



shaping tomorrow with you

Oracle Solaris IPCパラメーター設定手順書

2016年3月

第1.2版

富士通株式会社

■使用条件

- 著作権・商標権・その他の知的財産権について

コンテンツ(文書・画像・音声等)は、著作権・商標権・その他の知的財産権で保護されています。

本コンテンツは、個人的に使用する範囲でプリントアウトまたはダウンロードできます。ただし、これ以外の利用(ご自分のページへの再利用や他のサーバへのアップロード等)については、当社または権利者の許諾が必要となります。

- 保証の制限

本コンテンツについて、当社は、その正確性、商品性、ご利用目的への適合性等に関して保証するものではなく、そのご利用により生じた損害について、当社は法律上のいかなる責任も負いかねます。本コンテンツは、予告なく変更・廃止されることがあります。

- 輸出または提供

本ドキュメントを輸出又は提供する場合は、外国為替及び外国貿易法及び米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとり下さい。

■商標について

- UNIX は、米国およびその他の国におけるオープン・グループの登録商標です。
- SPARC Enterprise、SPARC64、SPARC64 ロゴおよびすべての SPARC 商標は、米国 SPARC International, Inc.のライセンスを受けて使用している、同社の米国およびその他の国における商標または登録商標です。
- Oracle と Java は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- その他各種製品名は、各社の製品名称、商標または登録商標です。

はじめに

本書の内容

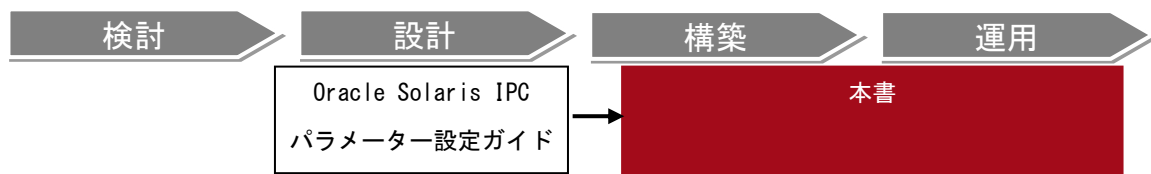
- 本書は、Oracle Solaris 10 および Oracle Solaris 11 環境における IPC (Inter Process Communication: プロセス間通信) パラメーターの設定手順について記載しています。

留意事項

- 本書は Oracle Solaris 11.3 に基づいています。
- 本書では Oracle Solaris を Solaris と記載することがあります。
- 本書では Oracle Solaris ゾーンをゾーン、non-global zone と記載することがあります。

ドキュメントの位置付け

- 本書は以下の位置づけになります。



目 次

1. IPC パラメーター概要	4
2. IPC パラメーターの見積り方法	6
3. IPC パラメーターの設定	7
3-1. IPC パラメーターの確認	8
3-2. IPC パラメーターの設定	11
3-2-1. プロジェクトデータベースの編集方法	11
3-2-2. IPC パラメーターを有効にする方法	12
3-2-3. デーモン起動時に IPC パラメーターを有効にする方法	13
3-3. IPC パラメーターの監視設定	15
4. パラメーターの変更方法	17

1. IPC パラメーター概要

Oracle Solaris 10 および Oracle Solaris 11 における IPC (Inter Process Communication: プロセス間通信) パラメーターの設定は資源制御パラメーターを使用します。

本書では、IPC 資源 (共有メモリ、セマフォ、メッセージキュー) の設定について、資源制御パラメーターを使用する手順を記載します。また、本手順の実行は global zone 上で実施していますが、non-global zone においても同様に設定可能です。

資源制御パラメーターはローカルシステムやネームサービス環境上のプロジェクトやユーザ単位に IPC の設定が可能です。non-global zone に対しても同様な設定が必要となります。Solaris 9 までは IPC 資源はカーネルパラメーターによって制御され、これらのデフォルト値を変更するためには /etc/system ファイルを変更し、システムを再起動する必要がありました。(Solaris 10、Solaris 11 でも/etc/system は有効)しかし、Solaris 10、Solaris 11 では IPC 機能を資源制御から調整することができるので、システムの起動中に変更することが可能です。

■IPC パラメーター

資源制御	旧パラメーター	旧デフォルト値	新デフォルト値	最大値
project.max-shm-memory	shminfo_shmmax	0x800000	約 1/4 物理メモリ量	UINT64_MAX
project.max-shm-ids	shminfo_shmmni	100	128	2**24
process.max-msg-qbytes	msginfo_msgmnb	4096	65536	ULONG_MAX
process.max-msg-messages	msginfo_msgtql	40	8192	UINT_MAX
project.max-msg-ids	msginfo_msgmni	50	128	2**24
process.max-sem-ops	seminfo_semopm	10	512	INT_MAX
process.max-sem-nsems	seminfo_semmsl	25	512	SHRT_MAX
project.max-sem-ids	seminfo_semmni	10	128	2**24

```
* SHRT_MAX      :          32,767 (max value of a "short int" )
* 2**24         :          16,777,216
* INT_MAX       :      2,147,483,647 (max value of an "int")
* UINT_MAX      :      4,294,967,295 (max value of an "unsigned int")
* UINT64_MAX    :      18,446,744,073,709,551,615
* ULONG_MAX:    :      18,446,744,073,709,551,615 (max of "unsigned long int" :64 ビット)
* ULONG_MAX:    :      4,294,967,295 (max of "unsigned long int" :32 ビット)
```

- 📌 project.max-shm-memory の実際のデフォルト値は 1/4 より小さい値です。正確なデフォルト値はコマンドで確認してください。確認コマンドについては「3-1. IPC パラメーターの確認 3)」の手順を参照してください。

👉 /etc/system に旧パラメータによる記述があった場合/etc/system の値がデフォルト値として設定されます。但し、新デフォルト値より小さい値を設定した場合はそのパラメータは無視されます。また、shminfo_shmmax 値のみカーネルにより自動設定されます。

	資源制御	説明	単位
共有メモリ	project.max-shm-memory	プロジェクトに許可された共有メモリの総バイト	サイズ(バイト)
	project.max-shm-ids	プロジェクトに許可された共有メモリ ID の最大数	共有メモリ数(IDs)
セマフォ	process.max-sem-ops	semop コール毎の許可されたセマフォ操作の最大数	セマフォ操作数
	process.max-sem-nsems	セマフォセット毎の許可されたセマフォの最大数	セマフォ数
	project.max-sem-ids	プロジェクトに許可されたセマフォ ID の最大数	セマフォ数(IDs)
メッセージキュー	process.max-msg-qbytes	メッセージキュー上のメッセージの最大バイト数	サイズ(バイト)
	process.max-msg-messages	メッセージキューにあるメッセージの最大数	メッセージ数
	project.max-msg-ids	プロジェクトに許可されたメッセージキュー ID の最大数	メッセージキュー数(IDs)

<参考>廃止された IPC パラメータ

パラメータ名	デフォルト値	最大値	廃止 Ver.
msgsys:msginfo_msgmax	2048	2,147,483,647(32Bit) ULONG_MAX (64Bit)	Solaris 10 から廃止
semsys:seminfo_semmns	60	2,147,483,647	Solaris 10 から廃止
semsys:seminfo_semvmx	32767	65,535	Solaris 10 から廃止
semsys:seminfo_semmnu	30	2,147,483,647	Solaris 10 から廃止
semsys:seminfo_semaem	16384	32,767	Solaris 10 から廃止
semsys:seminfo_semume	10	2,147,483,647	Solaris 10 から廃止
shmsys:shminfo_shmseg	6	32,767	Solaris 9 から廃止
shmsys:shminfo_shmmin	1	ULONG_MAX	Solaris 9 から廃止
msgsys:msginfo_msgmap	100	2,147,483,647	Solaris 8 から廃止
msgsys:msginfo_msgseg	1024	32,767	Solaris 8 から廃止
msgsys:msginfo_msgssz	8	2,147,483,647	Solaris 8 から廃止
semsys:seminfo_semmmap	10	2,147,483,647	Solaris 8 から廃止
semsys:seminfo_semusz	96	2,147,483,647	Solaris 2.6 から廃止

2. IPC パラメーターの見積り方法

IPC パラメーターの見積りは、以下の計算方法を参考に行ってください。

- IPC パラメーターを見積もるときは、IPC パラメーターの種類が Σ 型か MAX 型かを考慮する必要があります。詳細は「Oracle Solaris IPC パラメーター設定ガイド」を参照してください。

3. IPC パラメーターの設定

ここでは/etc/project へのパラメーター設定の方法を解説します。/etc/system は従来どおりの設定方法に従ってください。Solaris の資源制御機能は、project データベース (/etc/project) によって構成され、資源制御の属性は project データベースエントリの最後のフィールドで定義します。

- 定義例: ユーザ ora10 に project.max-shm-memory を定義する。

<pre>user:ora10:100:::project.max-shm-memory=(privileged,300,deny) ~~~~~</pre>				
(A)	(B)	(C)	(D)	(E)
<p>(A): プロジェクト名 (B): 資源制御パラメーター名 (C): 特権レベル(basic,privileged,system) basic(基本値): 呼び出し元プロセスの所有者が変更できる privileged(特権値): 特権を持っている呼び出し元(スーパーユーザ)だけが変更できる system(システム値): OSによる処理が実行されている間は、固定される ※通常管理者は一般ユーザに対しては privileged 権限で定義します。 (D): しきい値 (E): 特定のしきい値に対応付けられたアクション(deny, signal=シグナル名) deny : しきい値を超える量の資源要求を拒否できる signal=シグナル名: しきい値に達した場合は、違反プロセスまたは監視プロセスにシグナルを送信する</p>				

👉 特にマニュアル等に指示がない場合は、特権レベルに privileged、アクションに deny を設定してください。

- 資源制御の設定・管理に使用するコマンド

コマンド名	説明
projects(1)	ユーザのプロジェクトメンバーシップを表示する
newtask(1)	ユーザのデフォルトのシェルまたは指定されたコマンドを実行し、指定されたプロジェクトが所有する新しいタスクに実行コマンドを配置する。また、newtask は、実行中のプロセスに結合するタスクとプロジェクトを変更するためにも使用できる。
projadd(1M)	/etc/project ファイルに新しいプロジェクトエントリを追加する。projadd は、ローカルシステム上にだけプロジェクトエントリを作成する。projadd は、ネットワークネームサービスから提供される情報は変更できない。
projmod(1M)	ローカルシステム上のプロジェクトの情報を変更する。projmod は、ネットワークネームサービスから提供される情報は変更できない。
projdel(1M)	ローカルシステムからプロジェクトを削除する。projdel は、ネットワークネームサービスから提供される情報は変更できない。
rctladm(1M)	システム資源制御のグローバル状態を修正および表示する。
prctl(1)	動作中であるプロセス、タスク、プロジェクトの資源制御を取得及び設定する。

👉 Solaris 10、Solaris 11 では、sysdef(1M)コマンドから IPC パラメーター値を表示することはできません。代わりに資源制御の値を表示する prctl(1)コマンドから確認します。

3-1 .IPC パラメーターの確認

デフォルトで設定されている IPC パラメーター値を確認します。

1) ログインユーザに設定されているプロジェクト ID を確認します。

```
# id -p
uid=0(root) gid=0(root) projid=1(user.root)
```

✓ 上記では「プロジェクト名: user.root」「プロジェクト ID: 1」と確認できる。

2) プロジェクトの資源制御の設定値を確認する。

```
# projects -l user.root
user.root
    projid : 1
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
```

👁️ 「attribs:」欄に資源制御パラメーターが表示されます。空欄の場合はデフォルト値が定義されます。

3) IPC パラメーターを確認する。(※IPC 関連情報のみ抜粋。)

```
# prctl $$
process: 27627: -bash
NAME      PRIVILEGE      VALUE      FLAG      ACTION      RECIPIENT
process.max-msg-messages
  privileged 8.19K        -         deny      -
  system     4.29G        max       deny      -
process.max-msg-qbytes
  privileged 64.0KB       -         deny      -
  system     16.0EB       max       deny      -
process.max-sem-ops
  privileged 512          -         deny      -
  system     2.15G        max       deny      -
process.max-sem-nsems
  privileged 512          -         deny      -
  system     32.8K        max       deny      -
project.max-shm-memory
  privileged 15.3GB       -         deny      -
  system     16.0EB       max       deny      -
project.max-shm-ids
  privileged 128          -         deny      -
  system     16.8M        max       deny      -
project.max-msg-ids
  privileged 128          -         deny      -
  system     16.8M        max       deny      -
project.max-sem-ids
```

privileged	128	-	deny	-
system	16.8M	max	deny	-

✓ privileged の値 (VALUE) がデフォルト値

参考) IPC パラメータ単位に確認することも可能。

■ 書式

```
# prtcl -n [資源制御パラメーター名] [プロセス ID]
```

■ 実行例:

```
# prtcl -n process.max-msg-messages $$
process: 24687: bash
NAME      PRIVILEGE      VALUE      FLAG      ACTION      RECIPIENT
process.max-msg-messages
  privileged  8.19K          -          deny      -
  system     4.29G          max        deny      -
```

☞ \$\$はログインプロセスを意味します。

参考) プロセス単位にプロジェクト名を確認する方法。

■ 書式

```
# ps -efo [format]
```

☞ format には pid,project など取得したい情報を指定します。

■ 実行例: プロセス ID、プロジェクト名、プロジェクト ID、タスク ID、実行コマンドを表示

```
# ps -efo pid,project,projid,taskid,comm
PID  PROJECT  PROJID  TASKID  COMMAND
0    system   0        0      sched
5    system   0        0      zpool-rpool
6    system   0        0      kmem_task
1    system   0        1      /usr/sbin/init
2    system   0        0      pageout
3    system   0        0      fsflush
7    system   0        0      intrd
8    system   0        0      vmtasks
430  system   0        61     /usr/lib/efcode/sparcv9/efdaemon
11   system   0        1      /lib/svc/bin/svc.startd
13   system   0        1      /lib/svc/bin/svc.configd
146  system   0        23     /usr/lib/devfsadm/devfsadmd
42   system   0        3      /usr/sbin/dlmgmt
72   system   0        7      /lib/inet/ipmgmt
128  system   0        19     /usr/lib/utmpd
47   default  3        5      /lib/inet/netcfgd
76   system   0        8      /lib/crypto/kcfd
```

136	system	0	0	zpool-sol11
154	system	0	0	zpool-zones
:				

3-2. IPC パラメーターの設定

デフォルトで設定されている IPC パラメーター値を変更して、プロジェクト単位に設定します。設定にはプロジェクトデータベース(/etc/project)を編集します。

3-2-1. プロジェクトデータベースの編集方法

/etc/project ファイル編集用のコマンドを利用して資源制御パラメーターを設定します。

■ 書式

・プロジェクトを新規作成し資源制御パラメーターを設定する場合【projadd(1M)】

```
# projadd -K '[資源制御パラメーター名]=(privileged, [値], deny)' [プロジェクト名]
```

・既存プロジェクトに資源制御パラメーターを設定する場合【projmod(1M)】

```
# projmod -K '[資源制御パラメーター名]=(privileged, [値], deny)' [プロジェクト名]
```

・既存プロジェクトに資源制御パラメーターを追加設定する場合

```
# projmod -s -K '[資源制御パラメーター名]=(privileged, [値], deny)' [プロジェクト名]
```

☛ 「-s」オプションを付けることで、既に定義済みのパラメーターがある場合その値のみを変更します。

・プロジェクトを削除する場合【projdel(1M)】

```
# projdel [プロジェクト名]
```

■ 設定例

1) 新規にプロジェクトを追加し IPC 資源パラメーターの設定を行う。(例:user.guest)

```
# projadd -K 'project.max-shm-memory=(privileged, 1024, deny)' user.guest
```

2) その他の IPC 資源パラメーターを設定する。

```
# projmod -s -K 'project.max-shm-ids=(privileged, 1024, deny)' user.guest
# projmod -s -K 'project.max-msg-ids=(privileged, 1024, deny)' user.guest
# projmod -s -K 'project.max-sem-ids=(privileged, 1024, deny)' user.guest
# projmod -s -K 'process.max-sem-ops=(privileged, 1024, deny)' user.guest
# projmod -s -K 'process.max-msg-messages=(privileged, 1024, deny)' user.guest
# projmod -s -K 'process.max-msg-qbytes=(privileged, 1024, deny)' user.guest
```

3) 設定した IPC 資源パラメーターを確認する。

```
# projects -l user.guest
user.guest
  projid : 100
  comment: ""
  users  : (none)
  groups : (none)
  attribs: process.max-msg-messages=(privileged, 1024, deny)
           process.max-msg-qbytes=(privileged, 1024, deny)
           process.max-sem-ops=(privileged, 1024, deny)
           project.max-msg-ids=(privileged, 1024, deny)
           project.max-sem-ids=(privileged, 1024, deny)
           project.max-shm-ids=(privileged, 1024, deny)
           project.max-shm-memory=(privileged, 1024, deny)
```

参考)プロジェクトデータベース(/etc/project)からも設定の確認が可能です。

```
# more /etc/project
system:0:::
user.root:1:::
noproject:2:::
default:3:::
group.staff:10:::
user.guest:203:::process.max-msg-messages=(privileged, 1024, deny);process.max-msg-qbytes=(privileged, 1024, deny);process.max-sem-ops=(privileged, 1024, deny);project.max-msg-ids=(privileged, 1024, deny);project.max-sem-ids=(privileged, 1024, deny);project.max-shm-ids=(privileged, 1024, deny);project.max-shm-memory=(privileged, 1024, deny)
```

👉 /etc/project ファイルを直接編集してもパラメーター設定は有効になりますが、編集ミスしやすいためコマンドによる編集を推奨します。

3-2-2 .IPC パラメーターを有効にする方法

プロジェクトデータベース(/etc/project)に設定したパラメーターは、login(1),su(1M),at(1),cron(1),newtask(1)の何れかのコマンドが実行されると有効になります。

1) 現在の IPC 資源パラメーター値を確認する。(例: project.max-shm-ids パラメーター)

```
# prctl -n project.max-shm-ids $$

process: 16675: bash
NAME PRIVILEGE      VALUE  FLAG  ACTION  RECIPIENT
project.max-shm-ids
  privilege  128    -     deny   -
  system    16.8M max    deny   -
```

👉 '\$\$'は現在のシェルのプロセス ID。

2)現在のプロジェクト名を確認する。

```
# id -p
uid=0(root) gid=0(root) projid=1(user.root)
```

3)IPC パラメーターを設定する。

```
# projmod -K 'project.max-shm-ids=(privileged, 150, deny)' user.root

# projects -l user.root
user.root
    projid : 1
    comment: ""
    users  : (none)
    groups : (none)
    attribs: project.max-shm-ids=(privileged, 150, deny)
```

4)再ログインするか、下記コマンドを実行する。

```
# newtask -p user.root
```

5)変更後の IPC 資源パラメーター値を確認する。

```
# prctl -n project.max-shm-ids $$

process: 16680: bash
NAME      PRIVILEGE      VALUE  FLAG  ACTION      RECIPIENT
project.max-shm-ids
  privileged  150          -    deny      -
  system     16.8M       max   deny      -
```

3-2-3.デーモン起動時に IPC パラメーターを有効にする方法

OS 起動時に実行されるデーモンはシステムプロジェクト(system)で実行されます。但し、/etc/project の system プロジェクトに設定した IPC パラメーターはデーモン起動時には反映されません。デーモン起動時に system プロジェクトでプロセスを実行させるためには下記を実施します。

1)接頭辞が、project.~ のパラメーターの場合、デーモンを起動する SMF の method ファイルまたは RC スクリプトの最初に下記を追加する。

```
newtask -p system /usr/bin/true
```

- project レベルのパラメーターなので、一度の実行で、system プロジェクトから実行されるプロセス全てに /etc/project の値が適用されます。そのため、パラメーター設定したいデーモンが複数存在しても、いずれかのスクリプトファイルに設定するだけ十分です。

2) 接頭辞が、process.~ のパラメータの場合、デーモンを起動する SMF の method ファイルまたは RC スクリプトの最初に下記を追加する。

```
newtask -p system -c $$
```

- ☛ process レベルのパラメータなので、設定したスクリプトから発生するプロセスのみに適用されます。そのため、パラメータ設定したいデーモンが複数ある場合、すべてのスクリプトファイルに追加する必要があります。

3) 接頭辞が、project.~ のパラメータと process.~ のパラメータの両方を設定する場合は、上記 1)、2)の両方をスクリプトファイルへ記述します。

3-3 .IPC パラメーターの監視設定

システムの IPC 資源の利用が IPC パラメーターの値を超えた場合に /var/adm/messages にメッセージ出力する設定 (グローバルアクション) を行います。

■ 書式

```
# rctladm [-lu] [-e actions | -d actions] [rctl_name]
```

- l: イベント状態と利用可能な資源制御を表示する。
- u: /etc/rctladm.conf の内容を元に資源制御を構成する。
- e: 指定したグローバルアクションを活性化 (enable) する。
- d: 指定したグローバルアクションを非活性化 (disable) する。

rctl_name: 資源制御パラメーター名 (resource control name)

1) IPC 資源パラメーターのグローバル状態の設定を確認します。

```
# rctladm -l project.max-shm-memory
project.max-shm-memory          syslog=off      [ no-basic deny bytes ]
```

☞ デフォルトでは各資源の利用が資源制御パラメーターの設定値 (しきい値) に達してもメッセージを出さない設定 (syslog=off) になっています。

2) IPC 資源に関するエラー情報をシスログ (/var/adm/messages) へ出力する設定に変更します。

```
# rctladm -e syslog=err project.max-shm-memory
```

☞ syslog の後にメッセージレベルを設定しない場合は自動的に notice レベルとなります。

3) 変更を確認します。

```
# rctladm -l project.max-shm-memory
project.max-shm-memory          syslog=err      [ no-basic deny bytes ]
```

☞ err レベルのメッセージが出力されるよう syslog.conf の設定も確認してください。

必要に応じ、その他のパラメーターも同様に設定します。設定後、IPC 資源の監視が有効になります。

4) 設定されているグローバルアクションを確認する。

```
# rctladm
process.max-deferred-ports  syslog=off      [ lowerable deny count ]
process.max-sigqueue-size  syslog=off      [ lowerable deny count ]
process.max-itimers        syslog=off      [ deny count ]
process.max-port-events    syslog=off      [ deny count ]
process.max-msg-messages   syslog=off      [ deny count ]
process.max-msg-qbytes     syslog=off      [ deny bytes ]
process.max-sem-ops        syslog=off      [ deny count ]
process.max-sem-nsems      syslog=off      [ deny count ]
process.max-address-space  syslog=off      [ lowerable deny no-signal bytes ]
process.max-file-descriptor syslog=off      [ lowerable deny count ]
process.max-core-size      syslog=off      [ lowerable deny no-signal bytes ]
process.max-stack-size     syslog=off      [ lowerable deny no-signal bytes ]
```


process.max-data-size	syslog=off	[lowerable deny no-signal bytes]
process.max-file-size	syslog=off	[lowerable deny file-size bytes]
process.max-cpu-time	syslog=off	[lowerable no-deny cpu-time inf seconds]
task.max-cpu-time	syslog=off	[no-deny cpu-time no-obs inf seconds]
task.max-processes	syslog=off	[count]
task.max-lwps	syslog=off	[count]
project.max-contracts	syslog=off	[no-basic deny count]
project.max-locked-memory	syslog=off	[no-basic deny bytes]
project.max-port-ids	syslog=off	[no-basic deny count]
project.max-mrp-ids	syslog=off	[no-basic deny count]
project.max-shm-memory	syslog=err	[no-basic deny bytes]
project.max-shm-ids	syslog=off	[no-basic deny count]
project.max-msg-ids	syslog=off	[no-basic deny count]
project.max-sem-ids	syslog=off	[no-basic deny count]
project.max-crypto-memory	syslog=off	[no-basic deny bytes]
project.max-tasks	syslog=off	[no-basic count]
project.max-processes	syslog=off	[no-basic count]
project.max-lwps	syslog=off	[no-basic count]
project.cpu-cap	syslog=n/a	[no-basic deny no-signal inf no-syslog count]
project.cpu-shares	syslog=n/a	[no-basic no-deny no-signal no-syslog count]
zone.max-lofi	syslog=off	[no-basic deny count]
zone.max-swap	syslog=off	[no-basic deny bytes]
zone.max-locked-memory	syslog=off	[no-basic deny bytes]
zone.max-mrp-ids	syslog=off	[no-basic deny count]
zone.max-shm-memory	syslog=off	[no-basic deny bytes]
zone.max-shm-ids	syslog=off	[no-basic deny count]
zone.max-sem-ids	syslog=off	[no-basic deny count]
zone.max-msg-ids	syslog=off	[no-basic deny count]
zone.max-processes	syslog=off	[no-basic count]
zone.max-lwps	syslog=off	[no-basic count]
zone.cpu-cap	syslog=n/a	[no-basic deny no-signal inf no-syslog count]
zone.cpu-shares	syslog=n/a	[no-basic no-deny no-signal no-syslog count]

- ☛ 上記は、Oracle Solaris 11 環境での資源パラメータです。Oracle Solaris 11 では「process.max-deferred-ports」「task.max-processes」「project.max-mrp-ids」「project.max-processes」「zone.max-lofi」「zone.max-mrp-ids」「zone.max-processes」パラメータが新規に追加されています。
- ☛ Solaris 10 であった「project.max-device-locked-memory」資源パラメータは Solaris 11 では廃止されています。

4. パラメーターの変更方法

/etc/project ファイルの資源制御パラメーターを編集せずに設定値を変更します。実行中のプロセスに対して動的に設定値を変更します。

■ 書式

```
# prctl -n [資源制御パラメーター名] -r -v [設定値] [プロセス ID]
```

1) IPC パラメーターを確認します。(例: プロセス ID1234 の project.max-shm-ids パラメーター)

```
# prctl -n project.max-shm-ids 1234
process: 24687: bash
NAME  PRIVILEGE      VALUE  FLAG  ACTION  RECIPIENT
project.max-shm-ids
  privileged      128    -    deny   -
  system          16.8M  max   deny   -
```

2) IPC パラメーターを変更します。(例: パラメーター値を 512 に変更)

```
# prctl -n project.max-shm-ids -r -v 512 1234
```

👉 接頭辞が project.~ のパラメーターの場合、コマンド実行後、指定したプロセスが属するプロジェクトのプロセス全てに適用されます。

3) 変更を確認します。

```
# prctl -n project.max-shm-ids 1234
process: 24687: bash
NAME  PRIVILEGE      VALUE  FLAG  ACTION  RECIPIENT
project.max-shm-ids
  privileged      512    -    deny   -
  system          16.8M  max   deny   -
```

✓ prctl コマンドによるパラメーター値の変更は一時的な変更です。恒久的に変更したい場合は /etc/project ファイルの編集も同時に行ってください。

『Oracle Solaris IPC パラメーター設定手順書』以上

改版履歴

改版日時	版数	改版内容
2012.8	第 1.0 版	新規作成
2014.1	第 1.1 版	Oracle Solaris 11.1 に対応
2016.3	第 1.2 版	Oracle Solaris 11.3 に対応

FUJITSU

shaping tomorrow with you