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

scroll_lines

scroll_lines は、ITE コンソール・スクリーン・バッファ内の表示行数を指定します。

指定可能な値:

最小値	60
最大値	999
デフォルト値	100

このパラメータには、整数値を指定してください。詳細は、パラメータ値の指定を参照してください。

説明

scroll_lines は、スクロール領域のサイズ、つまりシステム内に構成されている各内部端末エミュレータ (ITE) ポート上の、エミュレートされた端末スクリーン・メモリの行数を定義します。

システム内に構成されている各グラフィックス・インターフェイスについて、グラフィックス・ドライバは、 $2 \times$ 行の長さ \times scroll_lines バイト分のデータを使用します。例えば、ITE 行の長さが 128 文字のグラフィックス・ディスプレイで scroll_lines を 100 に設定した場合は、各グラフィックス端末のスクリーン・メモリには $2 \times 128 \times 100 = 25600$ バイト分のカーネル・メモリが割り当てられます。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

sendfile_max

sendfile_max は、sendfile()システム・コールで一度に送信できるバッファ・キャッシュの最大ページ数を定義します。

指定可能な値:

最小値	0
最大値	0x40000
デフォルト値	0

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

sendfile_max は、sendfile()システム・コールで一度に占有できるバッファ・キャッシュのページ数を制限します。sendfile() は Web サーバーによって使われるシステム・コールであり、send()システム・コールを使ってユーザー・スペースからカーネル・スペースにデータをコピーするのに伴うオーバーヘッドを回避する目的で使用されます。ネットワーク・ソフトウェアは、データがワイヤを介して送信されている間に、バッファキャッシュ・バッファを直接使用します。通常この送信処理はごく短時間で終了しますが、低速リンクを介してデータを送信する場合や、エラーによる再送信が必要になった場合は、送信時間が通常よりもかなり長くなることがあります。

sendfile_max は、sendfile()システム・コールで一度にアクセスできるバッファ・キャッシュ・メモリ量を制限することにより、使用可能なすべてのバッファ・キャッシュが sendfile()によりロックされてしまうのを防止します。

sendfile_max は、sendfile で一度に送信できるバッファ・キャッシュのページの上限值です。最小値の 0 は、バッファ数に制限を設けないことを意味します。0 でない値を指定した場合は、バッファ・キャッシュへのアクセスが指定したページ数以内に制限されます。デフォルト値は 0 です。

sendfile_max を 1 に設定すると、事実上バッファを利用できなくなります。すべてのバッファは最低 1 つのページから構成されているため、この値を指定した場合は、バッファ・スペースに対する最初の要求で上限値を超えてしまい、バッファにアクセスできません。この場合 sendfile()処理は取り消されて、send()システム・コールで行われるように、malloc()とデータ・コピーを使った処理が実行されます。

sendfile_max に 1 でない値を最大 0x40000 までの範囲で設定すると、通常のシステム処理の間にバッファの空きスペースが不足するような場合に、sendfile()により大量のバッファが占有されてしまうのを防止できます。

付加情報

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

スピンロック・パラメータ (技術情報は非公開)。

以下に示すパラメータは、いずれもマルチプロセッサ・コンピュータ上のスピンロック・プールに関連するものであり、類似した働きを持つため、ここでまとめて説明します。各パラメータは、それぞれ対応するシステム・リソースについて、指定された数のスピンロックを割り当てます。

bufcache_hash_locks

バッファ・キャッシュ・スピンロック・プール

chanq_hash_locks

チャンネル待ち行列スピンロック・プール

ftable_hash_locks

ファイル・テーブル・スピンロック・プール

io_ports_hash_locks

I/O ポート・スピンロック・プール

pfdat_hash_locks

Pfdat スピンロック・プール

region_hash_locks

プロセス・リージョン・スピンロック・プール

sysv_hash_locks

System V プロセス間通信スピンロック・プール

vnode_cd_hash_locks

vnode クリーン/ダーティ・スピンロック・プール

vnode_hash_locks

vnode スピンロック・プール

これらのパラメータは、マルチ・プロセッサ上でのスピンロックの使われ方や、必要なスピンロック数とシステムのサイズや複雑さとの相関関係などについて、十分な知識のあるユーザーによる使用を前提としています。パラメータの変更に伴う影響を完全に把握できない場合は、これらのパラメータはデフォルト値のままにしておいてください。通常は、これらのパラメータに詳しい当社のサポート・エンジニアから指示がない限り、これらのパラメータを変更しないでください。

これらのパラメータに不適切な値を設定すると、マルチ・プロセッサ・システムの性能が大幅に低下する可能性があります。

指定可能な値:

これらのパラメータの下限値と上限値はすべて同一です。

以下に示すように、デフォルト値のみが異なります。

最小値	64
最大値	4096
デフォルト値	64 (ftable_hash_locks、io_ports_hash_locks)
デフォルト値	128 (bufcache_hash_locks、pfdat_hash_locks、region_hash_locks、 sysv_hash_locks、vnode_hash_locks、vnode_cd_hash_locks)
デフォルト値	256 (chanq_hash_locks)

このパラメータには、2 の累乗となる整数値を指定してください。それ以外の値を指定した場合は、SAM またはカーネル自体により、指定したパラメータ値よりも大きい最小の 2 の累乗値に自動的に変更されます。例えば 100 と指定した場合は、スピンロック・プール値は 128 に変更されます。詳細は、[パラメータ値の指定](#) を参照してください。

説明

簡単に説明すると、スピンロックとはマルチ・プロセッサ・システム上でプロセッサ間の対話を制御するための機構であり、あるプロセッサがタスクを終了するまで別のプロセッサを待機させて、先に終了したプロセッサから待機しているプロセッサに結果を渡せるようにします。このスピンロックにより、ファイル・システム vnode、I/O ポート、バッファ・キャッシュなどの各種リソースへのアクセスが制御されます。

以前の HP-UX バージョンでは、すべてのリソースに固定数のスピンロックを割り当てていましたが、HP-UX 11.0 からは、非常に大規模で複雑なシステムにも対応できるように、それぞれのリソース・タイプ別に異なるスピンロック値を割り当てられるようになりました。

通常、これらのハッシュされたプールのいずれかでロック競合が発生した場合は、まず競合が発生しているリソース・スピンロック・プールを特定し、次にそのリソース用のスピンロック・プール・パラメータの値を大きくします。スピンロック・プールの値は、必ず 2 の累乗となる整数でなければなりません。それ以外の値を指定した場合は、カーネルにより、指定した値よりも大きい最小の 2 の累乗値に自動的に変更されます。

前述したように、これらのパラメータは十分な経験と知識のあるシステム管理者による使用を前提としています。変更内容が適切かどうかははっきりしない場合は、これらのパラメータを変更しないでください。

付加情報

[スピンロックおよびその他パラメータの概要](#)

[スピンロックおよびその他パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

timezone

世界協定時刻 (UTC) からローカル・タイム・ゾーンへの西方向の時差を指定します。

指定可能な値:

最小値	-720
最大値	720
デフォルト値	420

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#)を参照してください。

説明

timezone は、世界協定時刻 (UTC) から、システムが存在するローカル・タイム・ゾーンへの西方向の時差を分単位で指定します。負の値は、世界協定時刻 (UTC) から東方向の時差(分単位)とみなされます。この値は、/usr/include/sys/time.h に定義されている以下の構造体に保存されます。

```
struct timezone tz = { TimeZone, DST };
```

```
struct timezone {
```

```
int tz_minuteswest; /* minutes west of Greenwich */
```

(グリニッチ標準時からの西方向の時差 (分単位))

```
int tz_dsttime; /* type of dst correction */
```

(夏時間補正タイプ)

```
};
```

```
#define DST_NONE 0 /* not on dst */
```

(夏時間使用不可能)

```
#define DST_USA 1 /* USA style dst */
```

(米国形式)

```
#define DST_AUST 2 /* Australian style dst */
```

(オーストラリア形式)

```
#define DST_WET 3 /* Western European dst */
```

(西ヨーロッパ形式)

```
#define DST_MET 4 /* Middle European dst */
```

(中央ヨーロッパ形式)

```
#define DST_EET 5 /* Eastern European dst */
```

(東ヨーロッパ形式)

関連するパラメータ

dst

timezone および dst はいずれもシステム・クロック・ルーチンで使用され、TZ 環境変数との整合性を保つ必要があります。詳細は、HP-UX Reference の environ(5)) および login(1)) を参照してください。

付加情報

その他の各種パラメータの概要

その他の各種構成可能パラメータの一覧

構成可能カーネル・パラメータ・ヘルプのブラウズ

unlockable_mem

システム・オーバーヘッドおよび仮想メモリ管理用に確保しておくメモリの最小量を指定します。

指定可能な値:

最小値	0
最大値	電源投入時に示される使用可能メモリ量
デフォルト値	0 (システムにより適切な値を設定)

このパラメータには、整数値を指定してください。詳細は、[パラメータ値の指定](#) を参照してください。

説明

unlockable_mem は、システム・オーバーヘッドおよび仮想メモリ管理用に常に残しておくメモリの最小量を定義します。以下の計算からわかるとおり、ロック不可能なメモリ量を増やすと、ロック可能なメモリ量が減少します。

- 使用可能なシステム・メモリ総量 - unlockable_mem = ロック可能なメモリ量

unlockable_mem は 4 KB ページ単位で指定します。現在の使用可能メモリ量とロック可能メモリ量は、起動時のメッセージ内に物理ページ・サイズと共に示されます。また後から/etc/dmesg を実行して調べることもできます。

unlockable_mem に指定した値が使用可能なシステム・メモリ量を超える場合は、使用可能なメモリ量と同じ値が自動的に設定されます。このときロック可能メモリ量は 0 になります。

ロック可能メモリ量が不足すると、ロック可能メモリを必要とするコールが失敗する可能性があります。ロック中である場合を除き、このロック可能メモリは仮想メモリに使用できる点に注意してください。

ロック可能メモリは、以下のような用途に使用されます。

- `plock()`システム・コールによりロックされるプロセス・イメージとオーバーヘッド (HP-UX Reference の `plock(2)` を参照)
- `shmctl()`システム・コールの `SHM_LOC` コマンドによりロックされる、共有メモリ・セグメント (HP-UX Reference の `shmctl(2)` を参照)
- 共有メモリ・システムおよび一部ドライバにより使用される、各種の動的カーネル・データ構造体

関連するパラメータ

`unlockable_mem` には、システム上で使用可能なメモリ総量以下の値を指定します。

付加情報

[その他の各種パラメータの概要](#)

[その他の各種構成可能パラメータの一覧](#)

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

ドキュメントに記載されていないカーネル・パラメータ

カーネル・マスター・ファイルに含まれている構成可能カーネル・パラメータの中には、以下に示すような理由により、ドキュメントに記載されていないものや、SAM による認識またはサポートの対象外となっているものもあります。

- 廃止されたパラメータ。現在の HP-UX リリースではすでに使われていないパラメータが、既存のカーネル構成内に残っていることがあります。SAM により、廃止されたパラメータが現在のカーネル構成内で検出された場合は、このパラメータは変更が可能な構成可能パラメータの一覧には示されません。また、カーネルの構築に使われる新しい構成ファイルの作成時には、このパラメータは削除されます。
- SAM ではサポートされていないパラメータ。廃止されたパラメータと同様に、このタイプのパラメータも構成可能パラメータの一覧には示されません。ただし、パラメータが失われたことによる誤作動が生じないように、新しいカーネル構成ファイル内にはそのまま残されます。
- カーネル構成ソフトウェアにより値を割り当てられるパラメータ。通常は外部要因に基づいて値が決定されます。割り当てられている値は、他のパラメータ値の計算に使われることもあります。
- 弊社内部またはサポート目的でのみ使われるパラメータ。弊社の公式なサポート要員から直接の指示がない限り、これらのパラメータはデフォルト値のままにしておいてください。
- 廃止されたか、または廃止予定のソフトウェアをサポートするパラメータ。デフォルト値以外の値を指定する方法については、そのパラメータがサポートするソフトウェアに付属するドキュメントを参照してください。

付加情報

[構成可能カーネル・パラメータ・ヘルプのブラウズ](#)

HP-UX

www.hpe.com/jp/hpux

© Copyright 2018 Hewlett Packard Enterprise Development LP.



本書の内容は、将来予告なく変更されることがあります。日本ヒューレット・パッカード製品およびサービスに対する保証については、当該製品およびサービスの保証規定書に記載されています。本書のいかなる内容も、新たな保証を追加するものではありません。日本ヒューレット・パッカードは、本書中の技術的あるいは校正上の誤り、脱字に対して、責任を負いかねますのでご了承ください。