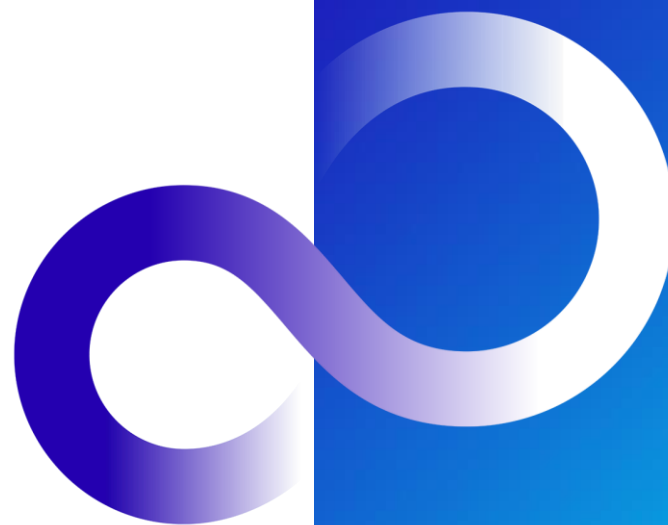


# Oracle Solaris IPCパラメーター 設定ガイド

2024年 4月

第1.7版

富士通株式会社



## ■ 内容

- 本書は、Oracle Solaris 10およびOracle Solaris 11におけるIPC (Inter Process Communication) パラメーターの設定に関する情報を記載しています。
  - Solaris 10, Solaris 11のIPCパラメーター設定方法やSolaris 9以前の設定との違い
  - 複数の製品を組み合わせた時の、IPCパラメーターの設定指針
  - OS稼働中に動的にパラメーターを変更する方法

## ■ 目的

- Oracle Solarisの資源管理について理解し、正しくIPCパラメーター設計を実施する。

## ■ 対象読者

- Oracle Solarisの基礎知識を有している方。

## ■ 留意事項

- Oracle Solaris は Solarisと表記する場合があります。
- 本資料は、Oracle Solaris 11.3に基づいています。
- ミドルウェア導入の際は各製品マニュアルに記載のIPC関連の項目も合わせて参照 してください。
- Oracle Solaris 11 に関する最新情報は、Oracle社のマニュアルをご参照ください。
  - Oracle Solaris 11 Documentation  
<http://www.oracle.com/technetwork/documentation/solaris-11-192991.html>

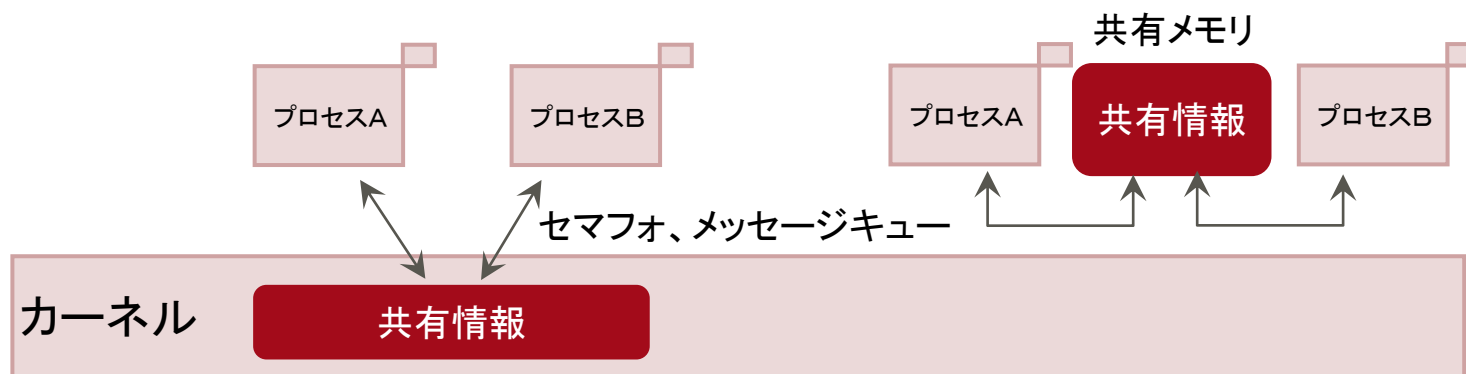
1. Oracle Solarisの資源管理
2. IPCパラメーター設定
3. IPCパラメーターの見積り
4. 運用時のパラメーター変更
5. 参考情報
6. 付録(IPCパラメーター見積り例)
7. 付録(IPCパラメーター見積り時の留意事項)
8. 付録(Oracle Solarisゾーン環境のIPCパラメーター)

# 1. Oracle Solaris の資源管理

## ■ IPC (Inter Process Communication)とは

IPCは「プロセス間通信」の意味であり、プロセスが仮想アドレス空間を介して、他のプロセスと情報の共有や、情報の受け渡しを行う仕組みです。IPCはプロセス間通信を行うための仕組みとして、Solarisでは共有メモリ(shared memory)、セマフォ(semaphore)、メッセージキュー(message queue)の3種類の資源があります。これら3種類の資源を制御するものがIPCパラメーターです。

### 《プロセス間の情報共有》



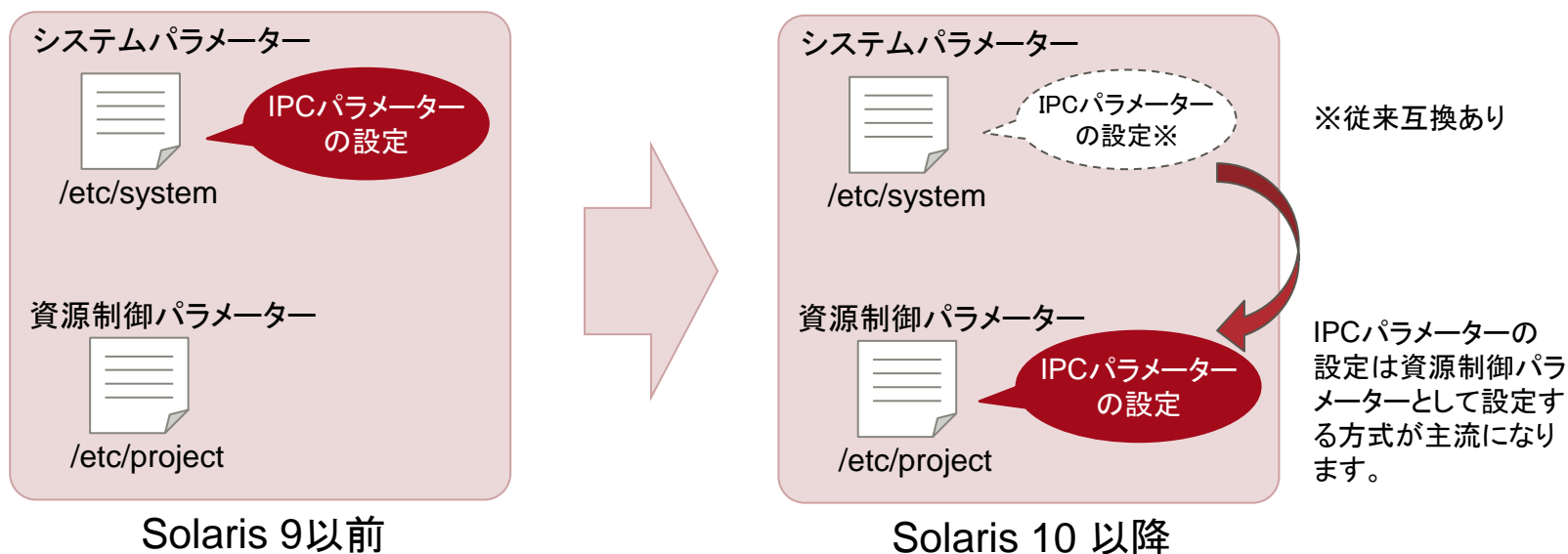
### 設定の必要性

Solaris OSではデフォルトでIPC資源の値が定義されていますが、導入するミドルウェアやアプリケーションには、正常動作に必要となるIPC資源の値が決められているため、OSデフォルト値では不足する場合があります。そのため、事前にミドルウェアやアプリケーションが必要とするIPC資源の設定値を見積もり、設定する必要があります。

## ■ Solaris 10 以降、IPCの制御は資源制御パラメーターとして設定します

IPCパラメーターはOracle Solaris 上で実行されるプロセスのIPCを制御するパラメーターです。

IPCパラメーターは、従来、システムパラメーター(/etc/systemファイル)の中で設定されていましたが、Solaris 10以降は、資源制御パラメーター(/etc/projectファイル)の中で設定が可能となりました。

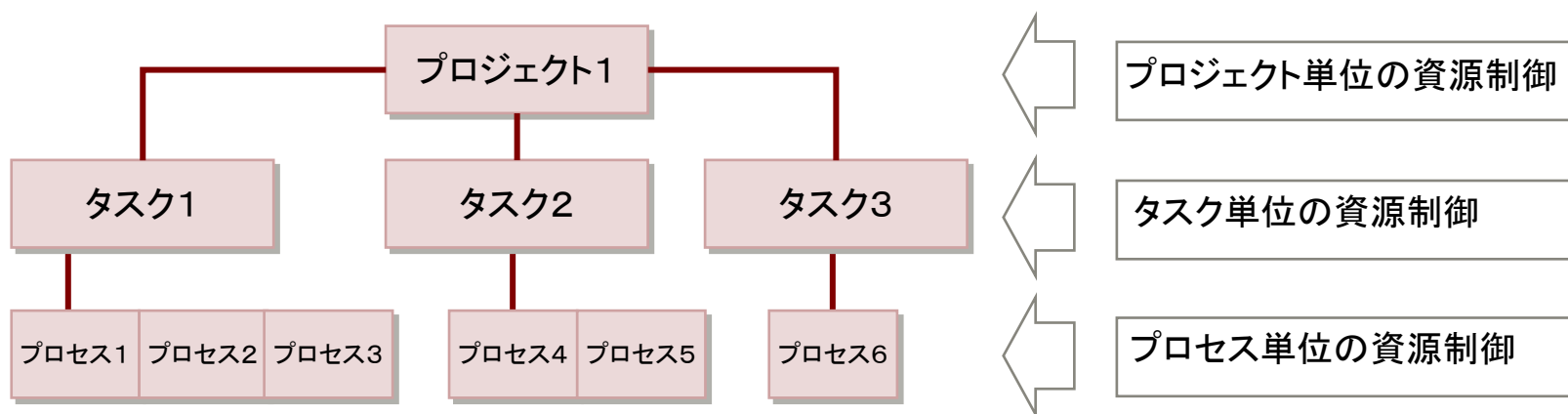


- ・資源制御は/etc/project ファイルへ「資源制御パラメーター」を追加することで実施します
- ・Solaris 10以降、資源制御パラメーターにIPC資源に関するパラメーターが含まれます

## ■ 資源制御はプロジェクト、タスクの管理下で実行されます

OS上で実行されるプロセスは、全て「プロジェクト」、「タスク」という識別子の下で管理されています。起動時のデーモンやユーザのプロセスは、必ずある「プロジェクト」に属し、実行されるプロセスは「タスク」でグループ化されています。管理者はこれらのプロジェクト、タスク、プロセスの各々の階層で資源制御を実施することができます。

### 《プロジェクトとタスクのツリー構造》



・プロジェクトやタスクなど上位の階層で設定される資源制御パラメーターは、その配下のプロセスに対して有効になります。(例: プロジェクト1に設定したパラメーターは全てのプロセスに影響します。)

## ■ プロジェクトには複数の種類があります

プロジェクトには下記の種類があり、OS上で実行されるプロセスは全て何れかのプロジェクトに属しています。

プロジェクト	project名	概要
システムプロジェクト	system	OS起動時のデーモンやcronのプロセスが属するプロジェクト
	user.root	rootユーザが属するプロジェクト
デフォルトプロジェクト	default	プロジェクトが未定義の一般ユーザが属するプロジェクト
ユーザ定義のプロジェクト (任意設定*)	user.[username]	[username]で定義されたユーザが属するプロジェクト
	group.[groupname]	[groupname]で定義されたグループのユーザが属するプロジェクト
	[projectname]	任意のユーザ、グループを属させることができるプロジェクト

[username][groupname]は/etc/passwd,/etc/groupで定義されている名前

(\*)インストールする製品によっては、製品が指定するproject名でプロジェクトを定義する必要があります。そのため、任意にproject名を定義する際は重複しない名前をつける必要があります。

各ユーザが所属するプロジェクトは下記の順に優先して判定されます

1. ユーザ拡張データベース (/etc/user\_attr) で指定したプロジェクト
2. user.[username]プロジェクト
3. group.[groupname]プロジェクト
4. default プロジェクト



制御範囲	資源パラメーター名	説明	備考
プロジェクト単位	project.max-shm-ids	共有メモリIDの最大数	IPCパラメーター
	project.max-sem-ids	セマフォIDの最大数	IPCパラメーター
	project.max-msg-ids	メッセージキューIDの最大数	IPCパラメーター
	project.max-shm-memory	共有メモリの合計量	IPCパラメーター
	project.cpu-shares	プロジェクトに対するFSSで使用するCPU配分	
	project.cpu-cap	プロジェクトで使用可能なCPUリソースの最大値	
	project.max-crypto-memory	暗号化処理で使用可能なカーネルメモリ量	
	project.max-device-locked-memory	ロックされるメモリの許容合計量	Solaris 10まで
	project.max-locked-memory	ロックされるメモリの許容合計量	
	project.max-port-ids	イベントポートの最大数	
	project.max-lwps	LWPの最大数	
	project.max-tasks	タスクの最大数	
	project.max-contracts	契約の最大数	
	project.max-processes	プロジェクト内で同時に使用できるプロセステーブルスロットの最大数	Solaris 11/11から
タスク単位	task.max-cpu-time	タスク内のプロセスが使用できる最長CPU時間	
	task.max-lwps	タスク内のプロセスが同時に使用できるLWPの最大数	
	task.max-processes	タスク内で同時に使用できるプロセステーブルスロットの最大数	Solaris 11/11から
プロセス単位	process.max-sem-nsems	セマフォセットあたりに許容されるセマフォの最大数	IPCパラメーター
	process.max-sem-ops	Semopコールに許容されるセマフォ操作の最大数	IPCパラメーター
	process.max-msg-qbytes	メッセージキュー内のメッセージの最大バイト数	IPCパラメーター
	process.max-msg-messages	メッセージキュー内のメッセージの最大数	IPCパラメーター
	process.max-cpu-time	プロセスが使用できる最長CPU時間	
	process.max-file-descriptor	プロセスが使用できる最大のファイル記述子インデックス	
	process.max-file-size	プロセスが書き込みに使用できる最大ファイルオフセット	
	process.max-core-size	プロセスによって作成されるコアファイルの最大サイズ	
	process.max-data-size	プロセスが使用できるヒープメモリの最大サイズ	
	process.max-stack-size	最大スタックメモリセグメント	
	process.max-address-space	セグメントサイズの総計としての最大アドレス空間	
	process.max-port-events	イベントポートあたりに許容されるイベントの最大数	
	process.max-itimers	インターバルタイマーの許容可能な最大数	Solaris 11.1から
	process.max-sigqueue-size	プロセスが保有できるキューに入れられた未処理のシグナルの最大数	Solaris 11.1から
	process.max-deferred-posts	遅延ポストの最大数	Solaris 11.2から

制御範囲	資源パラメーター名	説明	備考
ゾーン単位	zone.max-shm-memory	ゾーン内のプロセスが利用する共有メモリの合計量の上限	IPCパラメーター
	zone.max-shm-ids	ゾーン内のプロセスの共有メモリIDの合計数の上限	IPCパラメーター
	zone.max-sem-ids	ゾーン内のプロセスのセマフォIDの合計数の上限	IPCパラメーター
	zone.max-msg-ids	ゾーン内のプロセスのメッセージキューIDの合計数の上限	IPCパラメーター
	zone.max-swap	ゾーン内の目盛資源によるスワップ上限を設定	
	zone.max-locked-memory	ゾーン内のロックされるメモリの許容合計量	
	zone.max-lwps	ゾーン内のLWPの最大数	
	zone.cpu-cap	ゾーンで使用可能なCPUリソースの最大値	
	zone.cpu-shares	ゾーンに割り当てられるCPUリソースの配分比率	
	zone.max-processes	ゾーンで同時に使用できるプロセステーブルスロットの最大数	Solaris 11/11から
	zone.max-lofi	ゾーン内に作成可能なlofiデバイスの最大数	Solaris 11/11から

- ・Solaris ゾーン単位の資源パラメーターのうち、「zone.max～」パラメーターはzone内のプロセスが使用する資源の合計値を制限したいときに設定します。
- ・Solaris ゾーン内のミドルウェア用のIPCパラメーターの設定は、ゾーン内のプロジェクト単位（またはプロセス単位）のIPCパラメーターを使用する必要があります。

## 2. IPCパラメーター設定

# Oracle Solaris環境でのIPCパラメーターの設定 1/7 FUJITSU

## ■ IPC資源の設定方法の変更

Oracle Solaris 10およびOracle Solaris 11のIPCパラメーターの設定方法は、従来の/etc/system に記載する方法に加えて、/etc/project ファイルに記載する方法があります。

### 《Oracle Solaris のバージョンによるIPCパラメーター設定方法》

設定ファイル	Solaris 9以前	Solaris 10以降	備考
/etc/system	有効	有効	・設定後、OS再起動が必要 ・システム全体に適用
/etc/project	無効	有効	・設定後、OS再起動は不要 ・設定したプロジェクトのみ適用 ・Solaris 9で設定可能なIPCパラメーターは無し

### Oracle Solaris ゾーン環境のIPC設定

Oracle Solaris ゾーン環境では、/etc/systemファイルが存在しません。  
Oracle Solaris ゾーン環境におけるIPCパラメーターの設定は/etc/project のみです。

# Oracle Solaris環境でのIPCパラメーターの設定 2/7 FUJITSU

## ■ /etc/systemの動作

OS起動時にカーネルパラメーターに設定され、OS上で実行されるプロセスは全て同じ設定値になります。

OS起動

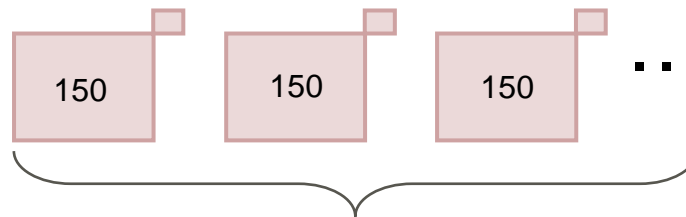


/etc/system



shmsys:shminfo\_shmmni=150

プロセス



カーネル shmsys:shminfo\_shmmni=150

shmsys:shminfo\_shmmniパラメーターは/etc/system に共有メモリIDの最大数を設定するIPCパラメーター

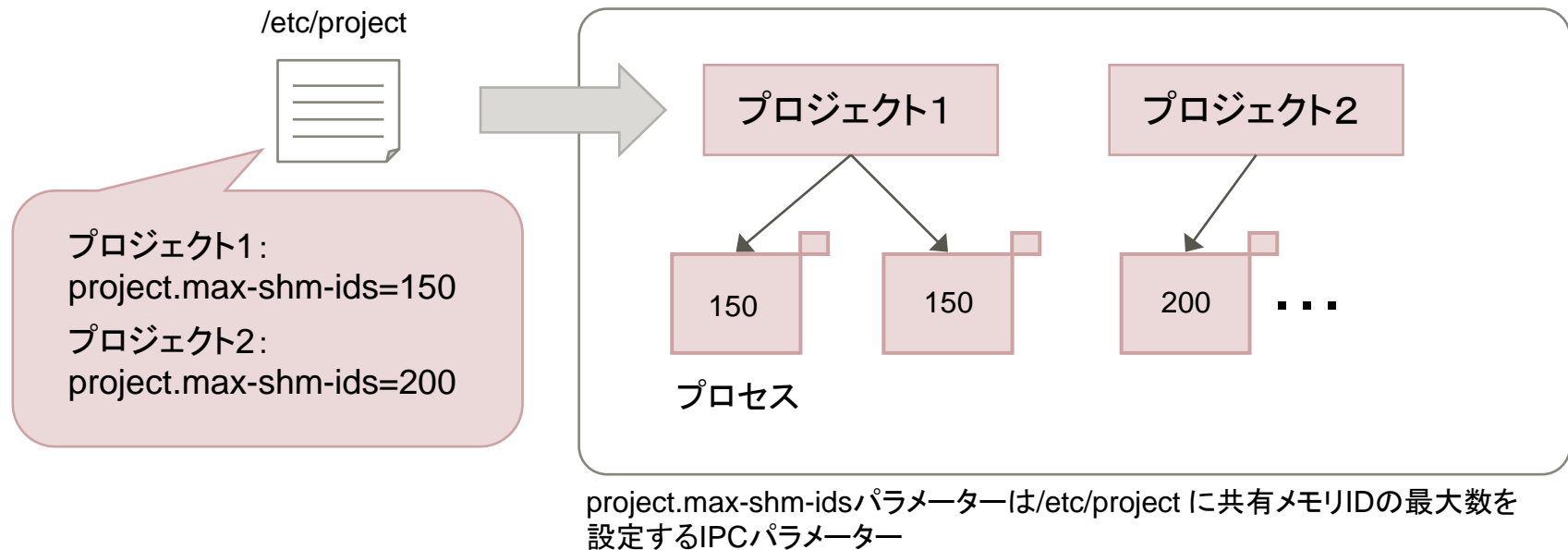
### /etc/system による設定の特長

パラメーター変更時にはOSの再起動が必要となります。

# Oracle Solaris環境でのIPCパラメーターの設定 3/7 FUJITSU

## ■ /etc/projectの動作

プロジェクト毎にパラメーター値が設定されるので、プロセスの設定値はプロジェクト毎に異なります。



### /etc/project による設定の特長

IPCパラメーター値の設定変更後、OSを再起動(リブート)せずに有効化が可能です。また、IPC資源はプロジェクト単位に設定します。

# Oracle Solaris環境でのIPCパラメーターの設定 4/7 FUJITSU

## ■ /etc/system と /etc/project 混在時の考え方

### 1) プロジェクト(/etc/project)、カーネルパラメーター(/etc/system) 両方に設定値がある場合

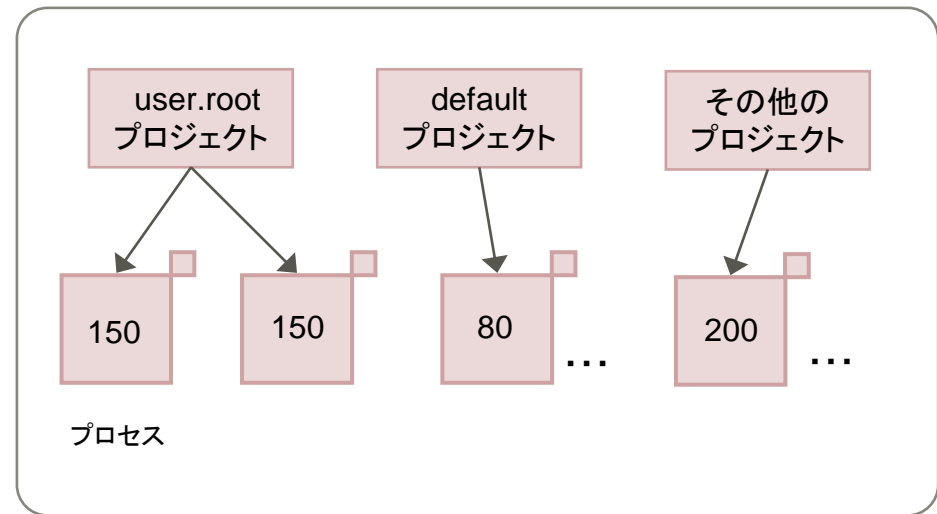
- ➡ パラメーターが設定されたプロジェクトから発生するプロセスは、**カーネルパラメーターの設定値に関わらずプロジェクトの設定値が優先**されて設定されます。
- ➡ パラメーターが設定されていないプロジェクトから発生するプロセスは、**カーネルパラメーターの設定値が有効になります**。但し、Solarisデフォルト値より小さい場合は、Solarisデフォルト値が設定されます。

#### /etc/project の設定

例 user.rootプロジェクト:  
project.max-shm-ids=150  
defaultプロジェクト:  
project.max-shm-ids=80  
その他のプロジェクト:  
project.max-shm-ids 設定なし

#### /etc/system の設定

例 カーネルパラメーター  
shmsys:shminfo\_shmmni=200



(\*)project.max-shm-ids と shmsys:shminfo\_shmmniは共有メモリIDの最大数を設定するIPCパラメーター。  
Solarisデフォルト値は128。

# Oracle Solaris環境でのIPCパラメーターの設定 5/7 FUJITSU

## 2) カーネルパラメーター(/etc/system)にのみ設定値がある場合

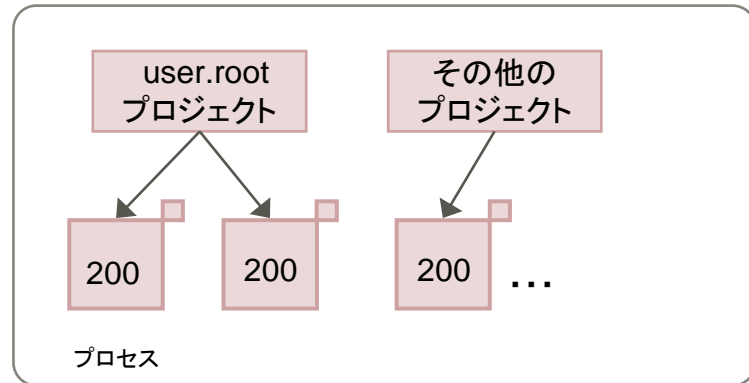
➡ Solarisデフォルト値より大きい場合、カーネルパラメーター(/etc/system)の値に従います。

### /etc/project の設定

例 user.rootプロジェクト:  
project.max-shm-ids 設定なし  
その他のプロジェクト:  
project.max-shm-ids 設定なし

### /etc/system の設定

例 カーネルパラメーター  
shmsys:shminfo\_shmmni=200



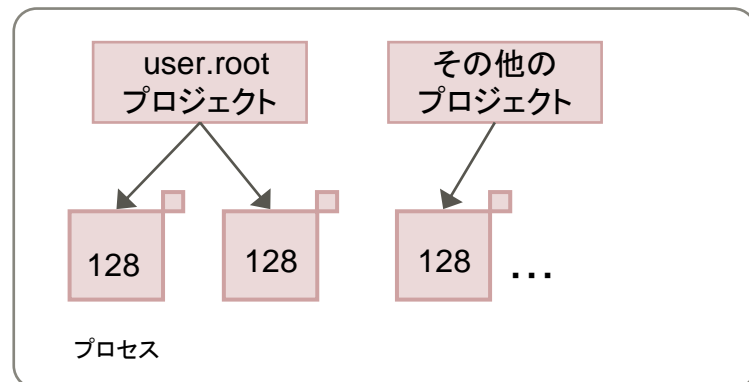
➡ 但し、Solarisデフォルト値より小さい場合は、Solarisデフォルト値が設定されます。

### /etc/project の設定

例 user.rootプロジェクト:  
project.max-shm-ids 設定なし  
その他のプロジェクト:  
project.max-shm-ids 設定なし

### /etc/system の設定

例 カーネルパラメーター  
shmsys:shminfo\_shmmni=80





## 3) プロジェクト(/etc/project)にのみ設定値がある場合

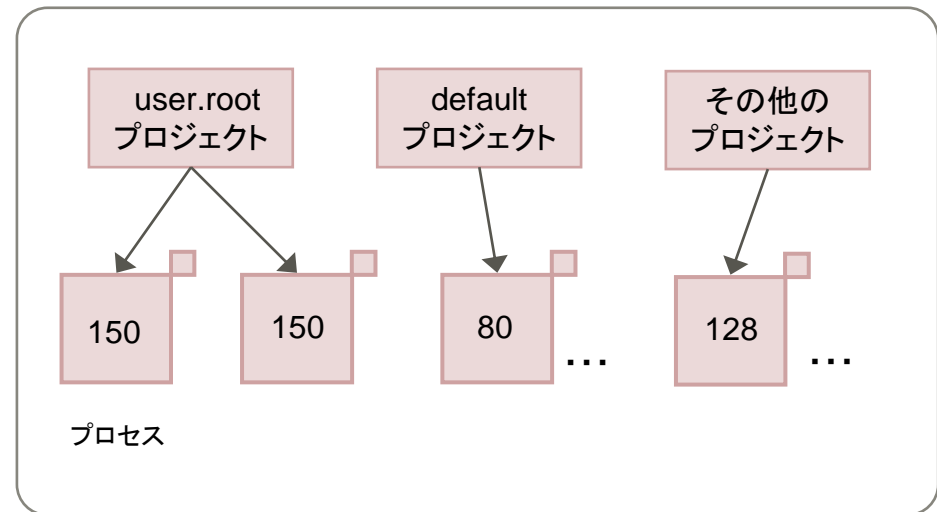
- ➡ パラメーターが設定されたプロジェクトから発生するプロセスは、プロジェクトの設定値が採用されます。Solarisデフォルト値よりも小さい値の設定も有効になります。
- ➡ パラメーターが設定されていないプロジェクトから発生するプロセスは、Solarisデフォルト値が設定されます。

### /etc/project の設定

例 user.rootプロジェクト:  
project.max-shm-ids=150  
defaultプロジェクト:  
project.max-shm-ids=80  
その他のプロジェクト:  
project.max-shm-ids 設定なし

### /etc/system の設定

例 カーネルパラメーター  
shmsys:shminfo\_shmmni 設定なし



## ■ /etc/system と /etc/project 混在時の設定値の例

### 覚えておくべきポイント！

- 1. /etc/project の設定値が/etc/system の設定値よりも常に優先される。
- 2. /etc/project に設定がない場合は/etc/system に従うが、デフォルト値未満には設定されない。

例) あるパラメーターのデフォルト値を100として、/etc/system、/etc/project両方にパラメーター設定した時の有効値

	Solaris デフォルト値	/etc/system	/etc/project (例: user.rootに 設定した場合)
パターン1	100	120	未設定
パターン2	100	80	未設定
パターン3	100	120	150
パターン4	100	80	150
パターン5	100	80	60
パターン6	100	未設定	150
パターン7	100	未設定	60



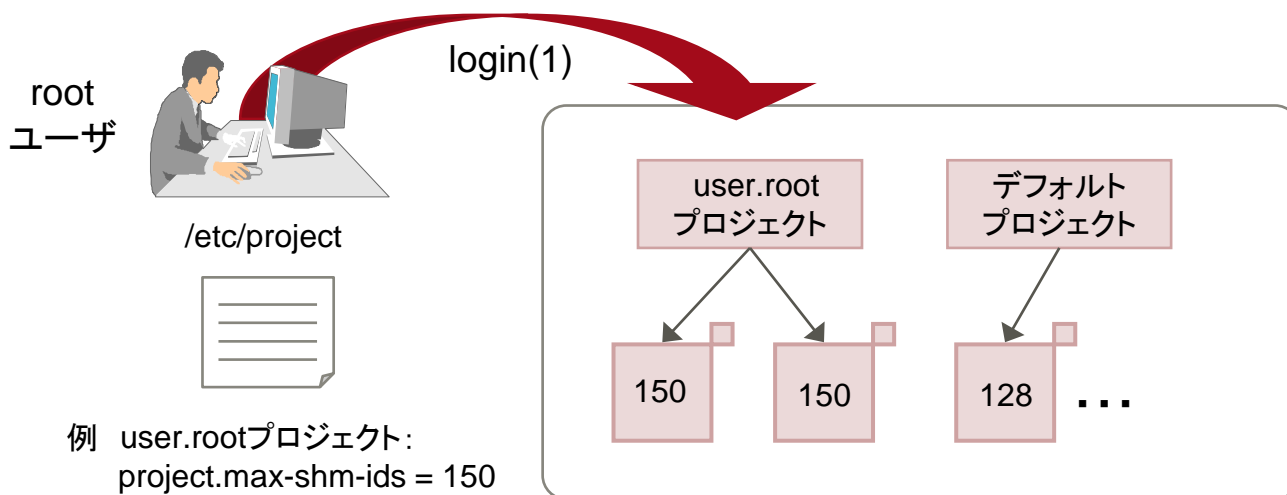
有効になる設定値	
user.root の プロセス	user.root 以外の プロセス
120	120
100	100
150	120
150	100
60	100
150	100
60	100

## ■ プロジェクトに設定したパラメーターはログイン or 特定コマンド実行後に有効になります

/etc/project で指定した値が有効になるのは、下記の何れかのコマンドが実行されるタイミングです。

・ユーザログイン時	login(1)
・ユーザ変更時	su(1M)
・ジョブ実行時	at(1)、cron(1)
・タスク変更時	newtask(1)

例えば、user.rootプロジェクトにパラメーターが設定されている状態で、rootユーザがログインすると実行されるシェルやプロセスは全てuser.rootプロジェクトのパラメーター値で実行されます。



## ■ システム起動中に起動するデーモンのパラメーター設定

SMFやRCスクリプトによりシステム起動中に実行されるデーモンプロセスは、「systemプロジェクト」で動作します。しかし、前述のコマンドを実行するまでは、/etc/project のsystemプロジェクトに設定した値は有効にならない、という問題があります。

### 《回避策》

デーモンを起動するmethodファイルやRCスクリプトの最初にnewtask(1)を実行するように編集します。設定するパラメーターによって記述の方法が異なります。

- 1) project.~パラメーターの場合、下記をスクリプトの先頭に追加する  
newtask -p system /usr/bin/true
- 2) process.~パラメーターの場合、下記をスクリプトの先頭に追加する  
newtask -p system -c \$\$

### 留意事項

- 1)の実行は一度の実行でsystemプロジェクトから実行される全てのプロセスに設定されるため、起動順番が最初のスクリプト内にのみ記述します(複数のスクリプトに記述する必要はありません)。
- 2)の実行はプロセス単位に設定されるため、目的とするデーモンプロセスを起動するスクリプト全てに記述する必要があります。

- ・システムプロジェクト(system, user.root)は、システムのデーモンや root権限で動作するプロセスのためのプロジェクトです。このため、システム全体の動作に影響を与える可能性があるため、システムプロジェクトへIPCなどの資源制御パラメーターを設定する場合は、十分な注意と検証が必要です。
- ・特に問題となるのは、Solaris のデフォルト値より小さい値の場合です。  
Solaris のデフォルトの値よりも小さな値に設定した場合、通常の動作でも資源不足や性能劣化を引き起こす可能性があります。

## 【例えば】

Solarisデフォルトの project.max-shm-ids の値は128です。これよりも小さな値(例えば10)を user.root プロジェクトに設定した場合、root ユーザが実行するプロセス全ての project.max-shm-ids のパラメーター値が10になってしまいます。

そのため、システムプロジェクトに IPCパラメーターを設定する場合は、Solaris デフォルト値よりも小さくならないようにしてください。**「IPCパラメーターの見積り(P22以降)」参照**  
やむを得ずパラメーター値をデフォルト値よりも小さくする必要がある場合には、他のプロセスやシステム全体に影響がないことを十分に検証する必要があります。

## ■ IPCパラメーターの設定

- ・資源制御機能は、project データベース (/etc/project) によって構成されます。
- ・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設定します。
- ・各資源制御に対応付けられる値は、括弧で囲まれ、コンマ区切りのテキストとして表示されます。

### 《設定例 (/etc/projectファイル)》

```
user.guest:100:::project.max-shm-memory=(privileged, 524288000,deny); project.max-shm-ids(privileged,1024,deny)
```

(A)

(B)

(C)

(D)

(E)

(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 (スーパーユーザー) だけが変更できる

system (システム値) — オペレーティングシステムによる処理が実行されている間は、固定される

(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できる

### 留意事項

- ・IPCパラメーター設定時、特にマニュアル等に指示がない場合は、特権レベルに privileged、アクションには deny を設定してください。
- ・/etc/projectファイルの編集は直接ファイル編集しても有効になりますが、オペミスを防ぐためコマンド (project(1)、projadd(1M)、projmod(1M)) による編集を推奨します。

## 3. IPCパラメーターの見積り

## ①/etc/system または システムプロジェクトに設定する場合

/etc/system の設定値の製品と、システムプロジェクト(system、user.root)の設定値の製品が混在している場合、他方を加算することで資源不足に陥らないようにします。IPCパラメーターの種類が **Σ型** か **MAX型** によって下記のように設定してください。

### Σ型の場合：

- /etc/system およびシステムプロジェクトの設定値には、以下の各値の合計値を設定して下さい。
  - /etc/systemに設定する製品の設定値
  - /etc/project のシステムプロジェクトに設定する製品の設定値
  - /etc/projectのユーザ定義プロジェクトに設定する製品の設定値
  - Solaris デフォルト値

### MAX型の場合：

- /etc/system の設定値の製品とシステムプロジェクトの設定値の製品の中での最大値を設定して下さい。
- ただし製品の最大値がSolaris デフォルト値より小さい場合、Solaris デフォルト値を採用して下さい。

※Solarisデフォルト値を忘れずに計算してください。

### 留意事項

- ・ process.max-sem-nsems に大きな値を設定すると性能面で問題が出る可能性があります。詳細は7章を参照してください。
- ・ システムプロジェクトのprocess.max-sem-nsemsに大きな値を要求する製品がある場合があります。性能面で問題が出る可能性もあるので、できるだけアプリケーション毎、業務毎にプロジェクトを分けて、個別にセマフォ関連の IPC資源を設定するようにしてください。

但し、上記は共有メモリ(project.max-shm-memory,shmsys:shminfo\_shmmax)以外のパラメーターの考え方です。共有メモリについては次ページを参照してください。



共有メモリ (project.max-shm-memory, shmsys:shminfo\_shmmax) の見積りについては、下記の考え方になります。

## project.max-shm-memory、shmsys:shminfo\_shmmax の場合：

- システムプロジェクトの設定値(project.max-shm-memory)には、以下の各値の合計値を設定して下さい。  
/etc/system に設定する製品が指定する shmsys:shminfo\_shmmax と shmsys:shminfo\_shmmni の積  
/etc/project のシステムプロジェクトに設定する製品が指定する project.max-shm-memory の値  
/etc/project のユーザ定義プロジェクトに設定する製品が指定する project.max-shm-memory の値  
Solaris デフォルト値  
※デフォルト値は物理メモリの約1／4の値ですが、正確な値は以下のコマンドで確認してください。  
# prctl \$\$
- /etc/systemの設定値(shmsys:shminfo\_shmmax)には、上記(システムプロジェクトの設定値)で算出した合計値を shmsys:shminfo\_shmmni で割った値を設定してください。

※Solarisデフォルト値を忘れずに計算してください。

## ②ユーザ定義プロジェクト(製品専用のプロジェクト)に設定する場合

ユーザ定義のプロジェクトの場合はプロジェクト単位に合計値、最大値を算出してください。

### Σ型の場合:

- 以下を考慮してプロジェクトのパラメーター値を設定してください。

Solarisデフォルト値 +  $\Sigma$ (そのプロジェクトで動作する製品の合計)

### MAX 型の場合:

「そのプロジェクトで動作する製品の中で最大の値」を設定する。

※Solarisデフォルト値以下となっても問題ありません。

### 留意事項

ユーザ定義プロジェクトにおけるMAX型パラメーターは、算出の結果が「Solarisデフォルト値」以下となっても「そのプロジェクトで動作する製品の中で最大の値」を設定します。(システムプロジェクトにおける設定指針とは異なります。)

## ■ 見積り例

複数製品が混在する環境での見積り例です。

パラメーター		製品A の値	製品B の値	製品C の値	製品D の値	製品E の値	それぞれのパラメーターに設定する 値	
							Σ型	MAX型
/etc/system		100	200	—	—	—	$A+B+C+D+E+d$ $=1500+d$	400 または d の大きい値
/etc/project	システムプロジェクト (system、user.root)	—	—	300	400	—	$A+B+C+D+E+d$ $=1500+d$	400 または d の大きい値
	ユーザ定義プロジェクト	—	—	—	—	500	$E+d=500+d$	500

d = Solaris OSのデフォルト値(次ページ参照)

IPC資源は/etc/system または/etc/project へ各々の記述方法でIPCパラメーターを設定します。しかし、OSに導入する製品(ミドルウェア、アプリケーション)によっては、/etc/system、/etc/project のどちらかのみで有効になる場合があるので、混在する場合には注意が必要です。

## 留意事項

上記の見積り方法はIPCパラメーターについての考え方であり、/etc/project に設定可能なIPC以外のパラメーターについては、マニュアル等に記載の値に従ってください。

## ■ 見積り例(共有メモリの場合)

複数製品が混在する環境での見積り例です。

パラメーター			製品A の値	製品B の値	製品C の値	製品D の値	製品E の値	設定する値
/etc/system	shmsys:shminfo_shmmax		A1	B1	—	—	—	$(A1 \cdot A2 + B1 \cdot B2 + C1 + D1 + E1 + d1) / s2$ ※
	shmsys:shminfo_shmmni		A2	B2	—	—	—	$A2 + B2 + C2 + D2 + E2 + d2 = s2$
/etc/project	システムプロジェクト (system、user.root)	project.max-shm-memory	—	—	C1	D1	—	$A1 \cdot A2 + B1 \cdot B2 + C1 + D1 + E1 + d1$
		project.max-shm-ids	—	—	C2	D2	—	$A2 + B2 + C2 + D2 + E2 + d2 = s2$
	ユーザ定義プロジェクト	project.max-shm-memory	—	—	—	—	E1	$E1 + d1$
		project.max-shm-ids	—	—	—	—	E2	$E2 + d2$

d1 = project.max-shm-memory のデフォルト値

d2 = project.max-shm-ids のデフォルト値

※ shmsys:shminfo\_shmmax の“s2”の値は、shmsys:shminfo\_shmmni から算出します。

## ■ 資源制御パラメーター (/etc/project 用)、システムパラメーター (/etc/system 用)

分類	パラメーター種	新旧パラメーター	定義場所	パラメーター名	デフォルト値	単位	特性	パラメーター意味	見積り方法
共用メモリ関連	共用メモリの総バイト数	旧	システムパラメーター (/etc/system 用)	shmsys:shminfo_shmmax	8388608	Byte	型なし	セグメント単位の共用メモリサイズ (project.max-shm-memoryに設定される共用メモリの総バイト数を算出するために使用される)	<p>本書の「IPC資源の見積り方法 5/5」の例に従って見積もる。</p> <p>&lt;注意事項&gt; 同一製品で、新パラメーター (max-shm-memoryの値) と旧パラメーター (shminfo_shmmax * shminfo_shmmniの値) の両者が定義されていて、かつ両者の値が異なっている場合、そのプロジェクトに関しては新パラメーター (max-shm-memoryの値) が採用される。</p> <p>(次ページへ続く)</p>
		新	資源制御パラメーター (/etc/project 用)	project.max-shm-memory	<p>約1/4物理メモリ</p> <p>※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。</p> <p>(補足) ただし、旧パラメーター (shminfo_shmmax またはshminfo_shmmni) が /etc/system に設定されていて、かつ (shminfo_shmmax) * (shminfo_shmmni) の値が実メモリの1/4より大きい場合はその値がデフォルト値として採用される。</p>	Byte	Σ型	システム全体の共用メモリサイズ	
	共用メモリ識別子の最大数	旧	システムパラメーター (/etc/system 用)	shmsys:shminfo_shmmni	100	個数	Σ型	1セグメントあたりの共用メモリIDの最大数 (project.max-shm-memoryに設定される共用メモリの総バイト数を算出することにも使用される。)	
		新	資源制御パラメーター (/etc/project 用)	project.max-shm-ids	128	個数	Σ型	システム全体の共用メモリIDの最大数	

(続き)

分類	パラメーター種	新旧パラメーター	定義場所	パラメーター名	デフォルト値	単位	特性	パラメーター意味	見積り方法
セマフォ関連	共用セマフォIDの最大数	旧	システムパラメーター (/etc/system用)	semsys:seminfo_semmni	128	個数	Σ型	システムに許可された共用セマフォIDの最大数	本書のIPC資源の見積もり方法の「Σ型」の見積もり方法に従って見積もる。
		新	資源制御パラメーター (/etc/project用)	project.max-sem-ids				プロジェクト毎に許可された共用セマフォIDの最大数	
	セマフォ毎に許可された共用セマフォIDの最大数	旧	システムパラメーター (/etc/system用)	semsys:seminfo_semmsl	512	個数	MAX型	システムのセマフォ毎に許可された共用セマフォIDの最大数	本書のIPC資源の見積もり方法の「MAX型」の見積もり方法に従って見積もる。
		新	資源制御パラメーター (/etc/project用)	process.max-sem-nsems				セマフォセットあたりに許容されるセマフォの最大数	
	semopコール毎に許可された共用セマフォIDの最大数	旧	システムパラメーター (/etc/system用)	semsys:seminfo_semopm	512	個数	MAX型	semopコール毎に許可された共用セマフォIDの最大数	本書のIPC資源の見積もり方法の「MAX型」の見積もり方法に従って見積もる。
		新	資源制御パラメーター (/etc/project用)	process.max-sem-ops				semopコール毎に許可された共用セマフォIDの最大数	
メッセージキュー関連	プロジェクトに許可されたメッセージキューの最大数	旧	システムパラメーター (/etc/system用)	msgsys:msginfo_msgmni	128	個数	Σ型	システムに許可されたメッセージキューの最大数	本書のIPC資源の見積もり方法の「Σ型」の見積もり方法に従って見積もる。
		新	資源制御パラメーター (/etc/project用)	project.max-msg-ids				プロジェクトに許可されたメッセージキューの最大数	
	メッセージキュー上のメッセージの最大バイト数	旧	システムパラメーター (/etc/system用)	msgsys:msginfo_msgmnb	65536	Byte	MAX型	システムのメッセージキュー上のメッセージの最大バイト数	本書のIPC資源の見積もり方法の「MAX型」の見積もり方法に従って見積もる。
		新	資源制御パラメーター (/etc/project用)	process.max-msg-qbytes				プロジェクトのメッセージキュー上のメッセージの最大バイト数	
	メッセージキューにあるメッセージの最大数	旧	システムパラメーター (/etc/system用)	msgsys:msginfo_msgtql	8192	個数	MAX型	システムのメッセージキューにあるメッセージの最大数	本書のIPC資源の見積もり方法の「MAX型」の見積もり方法に従って見積もる。
		新	資源制御パラメーター (/etc/project用)	process.max-msg-messages				システムのメッセージキューにあるメッセージの最大数	

# 《参考》Oracle Solaris 10から削除されたIPCパラメーター FUJITSU

旧パラメーター名	デフォルト値	最大値	備 考
msgsys:msginfo_msgmax	2048	2,147,483,647(32Bit) 18,446,744,073,709,551,615(64Bit)	
semsys:seminfo_semmns	60	2,147,483,647	
semsys:seminfo_semmvmx	32767	65,535	
semsys:seminfo_semmnu	30	2,147,483,647	
semsys:seminfo_semaem	16384	32,767	
semsys:seminfo_semume	10	2,147,483,647	

## 《参考》Solaris 9以前に削除されたIPCパラメーター

旧パラメーター名	デフォルト値	最大値	備 考
semsys:seminfo_semusz	96	2,147,483,647	Solaris 2.6 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_semmap	10	2,147,483,647	Solaris 8 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

### 留意事項

上記パラメーターが /etc/system ファイルに記述された場合、そのパラメーターの設定は無視されます。

## 4. 運用時のパラメーター変更



## ■ プロジェクトによるIPC設定方式ではパラメーター値の動的変更が可能

プロジェクト(/etc/project)に設定したIPCパラメーターは、OS起動中に実行中のプロセスのIPCパラメーターの変更が可能です。変更方法は2つの方法がありますが、パラメーターの種類によって動作が異なるので注意が必要です。

### 1. newtask(1)コマンドによる変更(永続的な変更)

- 1) /etc/projectファイルのパラメーター定義を変更する
- 2) newtaskコマンドを実行する

#### ⇒ process.~ パラメーター変更の場合

# newtask -p [project] -c [pid] を実行する。指定したプロセスのみパラメーター変更されます。

#### ⇒ project.~ パラメーター変更の場合

# newtask -p [project] を実行する。プロジェクトのプロセス全てのパラメーターが変更されます。

### 2. prctlコマンドによる変更(一時的な変更)

- 1) prctl コマンドで直接変更する

#### ⇒ process.~ パラメーター変更の場合

# prctl -n [process.~] -r -v [値] [pid] を実行する。指定したプロセスのみパラメーター変更されます。

#### ⇒ project.~ パラメーター変更の場合

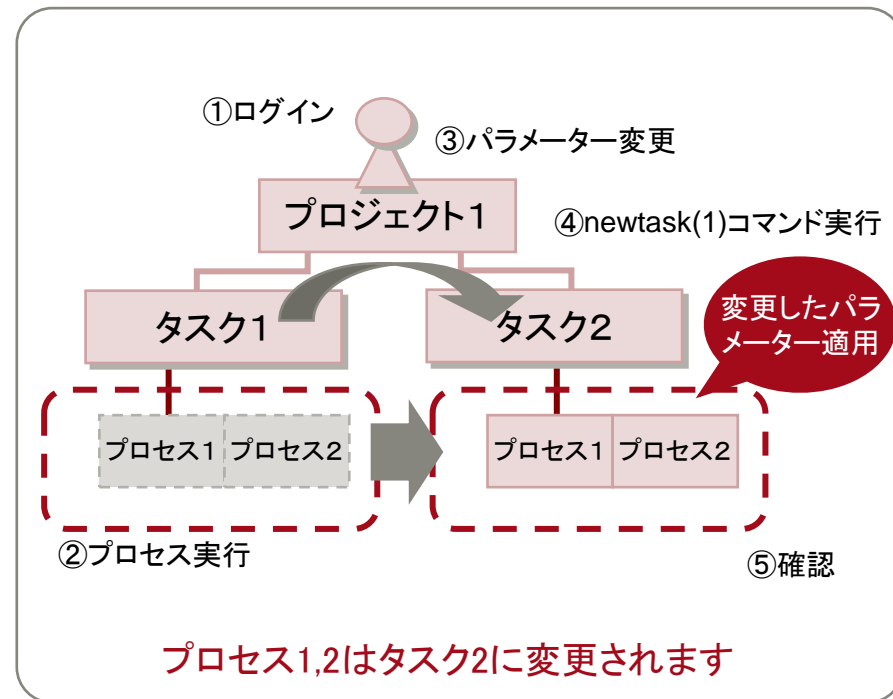
# prctl -n [project.~] -r -v [値] [pid] を実行する。プロジェクトのプロセス全てのパラメーターが変更されます。

## ■ 1. newtask(1)コマンドによる変更(永続的な変更)

プロジェクト(/etc/project)ファイルを修正しnewtaskコマンドを実行することで、実行中のプロセスのIPCパラメーター値の変更が可能です。

### 《パラメーター変更の流れ》

- ①ログイン (例:user.root)
- ②プロセス実行  
-etc/project に設定されたuser.root のパラメーター値が適用されます
- ③パラメーター変更  
-projadd、projmodコマンドで/etc/project を編集します
- ④newtaskコマンド実行  
-編集後の/etc/project のパラメーター値を反映させます
  - 1) project.~パラメーターの変更の場合  
# newtask -p user.root
  - 2) process.~パラメーターの変更の場合  
# newtask -p user.root -c [pid]  
※pidは実行中のプロセスを事前に確認
- ⑤変更後のパラメーター確認  
#prctl -n [project.~]or[process.~] [pid]



### 留意事項

systemプロジェクトから実行されるデーモンプロセスのパラメーター変更の場合は、上記③、④を実行します

## ■ 2. prctl(1)コマンドによる変更(一時的な変更)

プロジェクト(/etc/project)ファイルを編集せずに、実行中のプロセスのIPCパラメーター変更が可能です。

### 《パラメーター変更の流れ》

①ログイン (例:user.root)

②プロセス実行

-/etc/project に設定されたuser.root のパラメーター値が適用されます

③prctlコマンド実行

1) project.~パラメーターの変更の場合

# prctl -n [project.~] -r -v [値] [pid]

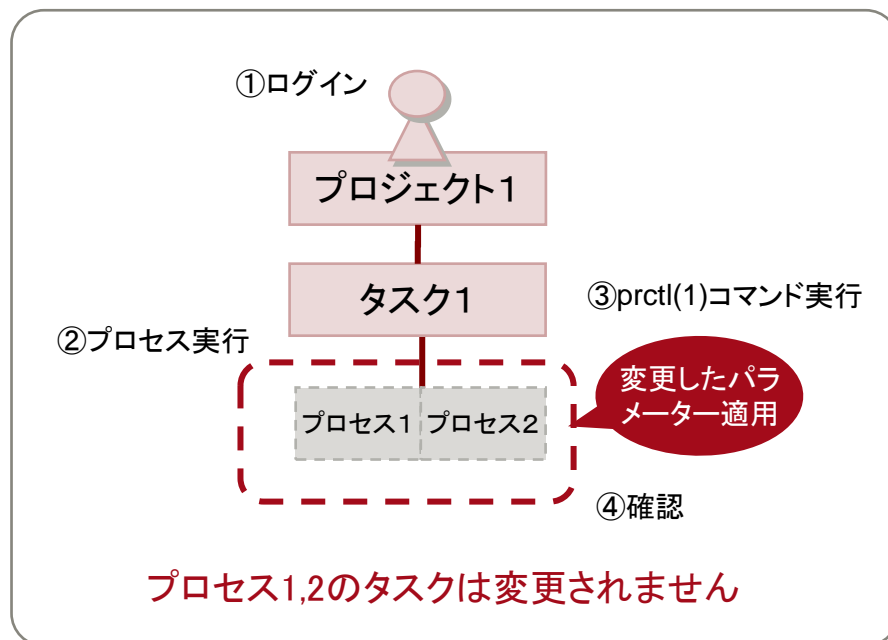
2) process.~パラメーターの変更の場合

# prctl -n [process.~] -r -v [値] [pid]

※pidは実行中のプロセスを事前に確認

④変更後のパラメーター確認

#prctl -n [project.~]or[process.~] [pid]



### 留意事項

上記方法でのパラメーター変更後に、login(1)やnewtask(1)などのコマンドがOS上で実行されると、再度/etc/projectの設定値が読み込まれるため、永続的に変更したい場合は/etc/projectを修正する前ページの方法を行います。

## ■ IPCパラメーターの設定に必要なコマンド

コマンド名	説明
projects(1)	ユーザーのプロジェクトメンバーシップを表示する。
newtask(1)	ユーザー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。

## 5. 参考情報

- Oracle Solaris カーネルのチューンアップ・リファレンスマニュアル
  - Oracle Solaris 10  
[http://docs.oracle.com/cd/E24845\\_01/html/819-0376/index.html](http://docs.oracle.com/cd/E24845_01/html/819-0376/index.html)
  - Oracle Solaris 11.3  
[http://docs.oracle.com/cd/E62101\\_01/html/E62779/index.html](http://docs.oracle.com/cd/E62101_01/html/E62779/index.html)

## 6. 付録（IPCパラメーター見積り例）

## ■ 見積りにあたって

初めに、導入する全ての製品が必要とするIPCパラメーターを調べ、/etc/system に設定するパラメーターなのか /etc/project に設定するパラメーターなのかパラメーターの設定方式を調査します。更に/etc/project に設定する場合は、システムプロジェクトに設定するのか、ユーザ定義プロジェクトに設定するのかを調査します。

各製品が指定するパラメーターの設定方式	パラメーターを設定するファイル	概要
全て /etc/system に設定する製品である	/etc/system	・従来 (Solaris 9まで) と同様の算出方法
全て /etc/project に設定する製品である	/etc/project	・パラメーターの特性に従いパラメーター値を算出
/etc/system と /etc/project に設定する製品が混在している (※)	/etc/system と /etc/project	1. パラメーターの特性に従いパラメーター値を算出し、システムプロジェクトに設定 ↓ 2. システムプロジェクトと同等の値を/etc/systemに設定 ↓ 3. ユーザ定義プロジェクトを利用する製品の場合は、その製品のパラメーターのみを対象に算出し、ユーザ定義プロジェクトに設定

(※) 次ページから、/etc/system と /etc/project に設定する製品が混在しているときのIPC資源の見積り例を説明します。



## ■ 見積り条件

- ・製品A,B,C,D が必要とするIPC資源は下記の通り
- ・製品A,Bは/etc/system に設定する方式。製品C,D は/etc/project に設定する方式である。
- ・物理メモリサイズは32GB 搭載とする
- ・各パラメーターのデフォルト値は P.28 参照

	IPC資源		/etc/system	型	/etc/project
製品A	共有メモリ	総バイト数	shmsys:shminfo_shmmax = 100MB	-	—
		メモリIDの最大数	shmsys:shminfo_shmmni = 10	Σ	
製品B	共有メモリ	総バイト数	shmsys:shminfo_shmmax = 200MB	-	
		メモリIDの最大数	shmsys:shminfo_shmmni = 20	Σ	

	IPC資源		/etc/system	/etc/project	型
製品C	共有メモリ	総バイト数	—	project.max-shm-memory = 150MB	Σ
		メモリIDの最大数		project.max-shm-ids = 30	Σ
製品D	共有メモリ	総バイト数		project.max-shm-memory = 250MB	Σ
		メモリIDの最大数		project.max-shm-ids = 40	Σ

※ shmsys:shminfo\_shmmax は、project.max-shm-memory の値を算出するためのパラメーターなので、～型という特性はありません。

## ■ 共有メモリの見積り手順

1) 初めに、共有メモリ `project.max-shm-memory` のデフォルト値を確認します

デフォルト値は物理メモリの約1／4ですが、正確な値は`prctl $$コマンド`にて確認します。

物理メモリが32GBの場合は、確認すると**7.84GB** (※)となります。

(※)7.84GBは一例です。環境により値が前後する場合があります。

なお、コマンドで確認できない場合は、物理メモリの1／4で見積もってください。

2) 次に、`/etc/project` の設定がない製品 (A、B) の共有メモリ `project.max-shm-memory` を算出します

`/etc/system` の共有メモリのパラメーター (`shmsys:shminfo_shmmax`、`shmsys:shminfo_shmmni`) と共有メモリ `project.max-shm-memory` には以下の関係があります。(→ [P.24 IPC資源の見積り方法](#))

`shmsys:shminfo_shmmax` × `shmsys:shminfo_shmmni` = `project.max-shm-memory`

よって、`project.max-shm-memory` が未定義の製品A、B は以下のように算出します。

・製品A `shmsys:shminfo_shmmax` : 100MB  
`shmsys:shminfo_shmmni` : 10

100MB × 10 = 1GB (`project.max-shm-memory`)

・製品B `shmsys:shminfo_shmmax` : 200MB  
`shmsys:shminfo_shmmni` : 20

200MB × 20 = 4GB (`project.max-shm-memory`)

このように製品A、B が必要とする `project.max-shm-memory` は、1GB と 4GB と分かります。

3) 次に、製品A、B、C、D混在時の共有メモリ project.max-shm-memory を算出します

project.max-shm-memory はΣ型なので、

$$\begin{aligned}\text{project.max-shm-memory} &= 1\text{GB(A)} + 4\text{GB(B)} + 150\text{MB(C)} + 250\text{MB(D)} + 7.84\text{GB(デフォルト値)} \\ &= 13.24\text{GB} \\ &\doteq 14216341750 \text{ Byte}\end{aligned}$$

となります。

4) 最後に、共有メモリ shmsys:shminfo\_shmmax を算出します

/etc/system における shmsys:shminfo\_shmmax は、

$\text{shmsys:shminfo\_shmmax} = \text{project.max-shm-memory} / \text{shmsys:shminfo\_shmmni}$  によって計算します。

shmsys:shminfo\_shmmni はΣ型なので、

$$\begin{aligned}\text{shmsys:shminfo\_shmmni} &= 10\text{(A)} + 20\text{(B)} + 30\text{(C)} + 40\text{(D)} + 128\text{(デフォルト値)} \\ &= 228 (= \text{project.max-shm-ids})\end{aligned}$$

よって

$$\begin{aligned}\text{shmsys:shminfo\_shmmax} &= 13.24\text{GB} / 228 \\ &\doteq 62352376 \text{ Byte}\end{aligned}$$

となります。

## 5)プロジェクト別の共有メモリの算出

### 5-1)システムプロジェクト

システムプロジェクトへは、全製品(製品A、B、C、D)混在時の値を設定します。

$$\begin{aligned}\text{project.max-shm-memory}(\Sigma\text{型}) &= 13.24\text{GB} \\ &\doteq 14216341750\text{ Byte}\end{aligned}$$

$$\text{project.max-shm-ids}(\Sigma\text{型}) = 228 (= \text{shmsys:shminfo\_shmmni})$$

### 5-2)ユーザ定義プロジェクトの場合

ユーザ定義プロジェクトの場合は、設定は該当プロジェクト内の製品のみ対象として算出します。

製品Cの場合、

$$\begin{aligned}\text{project.max-shm-memory}(\Sigma\text{型}) &= 150\text{MB(C)} + 7.84\text{GB(デフォルト)} \\ &= 7.99\text{GB} \\ &\doteq 8579197174\text{ Byte}\end{aligned}$$

$$\begin{aligned}\text{project.max-shm-ids}(\Sigma\text{型}) &= 30(\text{C}) + 128(\text{デフォルト}) \\ &= 158\end{aligned}$$

製品Dの場合、

$$\begin{aligned}\text{project.max-shm-memroy}(\Sigma\text{型}) &= 250\text{MB(D)} + 7.84\text{GB(デフォルト)} \\ &= 8.09\text{GB} \\ &\doteq 8686571356\text{ Byte}\end{aligned}$$

$$\begin{aligned}\text{project.max-shm-ids}(\Sigma\text{型}) &= 40(\text{D}) + 128(\text{デフォルト}) \\ &= 168\end{aligned}$$

となります。

## ■ 共有メモリの見積り結果

/etc/system への設定値



shmsys:shminfo\_shmmax = 62352376  
shmsys:shminfo\_shmmni = 228

/etc/project への設定値

システムプロジェクト



project.max-shm-memory = 14216341750  
project.max-shm-ids = 228

ユーザ定義プロジェクト  
(各製品が使用するプロジェクト)



(製品C)  
project.max-shm-memory = 8579197174  
project.max-shm-ids = 158



(製品D)  
project.max-shm-memory = 8686571356  
project.max-shm-ids = 168

実際の設定は各ファイルの書式に従うこと

## ■ 見積り条件

- ・製品A,B,C,D が必要とするIPC資源は下記の通り
- ・製品A,Bは/etc/system に設定する方式。製品C,D は/etc/project に設定する方式である。
- ・各パラメーターのデフォルト値は P.28参照

	IPC資源		/etc/system	型	/etc/project
製品A	セマフォ	セマフォセットのセマフォ最大数	semsys:seminfo_semmsl = 100	Max	—
		システムの共用セマフォの最大数	semsys:seminfo_semmni = 64	Σ	
	メッセージキュー	メッセージ最大数	msgsys:msginfo_msgmni = 10	Σ	
製品B	セマフォ	セマフォセットのセマフォ最大数	semsys:seminfo_semmsl = 200	Max	
		システムの共用セマフォの最大数	semsys:seminfo_semmni = 128	Σ	
	メッセージキュー	メッセージ最大数	msgsys:msginfo_msgmni = 20	Σ	

	IPC資源		/etc/system	/etc/project	型
製品C	セマフォ	セマフォセットのセマフォ最大数	—	process.max-sem-nsems = 256	Max
		システムの共用セマフォの最大数		project.max-sem-ids = 200	Σ
	メッセージキュー	メッセージ最大数		project.max-msg-ids = 50	Σ
製品D	セマフォ	セマフォセットのセマフォ最大数		process.max-sem-nsems = 600	Max
		システムの共用セマフォの最大数		project.max-sem-ids = 300	Σ
	メッセージキュー	メッセージ最大数		project.max-msg-ids = 60	Σ

## ■ セマフォの見積り手順

1)/etc/system に設定する セマフォ semsys:seminfo\_semmsl、semsys:seminfo\_semmni 値を算出します。

1-1) semsys:seminfo\_semmsl はMax型のパラメーターなので、

$100(A) < 200(B) < 256(C) < 512(\text{デフォルト値}) < 600(D)$

より、最大値の600を設定する。 semsys:seminfo\_semmsl = 600

1-2) semsys:seminfo\_semmni はΣ型のパラメーターなので、

$64(A) + 128(B) + 200(C) + 300(D) = 692$

より、合計値の692を設定する。 semsys:seminfo\_semmni = 692

2)/etc/project に設定する セマフォ process.max-sem-nsems、project.max-sem-ids 値を算出します。

2-1) システムプロジェクトの場合、

/etc/system の設定値と同じく

process.max-sem-nsems = 600

project.max-sem-ids = 692

2-2) ユーザ定義プロジェクトの場合、設定は該当プロジェクト内の製品のみ対象に算出するので、製品Cの場合、

process.max-sem-nsems = 256

project.max-sem-ids = 200

を設定する。

製品Dの場合、

process.max-sem-nsems = 600

project.max-sem-ids = 300

を設定する。

process.max-sem-nsems の見積りに関しては、[7. 付録\(IPCパラメーター見積り時の留意事項\)](#) も参照してください。

## ■ メッセージキューの見積り手順

1) /etc/system に設定するメッセージキュー msgsys:msginfo\_msgmni 値を算出します  
Σ型のパラメーターなので、

$$\begin{aligned} & \text{msgsys:msginfo\_msgmni} \\ &= 10(A) + 20(B) + 50(C) + 60(D) + 128(\text{デフォルト値}) \\ &= 268 \end{aligned}$$

2) /etc/project に設定するメッセージキュー project.max-msg-ids 値を算出します

2-1) システムプロジェクトの場合、  
process.max-msg-ids もΣ型のパラメーターなので、/etc/system の設定値と同じく  
process.max-msg-ids = 268

2-2) ユーザ定義プロジェクトの場合、設定は該当プロジェクト内の製品のみ対象に算出するので、

製品Cの場合、  
 $50(C) + 128(\text{デフォルト値})$  より  
project.max-msg-ids = 178

製品Dの場合、  
 $60(D) + 128(\text{デフォルト値})$  より  
project.max-msg-ids = 188



## ■ セマフォ、メッセージキューの見積り結果

/etc/system への設定値



```
semsys:seminfo_semmsl = 600  
msgsys:msginfo_msgmni = 268
```

/etc/project への設定値

システムプロジェクト



```
process.max-sem-nsems = 600  
project.max-msg-ids = 268
```

ユーザ定義プロジェクト  
(各製品が使用するプロジェクト)



(製品C)  
process.max-sem-nsems = 256  
project.max-msg-ids = 178



(製品D)  
process.max-sem-nsems = 600  
project.max-msg-ids = 188

実際の設定は各ファイルの書式に従うこと

## 7. 付録（IPCパラメーター見積り時の留意事項）

ここでは、process.max-sem-nsems(セマフォセットあたりに許容されるセマフォの最大数)を見積もる際の留意事項を記載します。

セマフォはセマフォID単位にカーネル内でロックを取得するため、process.max-sem-nsemsが大きい場合は、同一のセマフォID内でセマフォの獲得が競合し、遅延が発生することがあります。

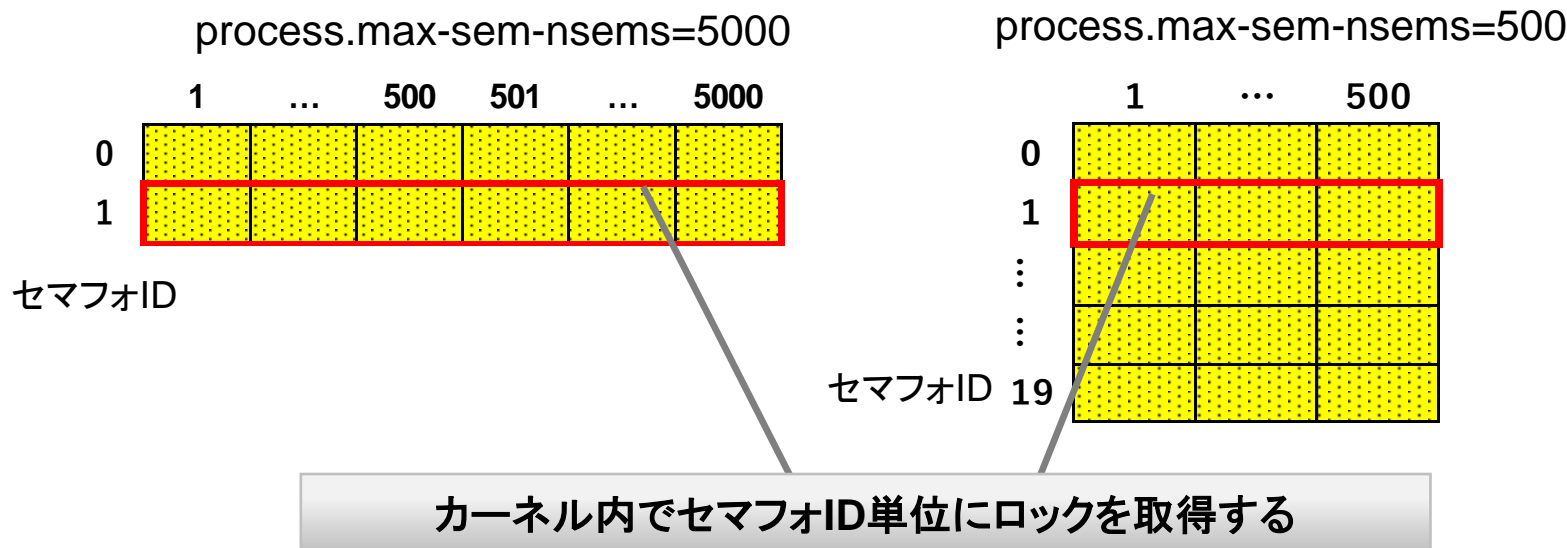
=>セマフォ獲得時の競合を考慮し、process.max-sem-nsemsを小さな値にすることが有効な対処になります。

“process.max-sem-nsems × project.max-sem-ids(セマフォIDの最大数) ≥ 必要とするセマフォ数” となるように設定してください。

例)

この例での前提

- ・ユーザアプリが必要とするセマフォ数: 10000
- ・ユーザアプリは、必要なセマフォ数をprocess.max-sem-nsemsで割って小数点以下を繰り上げた数分のセマフォID(最大: project.max-sem-ids)を使用します



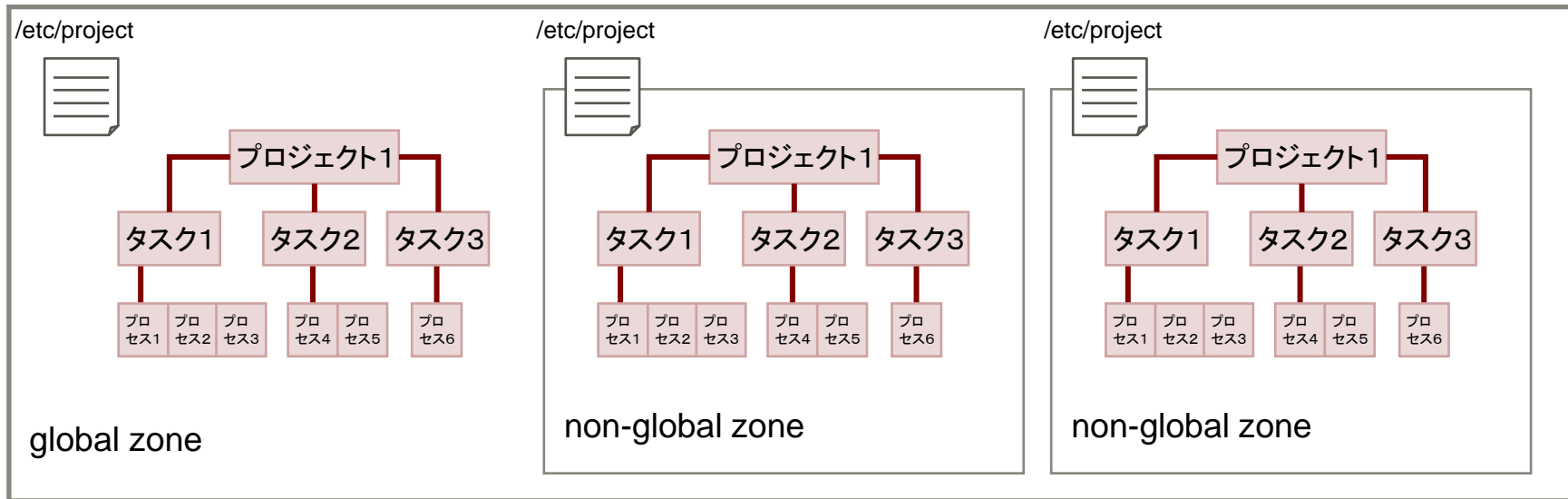
- ・ process.max-sem-nsemsを5000にした場合、セマフォIDは2個で、その下にセマフォが各々5000個存在するため、競合が発生しやすくなります。
- ・ process.max-sem-nsemsを500にした場合、セマフォIDは20個で、その下にセマフォが各々500個存在するため、競合が発生しにくくなります。

## 8. 付録（Oracle Solarisゾーン環境のIPCパラメーター）

## ■ Solaris ゾーン(non-global zone)単位に資源制御を実施します

Solaris ゾーン環境で実行されるプロセスはゾーン内の「プロジェクト」、「タスク」の下で制御されます。Solaris ゾーン環境のミドルウェアやアプリケーションの資源を制御する場合は、ゾーン単位に資源制御パラメーターを追加します。

### 《Solaris ゾーンを構成したときの構造》

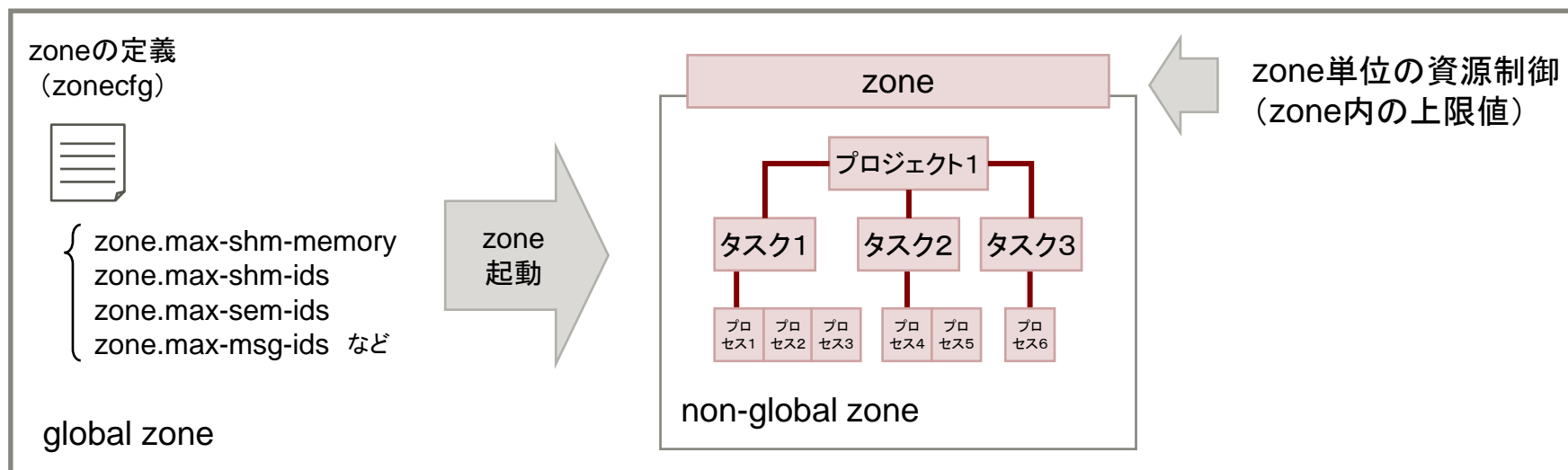


各zone(global zone、non-global zone)の資源制御は、そのzoneの/etc/projectファイルに定義します。  
/etc/projectファイルの定義は各zoneの管理者(rootユーザ)権限が必要です。

## ■ Solaris ゾーン(non-global zone)単位に使用資源の上限設定が可能です

Solaris ゾーン環境のプロセスが使用する資源の合計量を制限することができます。ゾーンで設定したプロジェクト単位(またはタスク単位、プロセス単位)の資源設定は、ゾーン単位に設定された上限値を超えて資源を使用することはできません。

### 《Solaris ゾーン用の資源制御》



- ・ゾーン単位の資源制御の設定はglobal zoneの管理者のみ実施可能です。zonecfgコマンドを使用してゾーンの構成定義の中で指定します。
- ・デフォルトではゾーン単位の制限は設定されていません。

# Oracle Solaris ゾーン環境のIPCパラメーターの設定 FUJITSU

## ■ ゾーン内のプロセスはゾーン内の/etc/project設定に従います

non-global zoneのプロセスはglobal zoneとは独立して資源管理されているため、non-global zoneはnon-global zoneの/etc/projectファイルを設定する必要があります。

global zone

/etc/project

プロジェクト1

プロセス  
150

プロセス  
150

例: プロジェクト1  
project.max-shm-ids=150

non-global zone

/etc/project

プロジェクト1

プロセス  
200

プロセス  
200

例: プロジェクト1  
project.max-shm-ids=200

カーネル

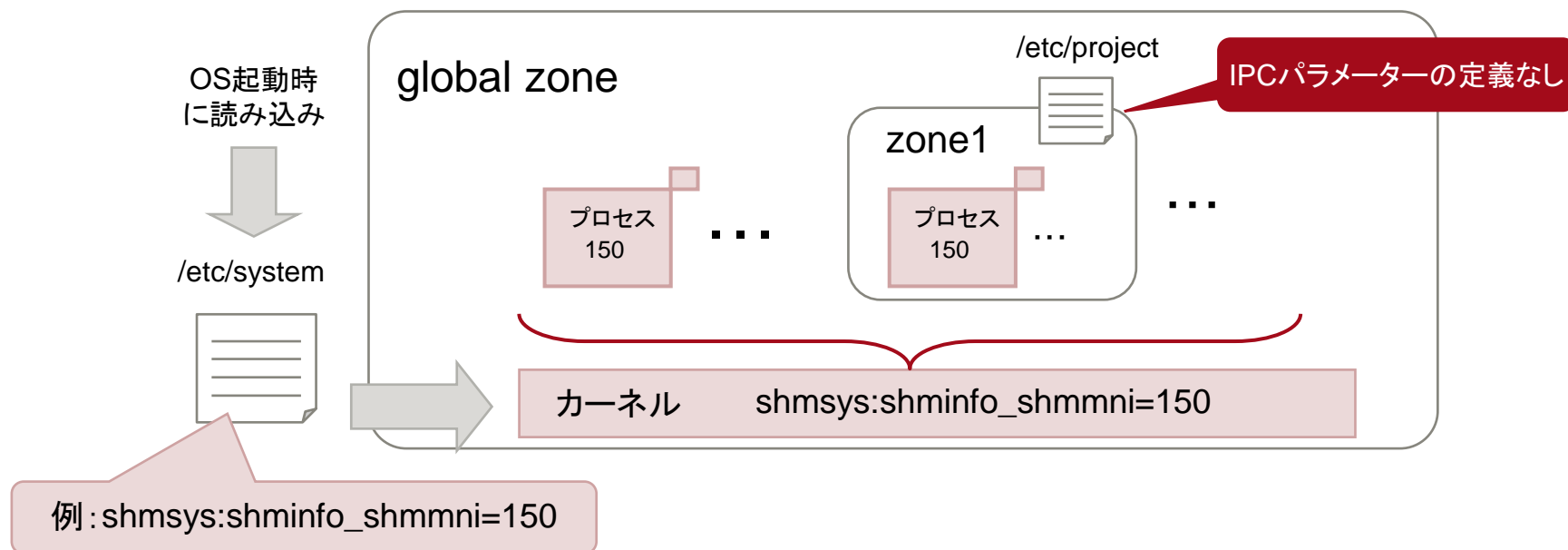
## non-global zoneのIPCパラメーターの見積り

ゾーン環境に導入するミドルウェアやアプリケーションが必要とするIPC資源を対象に、パラメーター値を見積ります。ゾーン環境では/etc/systemファイルは存在しないため、/etc/projectファイルのみに設定します。**/etc/systemの値は、デフォルト値として動作します(次ページ参照)**



## ■ /etc/systemのIPCパラメーター定義はゾーン環境にも引き継がれます

Solaris ゾーン環境では/etc/systemファイルは存在しないが、global zoneの/etc/systemファイルにIPCパラメーターを定義した場合はゾーン環境にも継承されます。そのため、global zoneの/etc/systemのみに定義されている(non-global zoneの/etc/projectファイルに定義されていない)パラメーターは/etc/systemの定義に従います。



non-global zoneの/etc/projectファイルに定義がある場合は、その設定値が優先されます。

改版日時	版数	改版箇所	改版内容
2012.8	1.0 版	—	新規作成
2014.1	1.1 版	P25,26 IPC資源の見積り方法 3/5,4/5 P45,47 見積り例 ～②セマフォ、メッセージキューの見積り～	ユーザプロジェクトの場合のMAX型のパラメタについて、各製品の指定する値の中で最大の値よりSolarisデフォルト値が大きい場合は、Solarisデフォルト値を採用する指針であったが、製品の指定する最大の値を採用する方針に変更
2016.3	1.2 版	P8,9 Oracle Solarisで制御可能な資源 パラメーター一覧	Solaris 11.3までの新規パラメータを反映
2019.4	1.3 版	P29 IPC資源パラメーター	表記見直し
2022.4	1.4 版	P9 Oracle Solarisで制御可能な資源パラメーター一覧	誤記修正(zone.max-shm-ids、zone.max-sem-ids、zone.max-msg-ids)
2023.10	1.5 版	7.付録(IPCパラメーター見積り時の留意事項)	新規追加
2024.1	1.6 版	P51 IPCパラメーター見積り時の留意事項	図の不要部分を削除 文章を全体的に見直し
2024.4	1.7 版	P23 IPC資源の見積り方法1/5 P45,46 見積り例 ～②セマフォ、メッセージキューの見積り～	セマフォの留意事項を追加 セマフォIDの最大数(ids)を追加、セマフォの留意事項の案内を追加

## 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

- 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。

