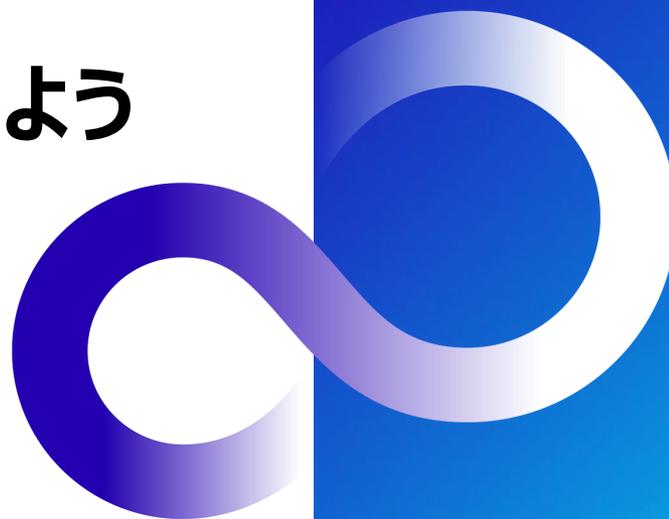


Oracle Solaris 10を使ってみよう ～構築・運用編～

2011年11月(第2版)

富士通株式会社



- 本書は、Oracle Solaris 10の基本操作やサービスの設定方法等について説明しています。
- 「Oracle Solaris 10 を使ってみよう(構築・運用手順書)」には、本書の説明に沿った具体的なインストール手順例を紹介しております。合わせてご参照ください。
 - Oracle Solaris 10 を使ってみよう(構築・運用手順書)
<https://www.fujitsu.com/jp/sparc-technical/document/solaris/os10.html#os>
- 本書は、Oracle Solaris 10 9/10で提供される機能をベースに作成しています。最新のOracle Solaris の情報については、マニュアルにてご確認ください。
 - Oracle Solaris 10 Documentation (Oracle社webサイトへリンク)
<http://www.oracle.com/technetwork/documentation/solaris-10-192992.html>
- 本書では Oracle Solaris をSolaris と記載することがあります。

1. サービスの起動・停止 (SMF)

2. 基本操作

3. Oracle Solaris 10の資源制御

4. 各種設定

5. システムの起動とシャットダウン

6. バックアップ／リストア

7. 情報収集

8. セキュリティ

1. サービスの起動・停止 (SMF)

SMFとは、Solaris 10から提供された新機能であり、従来のサービス起動スクリプト、init 実行レベル、および構成ファイルの管理を容易にするための基盤です。

ポイント

- サービス管理(起動、停止、再起動など)を、一貫したコマンドで操作可能です。【svcadm(1M)コマンド】
- サービス間の依存関係を考慮してサービスを起動します。【リポジトリDBによるサービス管理】
- 管理エラー、ソフトウェアのバグ、修復不可能なハードウェアエラーなどの原因によって停止したサービスを、自動的に再起動します。【予測的セルフヒーリング】
- 依存関係のない複数のサービスを並行起動し、システムの起動を迅速化します。

従来システム(～Solaris9)

`/etc/inittab`、`/etc/rc?.d`、`/etc/inet/inetd.conf`のような設定ファイル、実行制御スクリプトを用いて、ランレベルを基本としたサービスの管理

- `init`デーモンは、`inittab`ファイルから情報を読み取って、プロセスを起動。

`inittab`ファイルには、以下が格納。

- 起動・監視するプロセス、および停止時に再起動するプロセス
- システムが新しいランレベルに移行したときに行う処理

- `/etc/rc*.d`はランレベルを移行する場合、各ランレベルに対応するRCスクリプトを実行。実際のスクリプトは、`/sbin/rc*`に格納。

(`/etc/rc*.d`内のスクリプトは、スクリプト名をASCII文字列としてソートした順に実行。)

- 実行制御スクリプトは、`/etc/init.d`ディレクトリに追加しリンクを作成。

Solaris 10システム

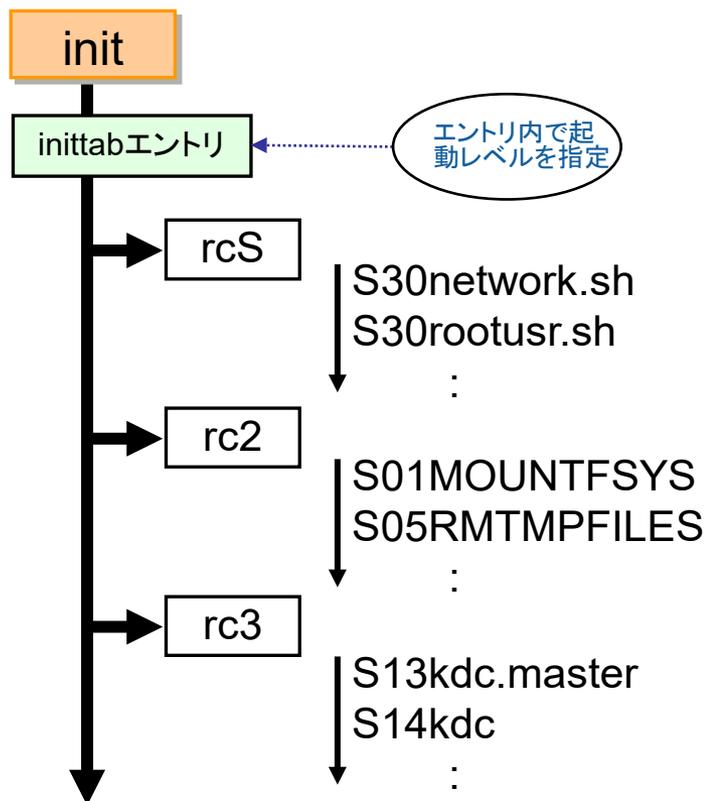
SMFによるサービスの一元管理

- ランレベル
変更なし。
7つ(0,1,2,3,5,6,S)のランレベル。
- `/etc/inittab`
互換性はあるが、SMFサービスに移行しているため、通常は、`inittab`にサービスを追加するのではなくSMFを使う。
- `/etc/rc*.d`のRCスクリプト
互換性あり。ただし、SMFサービスの起動後に実行されるRCスクリプトもあるため、他のSMFサービスとの依存関係には注意が必要。
- `inetd.conf`
互換性なし。`inetd.conf`の追加された内容は、後述する`inetconv(1M)`で、SMFに変換が必要。

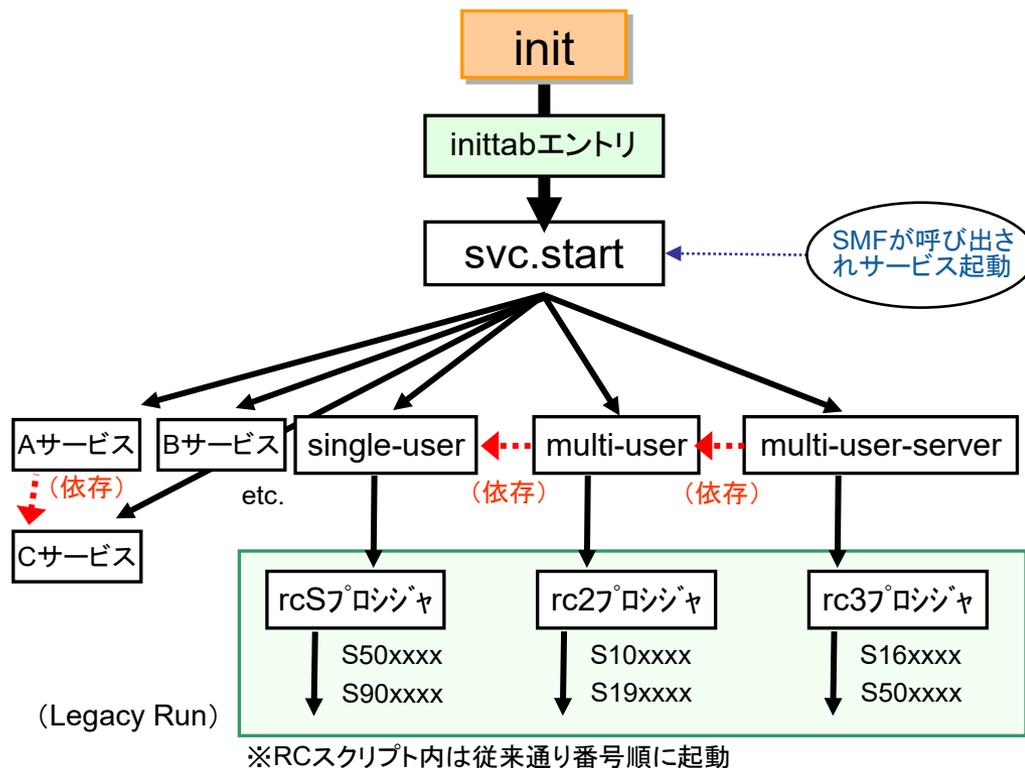
※従来のサービス管理(RCスクリプト)で起動したサービスは、SMFでサービス状態の確認はできるが、起動/停止などの操作はできません。

■ RCスクリプトとSMFの比較

《Solaris 9以前(RCスクリプト)》



《Solaris 10以降(SMF)》



- ・SMF登録サービスを同時に同時起動。
- ・サービス間の依存関係がある場合は順番に従い起動。

■ SMFが提供するサービス管理コマンド一覧

コマンド	機能
svcs	サービスのステータスをレポート
svcadm	サービスインスタンスの操作(従来の psコマンドまたは、killコマンドに相当)
svccfg	サービスのインポート、エクスポート、変更(従来の rcスクリプトの編集作業に相当)
svccprop	サービスのプロパティを回復
inetadm	inetd経由のサービス管理
inetconv	inetd.conf形式のサービスをSMF管理下にコンバートする



この2つは
良く使います

■ サービス表示名

FMRI (Fault Management Resource Identifier: 障害管理リソース識別子)

svc:/network/http:apache2

サービス名 インスタンス名

※lrcで始まるFMRI名はRCスクリプトから実行されるサービスであることを示します。

例: lrc:/etc/rc3_d/S90samba

■ サービスの状態確認の方法 [svcs コマンド]

```
# svcs -a

STATE      STIME      FMRI
legacy_run 20:50:33   lrc:/etc/rcS_d/S50FJSVmadmclr
legacy_run 20:50:37   lrc:/etc/rcS_d/S50fjgi_sh
legacy_run 20:50:37   lrc:/etc/rcS_d/S90FJSVscfdaemon
--(省略)--
disabled   20:50:23   svc:/network/routing/ripng:default
disabled   20:50:23   svc:/network/ipv6-forwarding:default
disabled   20:51:14   svc:/network/login:klogin
--(省略)--
online     20:52:08   svc:/system/basicreg:default
online     20:52:08   svc:/system/zones:default
online*    20:53:11   svc:/system/webconsole:console
offline    20:50:22   svc:/application/print/ipp-listener:default
offline    20:51:10   svc:/application/print/rfc1179:default
:
```

* が付いているサービスは、現在実行中の意味

ステータス	意味
degraded	サービスは有効だが制限された状態で動作している状態
disabled	サービスは無効で起動していない状態
legacy_run	SMFではなく、RCスクリプトで管理されているサービスを示す
maintenane	管理者が対応すべきメンテナンスモード
offline	サービスが有効だが起動していない状態
online	サービスが有効で無事起動している状態
uninitialized	設定ファイルが読み込まれる前の初期状態

■ SVCSコマンドオプション

実行形式	意味
svcs	ステータスがdisabled以外の有効なサービス表示 “-a”で全サービスリスト
svcs -d [FMRI]	自身が依存しているサービスを表示
svcs -D [FMRI]	自身に依存しているサービスを表示
svcs -l [FMRI]	詳細情報を表示
svcs -x [FMRI]	各サービスの説明（デバック時に多用）
svcs -p [FMRI]	サービスのプロセス情報を含めて表示
svcs -v	冗長化表示。-xが追加されている場合は、各説明に関する追加情報を表示。

■ サービスの起動・停止の方法 [svcadm コマンド]（従来のps、killコマンド相当）

サブコマンド	意味		実行形式
enable	サービスの起動	恒久的に起動	svcadm enable [FMRI]
		一時的に起動	svcadm enable -t [FMRI]
		依存サービス含め起動	svcadm enable -r [FMRI]
disable	サービスの停止	恒久的に停止	svcadm disable [FMRI]
		一時的に停止	svcadm disable -t [FMRI]
restart	特定サービスの再起動		svcadm restart [FMRI]
refresh	サービス構成の再読み込み		svcadm refresh [FMRI]
mark	maintenance状態の設定		svcadm mark [FMRI]
clear	maintenance状態の解除		svcadm clear [FMRI]
milestone ※	マイルストーン の設定	ランレベル 5	svcadm milestone milestone/single-user
		ランレベル 2	svcadm milestone milestone/multi-user
		ランレベル 3	svcadm milestone milestone/multi-user-server

※マイルストーンサービス: システムのランレベルに対応するサービス

これらのサービスは、サービスの依存関係の記述によって、事前に起動すべきサービスを規定します。それらのサービスが起動した後、対応するランレベルのinitスクリプトを実行することになります。

- サービスの起動

```
# svcadm enable svc:/network/telnet:default
```

もしくは

```
# svcadm enable telnet
```

FMRIは一意であれば省略可能

- サービスの停止

```
# svcadm disable telnet
```

- サービスの再起動

```
# svcadm restart telnet
```

- サービスの起動確認

```
# svcs -a | grep telnet
```

- サービスの詳細表示

```
# svcs -l telnet
```

- メンテナンス状態時の詳細確認

```
# svcs -xv telnet
```

詳細は、設計・運用手順書
「1. サービスの起動・停止 (SMF)」
を参照してください。

2. 基本操作

■ Solaris OSへのログイン

```
login: user01
Password:
Last login: Tue Jun 23 17:05:47 from 10.75.114.212
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
$
```

login: 一般ユーザ
Password: 一般ユーザのパスワード

■ 管理者アカウントへの変更 (suコマンド)

```
$ su -
Password:
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
#
# id
uid=0(root) gid=0(root)
#
```

一般的な管理者アカウントは
“root”ユーザ

Password: rootユーザのパスワード

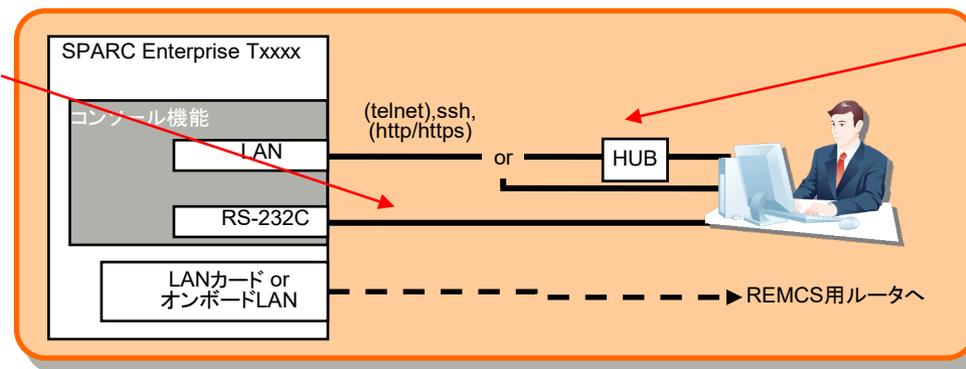
管理者アカウント (rootユーザ) での直接ログインはセキュリティ上推奨しません。通常は一般ユーザでログインしてから、su コマンドでrootユーザに変更します。
デフォルトの設定では、rootユーザの直接ログインは制限されています。

■ システム監視や保守作業のためにコンソールPCが必要です

■ SPARC Enterprise サーバのコンソール接続について

SPARC Enterpriseモデル	コンソール機能	接続プロトコル	備考
SPARC Enterprise M3000/M4000/M5000/M8000/M9000	XSCF (RS-232C、LAN)	telnet/ssh http/https	・https設定は必須
SPARC T3-1/T3-2/T3-4, SPARC Enterprise T5120/T5220/T5140/T5240/T5440	ILOM (RS-232C、LAN)	ssh http/https	・LAN経由のtelnet接続は不可 ・グラフィックカード接続も可能

RS-232C接続は初期
セットアップ時やCE作
業時に使用します。

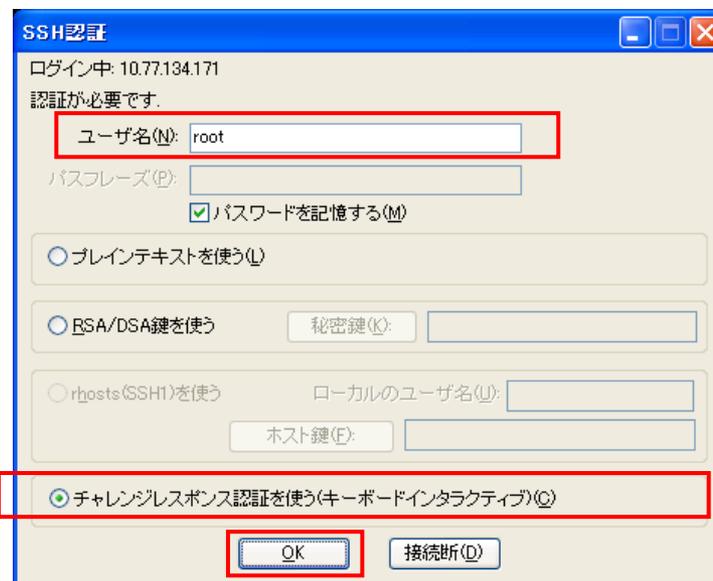


通常運用時はLAN経由で
コンソール接続を行います。
(HUB経由または直結接続)

- SPARC T3-1/T3-2/T3-4, SPARC Enterprise T5120/T5220/T5140/T5240/T5440のシステムコントローラ(ILOM)は、sshのみサポートしています (telnetでの接続不可)

接続方法

- ① ssh対応のターミナルソフトを起動 (例: UTF-8 Tera Term Pro 4.57)
- ② 「ホスト」にILOMのIPアドレスを入力
- ③ 「サービス」で "SSH"を選択 (SSHバージョンはSSH2のまま)
- ④ OKをクリック
- ⑤ 「ユーザ名」に"root"を入力
- ⑥ 「チャレンジレスポンス認証をつかう...」を選択
- ⑦ OKをクリック



- ⑧ パスワードを入力(※)
(デフォルトのrootパスワードは"changeme"です)

✓ターミナルソフトによって接続方法が異なります。

■ viの必要性

UNIXシステムの各種設定ファイルの表示・編集に必要

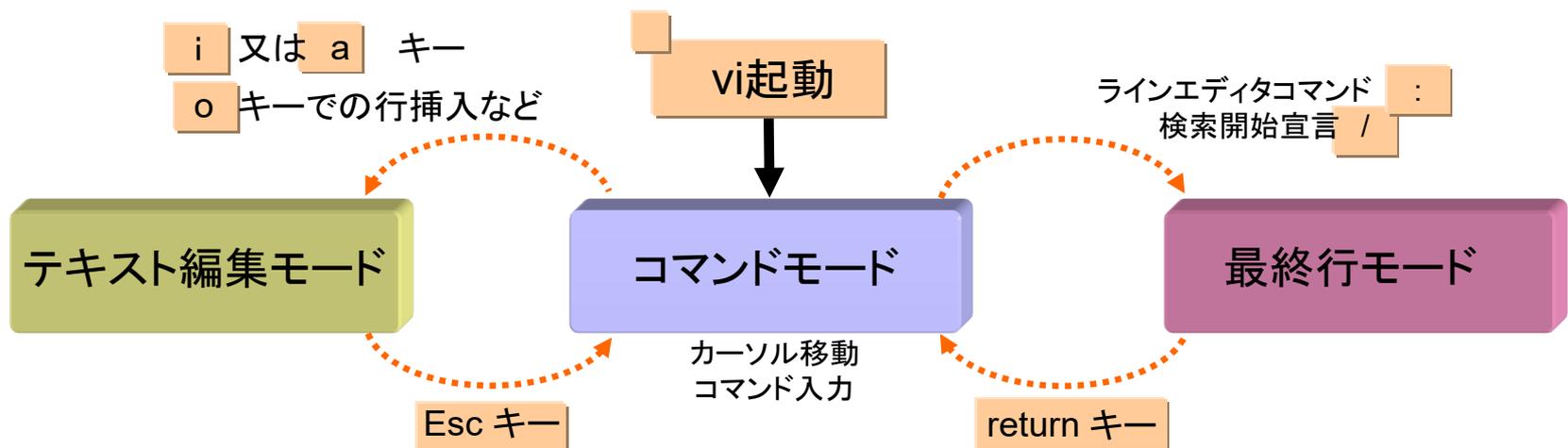
最小限のキー操作方法と保存方法の習得は必須！



■ 起動方法と3つのモード

● 起動方法 `$ vi [オプション] [ファイル名]`

● 3つのモード



● コマンドモードでの主なキー操作

キー操作	例	事例コメント	
カーソルを左に移動する	h		
カーソルを下に移動する	j		
カーソルを上を移動する	k		
カーソルを右に移動する	l(エル)		
カーソルの前に文字を挿入	i(アイ)		
カーソルの後に文字を追加	a		
カーソル位置の1文字を削除	x	3x	3文字削除
カーソル位置の1行を削除	dd		
カーソル位置の行から n行を削除 (nは数字)	ndd	3dd	カーソル位置の 行から3行削除
文字の下方方向検索	/検索文字列	/error	error文字を検索
行の追加 (カーソル位置の下)	o		
行の追加 (カーソル位置の上)	O		
1文字置換	r文字		
カレント行をバッファに入力 (コピー)	yy		
n行をバッファに入力(コピー)	nyy	3yy	3行コピー
バッファに格納されている 文字の挿入	p		

● 最終行モードでの主なキー操作

キー操作	例	事例コメント	
保存(1)	:w		
保存(2)(指定された ファイルに保存)	:w ファイル名	:w test	ファイル名“test” として保存
保存終了(1)	ZZ		
保存終了(2)	:wq		
強制保存	:w!		
強制保存,強制終了	:wq!		
保存しないで終了	:q!		
コマンドの取り消し	u		
他のファイルの編集	:e ファイル名	:e test	testファイル編集
行番号の表示	:set number		
モード状態の表示	:set showmode		
自動インデント	:set autoindent		
OSコマンド実行	!	! ls	編集中にls コマンドを実行

- 更新不可でファイルをオープン(viコマンドと同じ操作可)
view ファイル

- viコマンド実行時、以下のメッセージが出た場合

```
XXXXXXXX: 端末形式が不明です。  
ビジュアル (visual) モードにはカーソル位置指定、逆改行機能が必要です。  
:
```

```
':q'でコマンド終了  
# TERM=vt100 ; export TERM (Bourneシェルの場合)  
% setenv TERM vt100 (Cシェルの場合)
```

- viコマンドで編集のモードを画面に表示したい場合
: set showmode (コマンドモードで実行、画面右下にモード表示)

- viコマンドで編集中に画面が乱れた場合
[Esc] + Ctrl+L

- ~/.exrcファイルで vi 環境定義が可能
: set all (設定可能な環境を表示)

.exrcファイル(例)

```
set showmode  
set number  
set autoindent
```

■ rootユーザのtelnetログインとftp接続はデフォルトで無効化されています

● telnetを制限している設定 [/etc/default/login ファイル]

```
# If CONSOLE is set, root can only login on that device.  
# Comment this line out to allow remote login by root.  
#  
CONSOLE=/dev/console ← この行をコメントにするとrootユーザの直接ログインが可能
```

● ftpを制限している設定 [/etc/ftpd/ftpusers ファイル]

```
# ident "@(#)ftpusers 1.4 01/08/27 SMI"  
#  
# List of users denied access to the FTP server, see ftpusers(4).  
#  
root ← この行をコメントにするとrootユーザ権限でftp接続が可能  
daemon  
bin  
:
```

セキュリティ上、管理者権限での telnet や ftp は推奨されません。
特別な要件がない限り、デフォルトのまま無効化しておくこと。

■ 作成したユーザに関する情報は各ファイルにて管理されます

● グループの新規作成 [groupadd コマンド]

```
# groupadd -g 100 solaris ← 100~60000 のGIDを指定
#
```

実行後の管理ファイル



/etc/group

● ユーザの新規作成 [useradd コマンド]

```
# useradd -u 100 -g solaris -d /export/home/user01 -s /bin/sh -m user01
64 blocks
```



/etc/passwd

● パスワードの設定 [passwd コマンド]

```
# passwd user01 ← パスワード設定ユーザを指定
新しいパスワード: ***** ← 6~8文字の英数字、特殊文字の組み合わせ
新しいパスワードを再入力してください: *****
passwd: user01 のパスワードが変更されました
```



/etc/shadow

各ファイルを直接編集しても設定変更が可能ですが、書式を誤るとログインできなくなるため推奨しません。設定変更時はコマンドを実行して実施してください。

<参考>各ユーザ定義ファイルについて

●/etc/group ファイル

```
<group name>:<group password>:<gid>:<user-list>
```

<group name>	グループに付けた名前
<group password>	オプションフィールド
<gid>	グループID
<user-list>	グループに属するユーザ

●/etc/passwd ファイル

```
<user name>:<password>:<uid>:<gid>:<comment>:<home directory>:<login shell>
```

<username>	ユーザ名
<password>	パスワード(実体は/etc/shadow ファイルに存在します)
<uid>	ユーザID
<gid>	グループID
<comment>	任意の情報
<home directory>	ホームディレクトリ
<login shell>	ログインシェル

●/etc/shadow ファイル

```
<user name>:<password>:<lastchg>:<min>:<max>:<warn>:<inactive>:<expire>:<flag>
```

<username>	ユーザ名
<password>	暗号化されたパスワード
<lastchg>	最後にパスワードが変更された日
<min>	再度パスワードを変更する際に開けるべき最低日数
<max>	ユーザにパスワードを変更を強制させるまでの最大日数
<wam>	パスワードの失効前にユーザに警告を出す日数
<inactive>	ユーザのアカウントがロックされずに済む、最大未使用日数
<expire>	ユーザアカウントの失効日
<flag>	失敗したログイン回数を下位4ビットで保持

オプションを指定して、ユーザに詳細な制限を設定することが可能です。

■ ユーザは複数の環境設定ファイルを読み込むことで、独自のシェル環境を構成できます

- システムプロファイル [/etc/profile]

Bourneシェル(sh)、Kornシェル(ksh)環境において一番初めに読み込まれるファイルです。

- 初期設定ファイル

ユーザアカウントのホームディレクトリには、いくつかの初期設定ファイルが存在し、シェル環境によって使用するファイルが決まっています。

ファイル	sh	csh	ksh	説明
初期設定ファイル	.profile	.login	.profile	ユーザがログインした時に実行される
シェル起動ファイル	—	.cshrc	ユーザ定義 (一般的に.kshrc)	ログインユーザがシェルを起動すると実行される
ログアウトファイル	—	.logout	—	ユーザがログアウトした時に実行される

- 初期設定ファイルのテンプレート

初期設定ファイル(テンプレート)は、/etc/skel ディレクトリにあります。

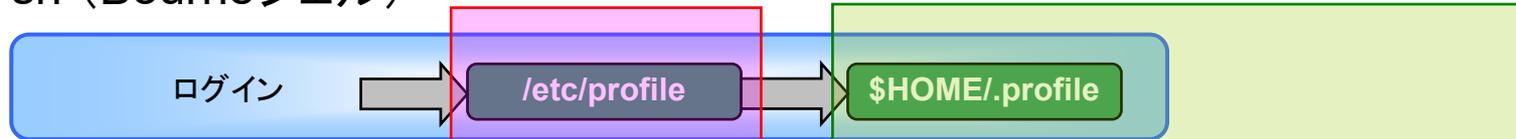
useraddコマンドでのユーザ作成時に、全ての初期設定ファイルがホームディレクトリにコピーされます

ファイル	sh	csh	ksh
テンプレート	/etc/skel/local.profile /etc/skel/.profile	/etc/skel/local.login /etc/skel/local.cshrc	/etc/skel/local.profile /etc/skel/.profile

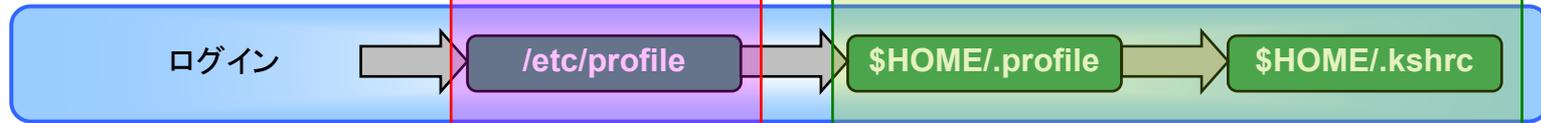
環境設定ファイルの実行

- シェルによって読み込む環境設定ファイルが異なるため、全ユーザ共通の設定やユーザ個別の設定を行うことが可能です。

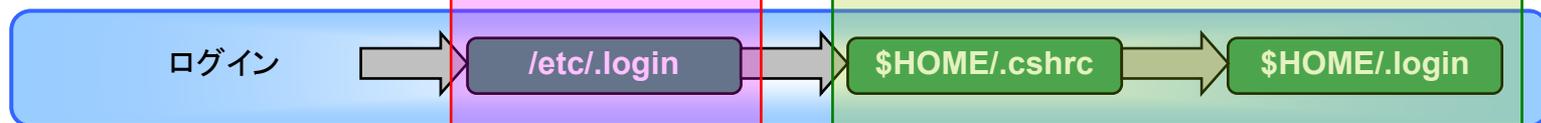
- sh (Bourneシェル)



- ksh (Kornシェル)



- csh (Cシェル)



↑
管理者 (rootユーザ) が
共通設定として定義

↑
一般ユーザが独自に
設定可能

■ 環境変数の定義方法はシェル環境によって構文が異なります

(例)ターミナルエミュレータの種類を指定する変数(TERM)を定義する

shの場合

```
TERM=vt100
export TERM
    もしくは、
TERM=vt100 ; export TERM
```

kshの場合

```
export TERM=vt100
```

cshの場合

```
set TERM=vt100
    もしくは、
setenv TERM vt100
```

ユーザ変数の定義(例) .profileに記述する

```
stty erase ^H
PS1=""hostname`# ` ; export PS1
PATH=$PATH:/usr/sbin:/usr/local/sbin:/usr/ucb:.
export PATH
TERM=vt100 ; export TERM
EDITOR=vi ; export EDITOR
LANG=ja ; export LANG
```

記述後は再読み込みを実行

```
sh, ksh の場合(ドット実行)
# . .profile
cshの場合
# source .cshrc
```

※再ログインでも同様

■ ユーザ環境変数を変更する例

1. ホームディレクトリに移動する。

```
$ cd
```

2. viエディタで.profileを開く

```
$ vi .profile
```

3. 以下の通りファイルを編集し、保存する。

```
stty erase "^H"  
PS1="`hostname`# ` "; export PS1  
PATH=$PATH:/usr/sbin:/usr/local/sbin:/usr/ucb:.  
export PATH  
TERM=vt100 ; export TERM  
EDITOR=vi ; export EDITOR  
LANG=ja ; export LANG
```

4. 環境変数の変更を反映させる

```
$ . .profile
```

※一般ユーザである場合の例

コマンド間は「`」(アクサン記号)で
囲む ([Shift]+[@キー])で入力)

「.」(ドット)を付ける場合は
最後のパスに付ける

〔 セキュリティ上「.」を
PATHに記載するのは
好ましくありません。 〕

詳細は、設計・運用手順書
「2. 基本操作」
を参照してください。

3. Oracle Solaris 10の資源制御

■ カーネルパラメータとは

OSの基本機能を提供するソフトウェアであるカーネルをチューニングするためのパラメータです。Solaris OSでは、`/etc/system` ファイルに記述し、カーネルパラメータを調整します。このファイルに指定された値は、ブート時に読み込まれ適用されるため、カーネルパラメータの変更は、システムがリブートされるまでOSには適用されません。

■ 資源制御パラメータ

カーネルパラメータの中には資源制御に関するパラメータが含まれています。Solaris9までは全て`/etc/system`で管理していましたが、Solaris 10からは`/etc/project`の資源制御パラメータによって管理できるようになりました。(`/etc/system`も互換あり)

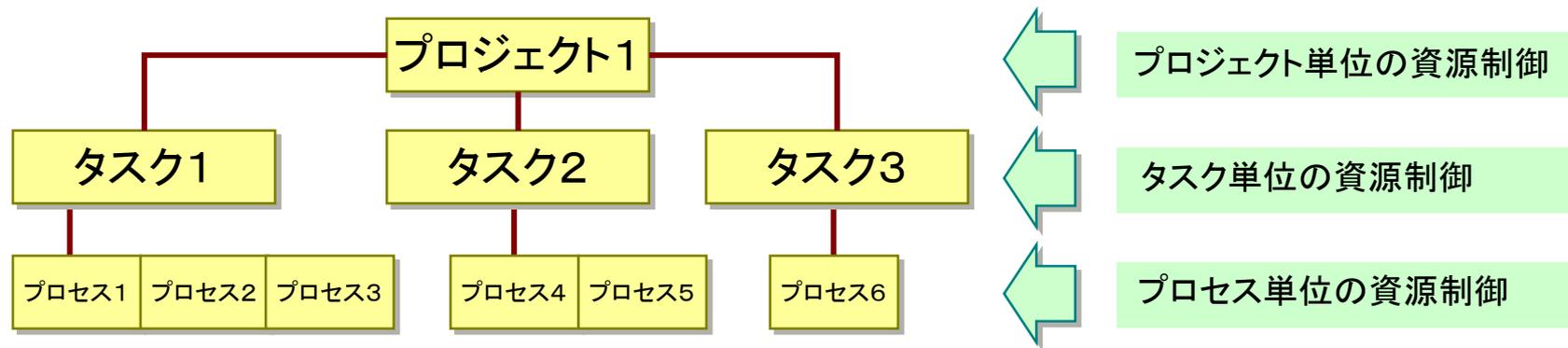
OS	カーネルパラメータ	
		資源制御(IPC)
~Solaris 9	<code>/etc/system</code>	<code>/etc/system</code>
Solaris 10	<code>/etc/system</code>	<code>/etc/project</code> (<code>/etc/system</code>)

Solaris 10からの新しい方式

■ Solaris 10の資源制御はプロジェクト、タスクという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ます。起動時のデーモンやユーザのプロセスも必ずある「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》

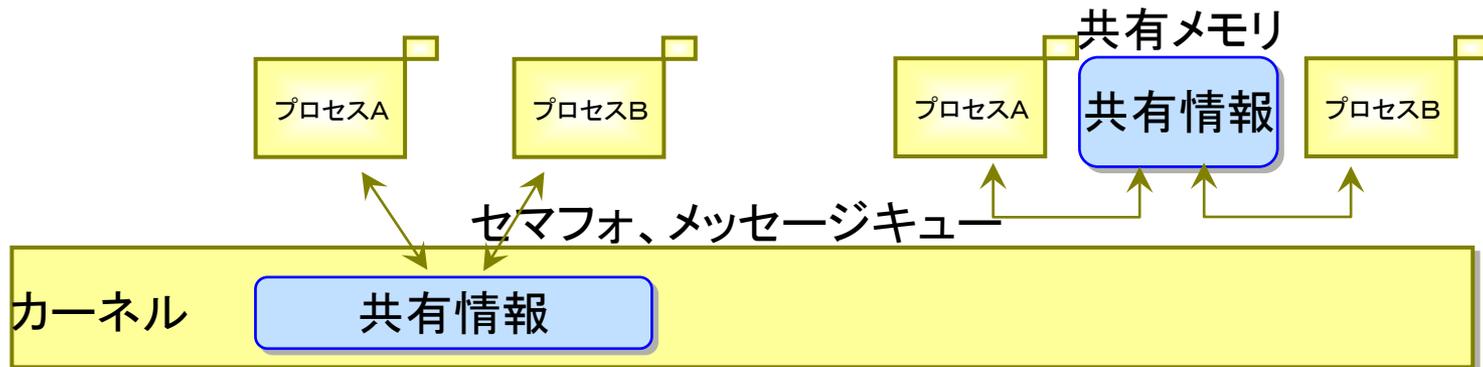


- ・資源制御は/etc/project ファイルへ「資源制御パラメータ」を追加することで実施します
- ・資源制御パラメータには「IPC資源」に関するパラメータが含まれています

■ 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種類の機能があります。

《プロセス間の情報共有》



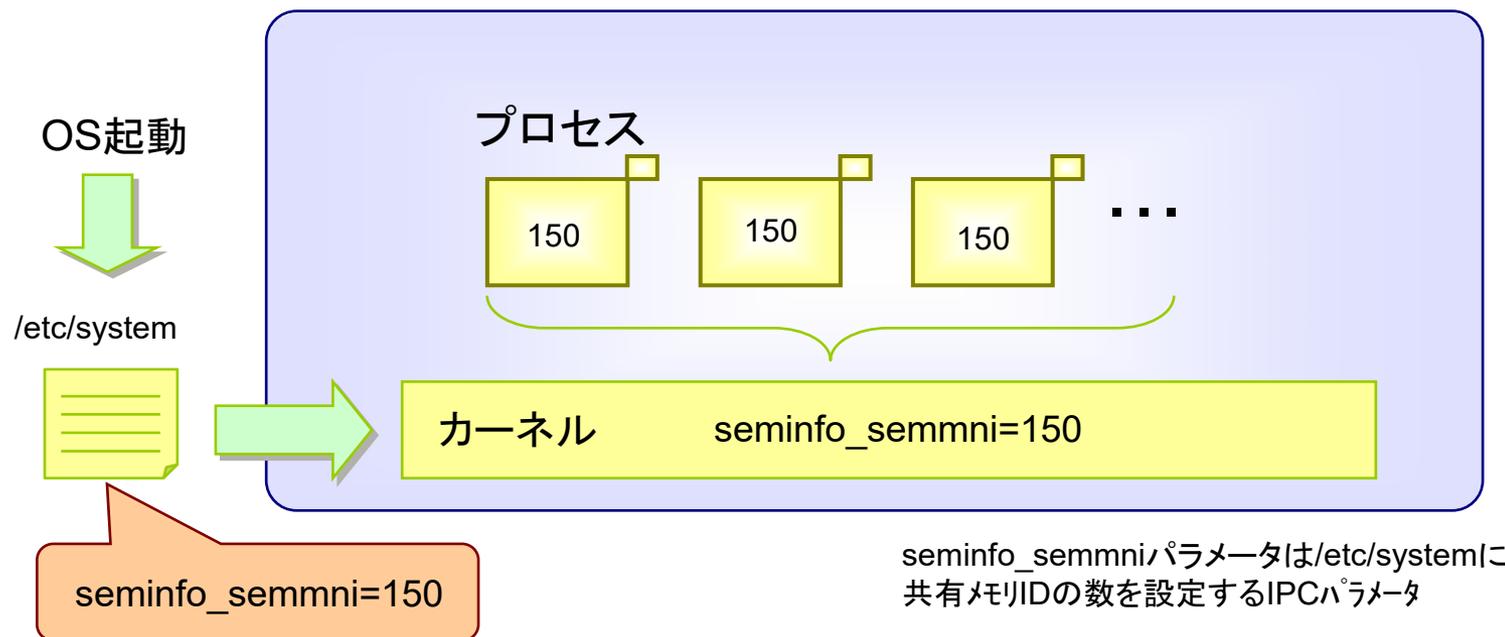
設定の必要性

Solaris OSではデフォルトでIPC資源の値が定義されているが、導入するミドル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おり、OSデフォルト値では不足する場合があります。そこで、事前にミドルやアプリが必要とするIPC資源の設定値を見積もり、OSに設定する必要があります。

IPCパラメータの動作（1）

■ /etc/system に設定した場合

OS起動時にカーネルパラメータに設定され、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。



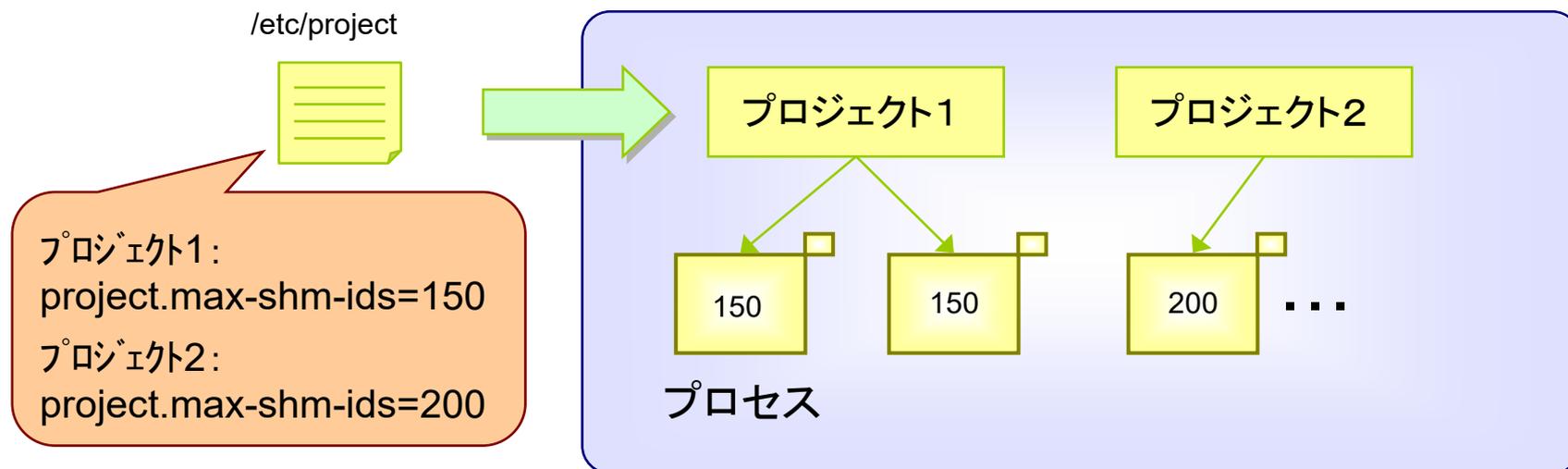
/etc/system による設定の特長

パラメータ変更時にはOSの再起動が必要となります。

IPCパラメータの動作（2）

■ /etc/project に設定した場合

プロジェクト毎にパラメータ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。



project.max-shm-memoryパラメータは/etc/projectに共有メモリIDの数を設定するIPCパラメータ

/etc/project による設定の特長

IPCパラメータ値の設定変更後、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で設定します。

■ /etc/system と /etc/project 混在時

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

- ➡ パラメータ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 の設定あり

例 user.rootプロジェクト:

project.max-shm-ids=150

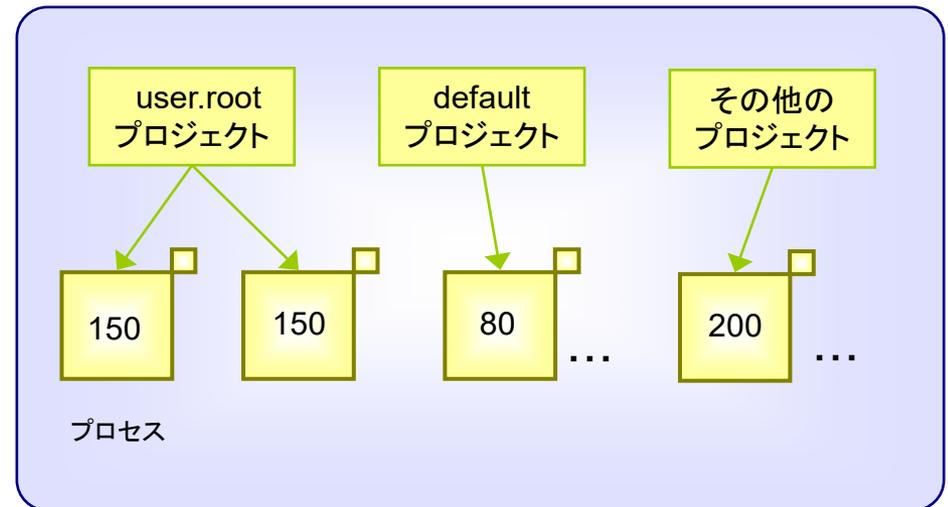
defaultプロジェクト:

project.max-shm-ids=80

/etc/system の設定あり

例 カーネルパラメータ

seminfo_semmni=200



(*)project.max-shm-ids と seminfo_semmni はセマフォID数を設定するIPCパラメータ。Solarisデフォルト値は128。

■ IPCパラメータの設定に必要なコマンド

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。

<参考> Oracle 11g 導入時のパラメータ設定

■ Oracle環境構築時は/etc/projectファイルにIPCパラメータを設定します。

- Oracleユーザの所属するプロジェクトに、共有メモリの最大値を設定します。

```
# projadd -K 'project.max-shm-memory=(privileged,4294967296,deny)' user.oracle
```

/etc/projectファイルに設定値が反映されます。

```
system:0:::
user.root:1:::
noproject:2:::
default:3:::
group.staff:10:::
user.oracle:100:::project.max-shm-memory=(privileged,4294967296,deny)
```

/etc/projectファイルを直接編集することは推奨しません。

- Oracleユーザにて設定値を確認します。

```
$ prctl -n project.max-shm-memory $$
process: 7647: bash
NAME PRIVILEGE VALUE FLAG ACTION
RECIPIENT
project.max-shm-memory
privileged 4096MB - deny
-
system 16.0EB max deny
```

詳細は、設計・運用手順書
「3. Oracle Solaris 10の資源制御」
を参照してください。

4. 各種設定

cronの設定（ジョブスケジューラ）

- cronを設定することで、指定した時間にプログラムを実行させることが可能です。

- cronのディレクトリ構成

/var/spool/cron/crontabsディレクトリ

ファイル名

adm

lp

root

sys

uucp

一般ユーザ

- cron設定の流れ

```
# crontab -l (crontabファイルの確認)
#ident "@(#)root 1.20 01/11/06 SMI"
#
10 3 * * 0,3 /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rpc ] && /usr/sbin/rpc -c > /dev/null 2>&1
30 3 * * * [
#10 3 * * *
# Copyright
# @(#)root
#
0 1 * * * /op
# Copyright
# @(#)root
30 3 * * 0 if
px; fi #FJSVael
#12 9 * * * sh export/home/mino/copy/copy.sh > /dev/null 2>&1

# crontab -e (crontabファイルの編集)
```

10 3 * * 0,3 /usr/sbin/logadm

分 時 日 月 週

実行プログラム

日:0 月:1 火:2 水:3 木:4 金:5 土:6

EDITOR=vi ; export EDITOR
を実行しておく。(もしくは .profile
に記述しておくこと。)

■ cron設定コマンド

- ① cronサービスが起動していることを確認します。

```
$ svcs -a | grep cron  
online      6月_01  svc:/system/cron:default
```

- ② -e オプションを指定したときに呼び出すエディタを指定します。

```
$ EDITOR=vi ; export EDITOR  
$ echo $EDITOR  
vi
```

EDITOR変数を指定していないと編集できません

- ③ crontabを編集します。

```
$ crontab -e  
* * * * * df -h > /df_file
```

※ ファイルを保存すると、/var/spool/cron/crontab ディレクトリ配下に作成ユーザ名のファイルが作成(更新)されます

- ⑤ crontabの編集内容を確認します。

```
$ crontab -l
```

- ⑥ 指定したファイルを参照します。

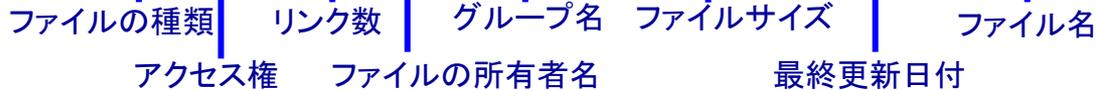
```
$ cat /df_file
```

ファイルのアクセス権 (パーミッション)

- 全てのファイルにはアクセス権が設定され保護されています。
- ファイルには3つのユーザ区分があり、各区分に読み込み/書き込み/実行の権限を設定することが可能です。

● アクセス権の確認方法 (=ファイル詳細表示: [ls -l])

```
# ls -l
-rw-r--r--  1 root  other    0  9月 13日 17:14 a_file
-rw-rw-r--  1 root  other    0  9月 13日 17:14 b_file
-rwxrwxrwx  1 root  other    0  9月 13日 17:14 c_file
drwxr-xr-x  2 root  other  512  9月 13日 17:14 d_file
```



● アクセス権の見方



ファイルの種類	権利の種類	
d	r (Read)	:読み込み
-	w (Write)	:書き込み、削除
l	x (eXecute)	:実行
b	s (set uid,set gid)	:所有者権限、グループ権限での実行
c	t (sticky bit)	:作成者のみ変更・削除

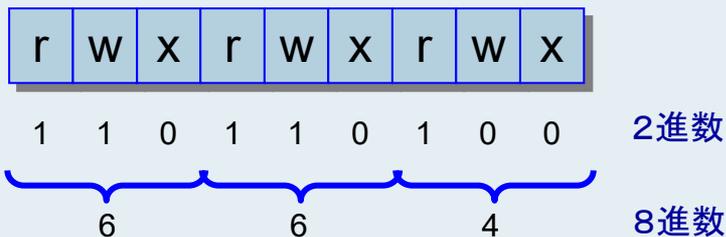
ファイルのアクセス権の変更

■ アクセス権の変更

chmod <設定対象><操作><アクセス権> ファイル名
 chmod <8進数> ファイル

```
# ls -l
-rw-r----- 1 root other 0 9月 13日 17:14 a_file
# chmod o+r a_file
# ls -l
-rw-r--r-- 1 root other 0 9月 13日 17:14 a_file
# chmod 664 a_file
# ls -l
-rw-rw-r-- 1 root other 0 9月 13日 17:14 a_file
```

8進数によるアクセス権を指定



《設定対象》

u (user) :ファイルの所有者
 g (group) :同グループのユーザ
 o (other) :他グループのユーザ
 a (all) :全てのユーザ

《操作》

+ :追加
 - :拒否
 = :許可

《アクセス権》

r (Read) :読み込み
 w (Write) :書き込み、削除
 x (eXecute) 実行

s (set uid , set gid)
 t (sticky bit)

<参考>パイプ（標準出力/入力の連結）

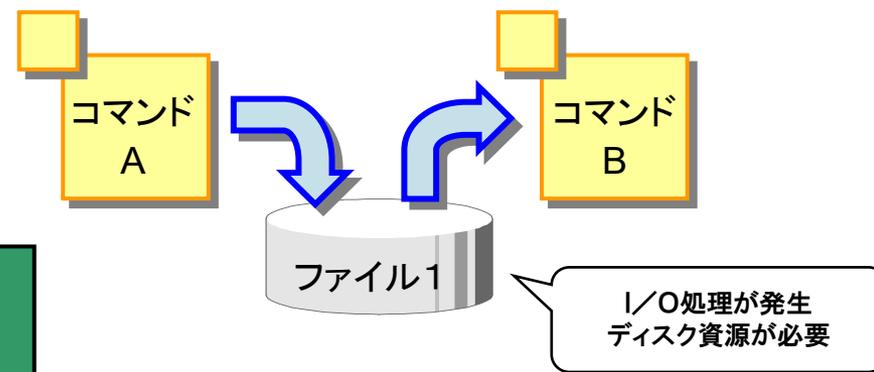
■ パイプを利用するとコマンドの出力結果を別のコマンドに渡すことができます

例えば、コマンドAの結果をファイルに出力して、そのファイルを別のコマンドに渡し処理する場合、下記のコマンド実行が必要

```
# コマンドA > ファイル1
# コマンドB < ファイル1
```

例)telnetプロセスの調査

```
$ ps -ef > /tmp/pslist
$ grep telnet < /tmp/pslist
root 9926  421  0 13:29:37 ?        0:00 /usr/sbin/in.telnetd
root 10661 421  0 14:21:43 ?        0:00 /usr/sbin/in.telnetd
$
```

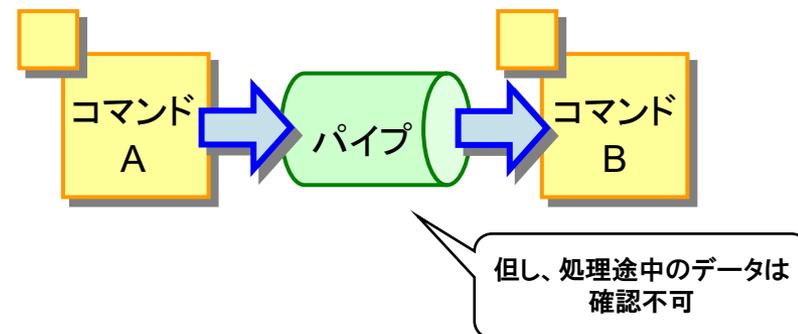


パイプ(|)を使いコマンドを連結させ実行する

```
# コマンドA | コマンドB
```

例)telnetプロセスの調査

```
$ ps -ef | grep telnet
root 9926  421  0 13:29:37 ?        0:00 /usr/sbin/in.telnetd
root 10661 421  0 14:21:43 ?        0:00 /usr/sbin/in.telnetd
$
```



- ・I/O処理が発生しないのでより高速処理が可能
- ・複数(2つ以上)のコマンドを連結可能

<参考>リダイレクション (標準入力/出力/標準エラー出力)

- Bシェルの場合 (標準出力/標準エラー出力: ディスプレイ、標準入力: キーボード)

```
# ls -l test hosts
test: ファイルもディレクトリもありません。
-r--r--r-- 1 root  sys      174 7月 20日 14:42 hosts
#
```

- エラーメッセージを分離

```
# ls -l test hosts 2> stderr.txt > stdout.txt
#
```

標準出力の切換(1>と同等)
'>>'はファイルに追加

標準エラー出力
の切換(2>)

stdout.txt

```
-r--r--r-- 1 root  sys      174 7月 20日 14:42 hosts
```

stderr.txt

```
test: ファイルもディレクトリもありません。
```

- 標準エラー出力を標準出力に切換 (メッセージ全てを同一のファイルに格納)

```
# ls -l test hosts > stdout.txt 2>&1
```

Cシェルの場合

```
# ls -l test hosts >& message
注) エラーメッセージを分離できない。
```

- キーボード入力 (標準入力) をファイルに切換

```
# コマンド < stdin.txt
```

■ マニュアル利用方法

man コマンド名[or ファイル名]

【例) manコマンド】

```
# man man
マニュアルを清書中です。しばらくお待ちください... 終了
ユーザーコマンド                man(1)
```

【名前】 man - マニュアルページの表示 **日本語表示**

【形式】
man [-] [-adFlrt] [-M path] [-T macro-package] [-s section] name...

man [-M path] -k keyword...
man [-M path] -f file...

【機能説明】
man コマンドはマニュアルページから得た情報を表示します。具体的には、name 引数が示すタイトルを持つマニュアルページの全内容を出力します。また -k オプションまたは -f オプションが指定された場合には、それぞれ keyword または filename が示すキーワードやファイル名により選択された 1 行からなる要約情報を出力します。指定条件に一致するマニュアルページが見つからない場合には、man はエラーメッセージを出力します。

【ソース形式】
参照用マニュアルページには、nroff (nroff(1)を参照) または SGML (Standard Generalized Markup Language) (sgml(5)を参照)

-- 継続 --(6%) **ページ単位で表示**

【例) defaultrouterファイル】

```
# man defaultrouter
マニュアルを清書中です。しばらくお待ちください... 終了
File Formats                      defaultrouter(4)
NAME
  defaultrouter - configuration file for default router(s)
SYNOPSIS
  /etc/defaultrouter 英語表示
DESCRIPTION
  The /etc/defaultrouter file specifies a IPv4 host's default router(s).
  The format of the file is as follows:
  IP_address
  ...
  The /etc/defaultrouter file can contain the IP addresses or hostnames of one or more default routers, with each entry on its own line. If you use hostnames, each hostname must also be listed in the local /etc/hosts file, because no name service is available for hostnames.
vi
Li
r
T
b
/e
b
-- 継続 --(100%)
```

留意事項

- オンラインマニュアルは全てが日本語ではない
- OSで標準提供されないマニュアル(ミドルウェア等)は環境変数(MANPATH)に設定する

■ Solaris OSのネットワーク管理系コマンド

コマンド		概要
TCP/IP プロトコル	ping	IP接続をテストする。通信できない区間を特定して、ネットワークの問題かコンピュータの問題かなどを絞り込む。
	ifconfig	現在の TCP/IP 設定を表示することにより、TCP/IP の設定に問題がないかの確認や、一時的な TCP/IP の設定、有効/無効の設定が可能。 (サブネットマスク、ブロードキャストアドレスの設定も可能)
	netstat	ルーティングテーブルの表示、プロトコルの統計情報と現在の TCP/IP 接続状態を確認することができる。
	route	ローカルルーティングテーブルの変更を行う。設定に問題がないかどうかは「traceroute」コマンドを使用して確認する。
	nslookup	DNSサーバに照会することにより、DNSサーバに登録されているレコード情報など取得する。また、ローカルコンピュータのDNS設定に問題がないか確認することもできる。
	traceroute	パケットが送信先までにたどるルートを追跡する。ネットワーク上に複数の経路がある場合に、正しい経路を使用しているか、ルータの設定に問題がないかなどを確認する。経路の設定は「route」コマンドで行う。
	arp	IP アドレスと物理アドレスの対応表を表示、修正する。 特定のコンピュータ間のアドレス解決に関する問題を調査する。

特に、ifconfigコマンドはネットワークインターフェースの設定/変更やIPアドレスの確認/変更などを実行する、ネットワークの基本コマンドです。

- IPアドレスとMACアドレスの対応表

arp -a

```
# arp -a
Net to Media Table: IPv4
Device  IP Address      Mask            Flags  Phys Addr
-----  -
bge0    10.77.116.1     255.255.255.255  00:00:0c:07:ac:00
bge0    10.77.116.13    255.255.255.255  00:00:e2:3b:96:be
bge0    10.77.116.181   255.255.255.255  00:80:17:28:28:95
bge0    sv-pw-05        255.255.255.255 SP  00:80:17:28:7a:0e
bge0    10.77.116.202   255.255.255.255  00:00:e2:2f:40:dc
bge0    224.0.0.0       240.0.0.0       SM    01:00:5e:00:00:00
```

- TCP/IP構成情報

ifconfig -a

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ffffffff
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 10.73.13.123 netmask fffffff0 broadcast 10.73.13.255
    ether 0:80:17:XX:36:ab
```

- ルーティングテーブル表示

netstat -rn

```
# netstat -rn

Routing Table: IPv4
Destination      Gateway          Flags Ref  Use  Interface
-----
10.77.116.0      10.77.116.135  U      1    7  bge0
224.0.0.0        10.77.116.135  U      1    0  bge0
default          10.77.116.1    UG     1   12
127.0.0.1        127.0.0.1     UH     8 52297 lo0
```

- ルーティングテーブル設定

route add/delete

```
# route add 10.77.114.0 10.77.116.135
add host 10.77.114.0: gateway 10.77.116.135

# route add default 10.77.116.1
add net default : gateway 10.77.116.1

# route delete 10.77.114.0 10.77.116.135
delete host 10.77.114.0: gateway 10.77.116.135
```

● 名前解決

nslookup

```
# nslookup

Default Server: hornet.ist.support.fujitsu.com
Address: 10.77.116.10

> seal.ist.support.fujitsu.com
Server: hornet.ist.support.fujitsu.com
Address: 10.77.116.10

Name: seal.ist.support.fujitsu.com
Address: 10.77.115.167

>
```

● ネットワークトレース

tracert

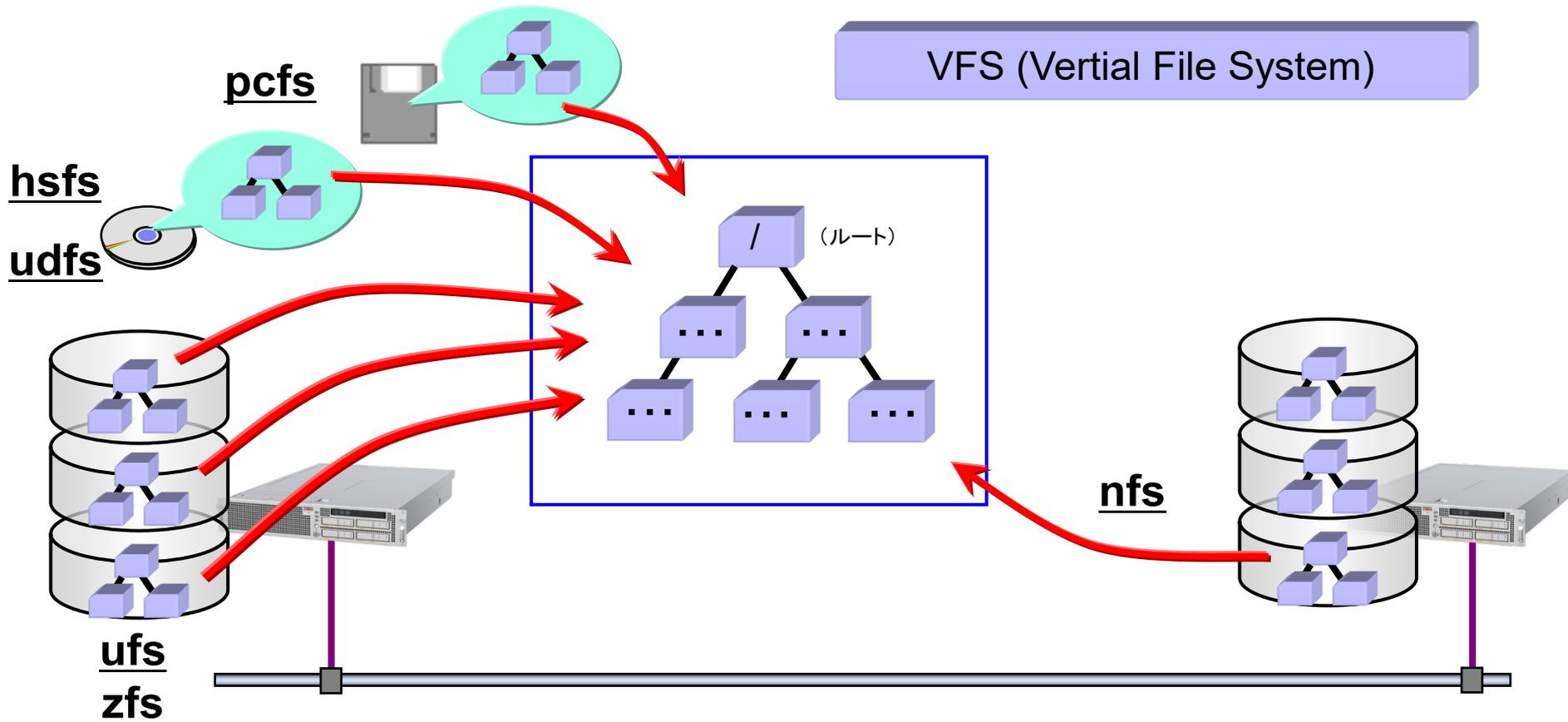
```
# traceroute -I 10.77.160.26
traceroute to 10.77.160.26 (10.77.160.26), 30 hops max, 40 byte packets
 1 10.77.116.2 (10.77.116.2) 0.596 ms 0.403 ms 0.370 ms
 2 10.77.160.26 (10.77.160.26) 0.387 ms 0.345 ms 0.313 ms
#
```

■ Solaris でサポートされるファイルシステム

ディスクベースのファイルシステム `ufs` `zfs` `hsfs` `pcfs` `udfs`

分散ファイルシステム `nfs`

擬似ファイルシステム `tmpfs` `swapfs` `fdfs` `procfs`

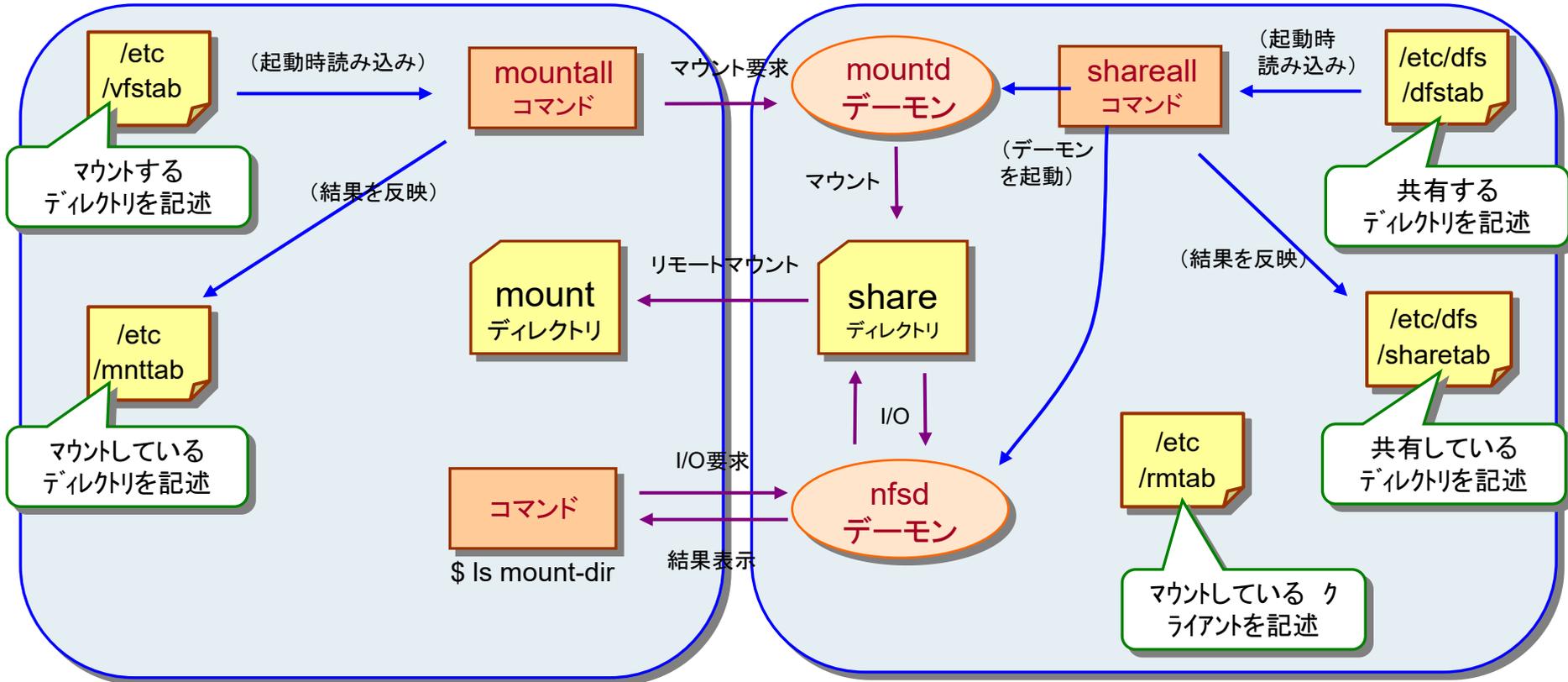


NFS (Network File System) の仕組み

■ NFSはサーバ、クライアント間でネットワークを介したファイル共有の仕組みです

●クライアント (SMF: svc:/network/nfs/client:default)

●サーバ (SMF: svc:/network/nfs/server:default)



<参考>NFSデーモンの確認方法

```
# rpcinfo -b mountd 3
```

NFSバージョン (Solaris 10からバージョン2~4をサポート) どのシステムがmountdを実行しているかわかる

■ share (シェア)コマンド

① 手動でシェアする方法

```
# share -F nfs -o ro,anon=0 /export/home/usr1
```

② 起動時に自動でシェアする方法 /etc/dfs/dfstab ファイルに下記を追加

```
share -F nfs -o ro,anon=0 /export/home/usr1
```

```
# shareall ← 即時実行する場合
```

- シェアの状況を書込むファイル
/etc/dfs/sharetab
- 現在のシェア状況の確認コマンド
share

■ mount (マウント)コマンド

① 手動でマウントする方法

```
# mount -F nfs [host名]:/export/home/usr1 [mountポイント]
```

② 起動時に自動でマウントする方法 /etc/vfstab ファイルに下記を追加

```
#device    device    mount    FS    fsck    mount    mount  
#to mount  to fsck   point    type  pass   at boot  options  
(略)  
[host名]:/export/home/usr1 - /mnt nfs - yes -
```

```
# mountall ← 即時実行する場合
```

- マウントの状況を書込むファイル
/etc/mnttab
- 現在のマウント状況の確認コマンド
mount

<参考> NFS関連ファイル

■ NFSの機能を実現するファイル群

ファイル名	機能
/etc/default/fs	ローカルファイルシステムにおけるデフォルトファイルシステムのタイプを示します。
/etc/default/nfs	lockd および nfsd の構成情報を示します。
/etc/default/nfslogd	NFS サーバーログデーモン (nfslogd) の構成情報を示します。
/etc/dfs/dfstab	共有するローカルディレクトリ情報を設定します。追加する場合、手動で編集します。
/etc/dfs/fstypes	リモートファイルシステムにおけるデフォルトファイルシステムのタイプを示します。
/etc/dfs/sharetab	共有されるローカルとリモートのリソースを示します。このファイルは編集しないでください。
/etc/mnttab	自動マウントしたディレクトリを含む、現在マウントしているファイルシステムを示します。このファイルは編集しないでください。
/etc/netconfig	トランスポートプロトコルのリストです。このファイルは編集しないでください。
/etc/nfs/nfslog.conf	NFS サーバーログのための一般的な構成情報を示します。
/etc/nfs/nfslogtab	nfslogd によるログ後処理のための情報を示します。このファイルは編集しないでください。
/etc/nfssec.conf	NFS のセキュリティサービスのリストです。このファイルは編集しないでください。
/etc/rmtab	NFS クライアントがリモートでマウントしたファイルシステムを示します。このファイルは編集しないでください。
/etc/vfstab	ファイルシステムを自動マウントさせる設定をします。追加する場合、手動で編集します。

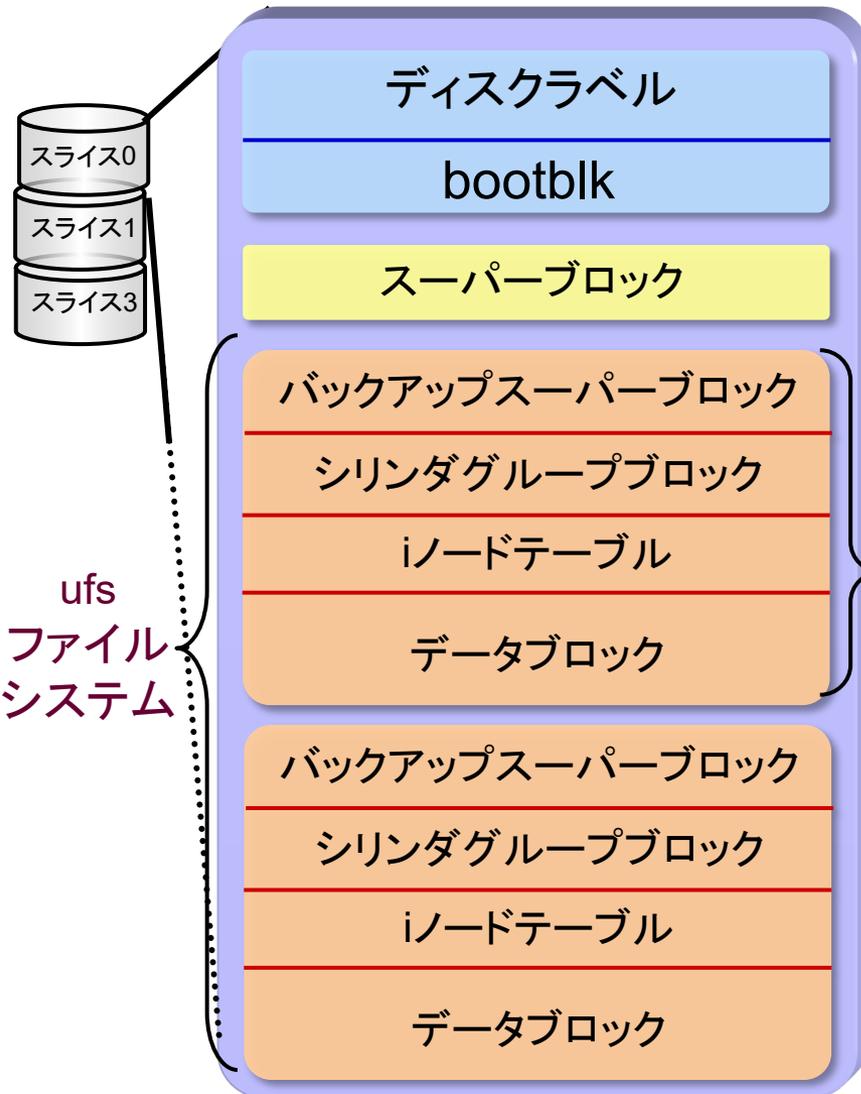
特に、/etc/vfstabファイルはシステムの起動に影響するため編集には注意が必要です。

システム起動時には、/etc/vfstabの記述に基づき、自動的にマウントが行われる。
定常的なマウント情報は、このファイルに登録する。

/etc/vfstabファイル内容

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
fd	-	/dev/fd	fd	-	no	-
/proc	-	/proc	proc	-	no	-
/dev/dsk/c0t0d0s1	-	-	swap	-	no	-
/dev/dsk/c0t0d0s0	/dev/rdisk/c0t0d0s0	/	ufs	1	no	-
/dev/dsk/c0t0d0s6	/dev/rdisk/c0t0d0s6	/usr	ufs	1	no	-
/dev/dsk/c0t0d0s7	/dev/rdisk/c0t0d0s7	/export/home	ufs	2	yes	-
/dev/dsk/c0t0d0s5	/dev/rdisk/c0t0d0s5	/opt	ufs	2	yes	-
swap	-	/tmp	tmpfs	-	yes	-

- device to mount : mountコマンドを実行する時のデバイスファイル名(ブロック型)
- device to fsck : fsckコマンドを実行する時のデバイスファイル名(キャラクタ型)
- mount point : マウントポイントとなるディレクトリ
- FS type : ファイルシステムのタイプ
- fsck pass : システム起動時に、fsckコマンドにより、ファイルシステムチェックを行う順序
- mount at boot : yesであれば、システム起動時に自動マウント
- mount options : マウントオプション(mountコマンドのオプション)



ディスクラベル (VTOC)

ディスクラベル (VTOC) はディスクのパーティションテーブルを格納。最初のセクタ (512 バイト) に格納。

ブートブロック

ブートプログラム (bootblk) は VTOC に続く 15 セクタに格納。

スーパーブロック (一次)

ファイルシステムに関する以下の情報を格納。ブートブロックに続く 16 セクタに格納。

- ・データブロックの数
- ・シリンダグループの数
- ・データブロックとフラグメントのサイズ
- ・ハードウェアの説明
- ・マウントポイントの名前
- ・ファイルシステムの状態フラグ
(clean, stable, active, logging, unknown)

バックアップスーパーブロック

スーパーブロックの損失に備えて、各シリンダグループ内に複製されている。ファイルシステム作成時に行われる。

■ 新規ディスクの設定

①パーティションを設定します。

format

Searching for disks...done

AVAILABLE DISK SELECTIONS:

0. c0t0d0 <FUJITSU-MAJ3182MC-3702 cyl 12435 alt
/pci@17,4000/scsi@3/sd@0,0

1. c0t1d0 <FUJITSU-MAJ3182MC-3702 cyl 12435 alt
/pci@17,4000/scsi@3/sd@1,0

Specify disk (enter its number): **1**

selecting c0t1d0

[disk formatted]

FORMAT MENU:

disk - select a disk
type - select (define) a disk type
partition - select (define) a partition table
current - describe the current disk
format - format and analyze the disk
repair - repair a defective sector
label - write label to the disk
analyze - surface analysis
defect - defect list management
backup - search for backup labels
verify - read and display labels
save - save new disk/partition definitions
inquiry - show vendor, product and revision
volname - set 8-character volume name
!<cmd> - execute <cmd>, then return
quit

format>

format> **partition**

PARTITION MENU:

0 - change `0' partition
1 - change `1' partition
2 - change `2' partition
3 - change `3' partition
4 - change `4' partition
5 - change `5' partition
6 - change `6' partition
7 - change `7' partition

select - select a predefined table

modify - modify a predefined partition table

name - name the current table

print - display the current table

label - write partition map and label to the disk

!<cmd> - execute <cmd>, then return

quit

partition> **print**

Current partition table (original):

Total disk cylinders available: 12435 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks	
0	root	wm	0 - 91	128.93MB	(92/0/0)	264040
1	swap	wu	92 - 183	128.93MB	(92/0/0)	264040
2	backup	wu	0 - 12434	17.02GB	(12435/0/0)	35688450
3	unassigned	wm	0	0	(0/0/0)	0
4	unassigned	wm	0	0	(0/0/0)	0
5	unassigned	wm	0	0	(0/0/0)	0
6	usr	wm	184 - 12434	16.77GB	(12251/0/0)	35160370
7	unassigned	wm	0	0	(0/0/0)	0

partition>

ディスクのパーティショニング

② スライスのサイズを指定します。

```
partition> p
Current partition table (unnamed):
Total disk cylinders available: 12435 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	unassigned	wm	0 - 3653	5.00GB	(3654/0/0) 10486980
1	unassigned	wu	10903 - 12364	2.00GB	(1462/0/0) 4195940
2	backup	wm	0 - 12434	17.02GB	(12435/0/0) 35688450
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	unassigned	wm	0	0	(0/0/0) 0
7	unassigned	wm	0	0	(0/0/0) 0

```
partition> 1
Part Tag Flag Cylinders Size
1 unassigned wu 10903 - 12364 2.00GB
```

Enter partition id tag[unassigned]:
 Enter partition permission flags[wu]:
 Enter new starting cyl[10903]: **3654**
 Enter partition size[4195940b, 1462c, 5115e, 2048.80mb, 2.00gb]: **5g**

スライス毎に
設定する

③ ラベルを設定します。

```
partition> label
Ready to label disk, continue? y
```

※ 上記を実行して初めてディスク(VTOC)に情報が書き込まれます。

スライスのサイズ指定時の留意ポイント

ポイント1

最初のスライス(0番目)の先頭にはVTOC領域が格納されています。そのため、0番目スライスをrawデバイスとして使用する場合は、開始シリンダ番号を1に設定してください。

ポイント2

スライス間のシリンダ番号は重複させないように設定してください。重複したシリンダ領域に書き込みがあるとデータに不整合が起こりファイルシステムが破壊されます。

ポイント3

スライスはシリンダ境界で区切られるため、スライスの構成単位はそのディスクの1シリンダのサイズとなります。

左記の例の場合:

1シリンダあたり = 1435バイト

ファイルシステムの作成

④ ファイルシステムを作成します。

```
# newfs /dev/rdisk/c0t1d0s0
newfs: 新しいファイルシステム /dev/rdisk/c0t1d0s0 を作成しま
すか: (y/n)? y
/dev/rdisk/c0t1d0s0: 全セクター数: 261170 (シリンダ数: 91、トラック数: 5、
セクター数: 574)
    127.5MB、6 シリンダグループ (16 c/g, 22.42MB/, 10752 i/g)
スーパーブロックのバックアップの位置 (fsck -F ufs -o b=# のため):
    32, 46528, 93024, 139520, 186016, 232512,
#
```

⑤ ファイルシステムの整合性チェックを実施します。

```
# fsck -y /dev/rdisk/c0t1d0s0
** /dev/rdisk/c0t1d0s0
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 9 used, 122399 free (15 frags, 15298 blocks, 0.0% fragmentation)
#
```

⑥ マウントポイントを作成します。

```
# mkdir /test
```

⑦ マウントを実行します。

```
# mount -F ufs /dev/dsk/c0t1d0s0 /test
```

⑧ マウントの確認をします。

```
# mount
.....
/test on /dev/dsk/c0t1d0s0 read/write/setuid/intr/largefiles/xattr/
onerror=panic
/dev=800008 on (月) 9月 13 15:16:46 2004
```

⑨ サイズを確認します。

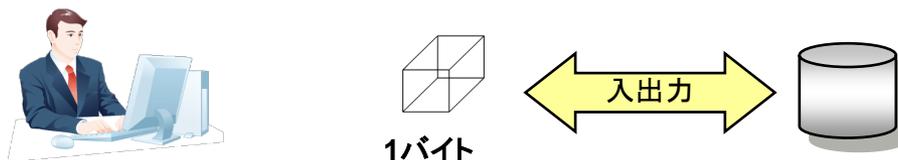
```
# df -h
```

ファイルシステム	サイズ	使用済み	使用可能	容量	マウント先
/dev/dsk/c0t0d0s0	65G	10G	54G	17%	/
/devices	0K	0K	0K	0%	/devices
(略)					
/dev/dsk/c0t1d0s0	5G	0G	5G	0%	/test

■ キャラクタ型とブロック型デバイスの違い

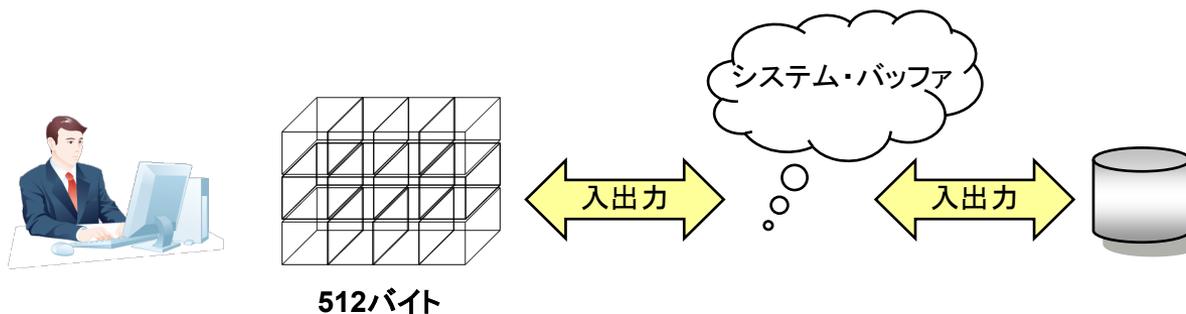
1) キャラクタ型デバイス (=rawデバイス) </dev/rdisk ディレクトリ>

キャラクタ型とは
キャラクタ型デバイスは1バイトを単位として入出力する装置です。



2) ブロック型デバイス </dev/dsk ディレクトリ>

ブロック型とは
ブロック型デバイスは512バイトなどの固定長ブロックを単位として入出力する装置です。



<参考>デバイス名と使用コマンド

■ コマンドと論理デバイス・ファイルの種類

コマンド	機能	引数に指定する論理 デバイス・ファイルの種類	コマンド実行例
df	ディスクの使用状況の確認	ブロック型	df /dev/dsk/c0t0d0s7
mount	ファイルシステムのマウント		mount /dev/dsk/c0t0d0s7 /export/home
fsck	ファイルシステムの矛盾の検出/修復	キャラクタ型	fsck /dev/rdisk/c0t0d0s0
newfs	ファイルシステムの作成		newfs /dev/rdisk/c0t0d0s7
prtvtoc	ディスクラベルVTOCの内容確認		prtvtoc /dev/rdisk/c0t0d0s2

ブロック型

キャラクタ型
(ローデバイス)

論理デバイス名

/dev/dsk/c0t0d0s0

物理デバイス名

/devices/pci@17,4000/scsi@3/sd@0,0:a

/dev/rdisk/c0t0d0s0

/devices/pci@17,4000/scsi@3/sd@0,0:a,raw



キャラクタ型を示す



キャラクタ型を示す

ディスクへのアクセス方式の違いにより2通りのデバイス名を使い分けている

ブロック型 = ランダムアクセス キャラクタ型 = シーケンシャルアクセス

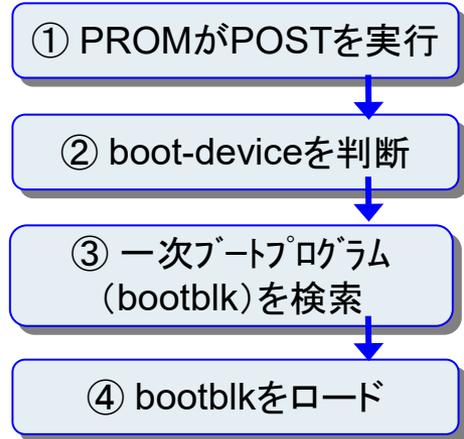
詳細は、設計・運用手順書
「4. 各種設定」
を参照してください。

5. システムの起動とシャットダウン

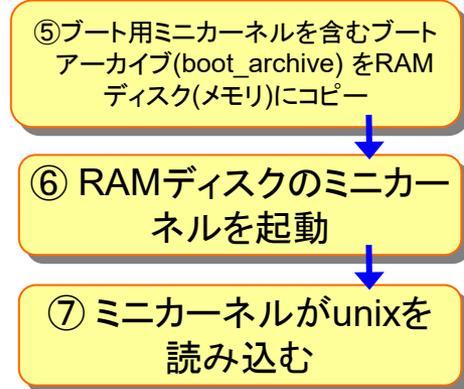
OS起動までの流れ

●電源投入時の処理

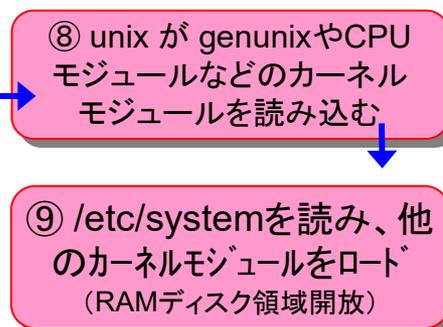
《ブートPROMフェーズ》



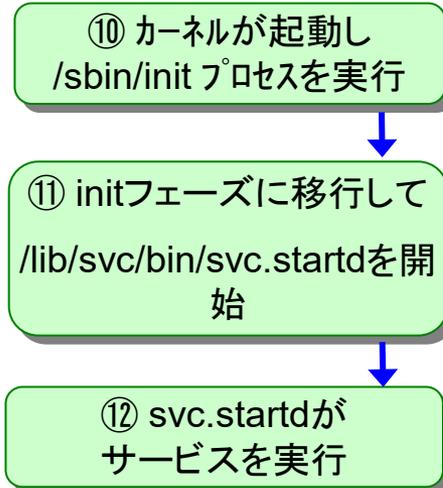
《ブートプログラムフェーズ》



《カーネル初期化フェーズ》



《initフェーズ》



コンソールメッセージ(例)

(POST実行:①)

```
SPARC Enterprise T2000, No Keyboard
Copyright 2008 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.30.0, 2048 MB memory available, Serial #70065586.
Ethernet address 0:14:4f:2d:1d:b2, Host ID: 842d1db2.
(↑バナー表示)
```

```
Boot device: /pci@780/pci@0/pci@9/scsi@0/disk@0,0:a File and args:
(↑ブートデバイスを判断、制御がブートプログラムへ:②~④)
```

```
SunOS Release 5.10 Version Generic_139555-08 64-bit
(↑このメッセージ表示後、制御がカーネルへ:⑤~⑦)
```

```
Copyright 1983-2009 Sun Microsystems, Inc. All rights reserved
.(↑カーネルが実行を開始:⑧~⑨)
```

(以下、initプロセスによるシステムの起動:⑩~⑫)

```
Use is subject to license terms.
Hostname: t2000-3
/dev/rdisk/c0t1d0s0 is clean
Reading ZFS config: done.
```

(システム起動完了)

```
t2000-3 console login:
```

Solaris 10 10/08[Update6]から新しいブートの仕組みに変更されています。

(ブートフェーズにおけるミニカーネルを含むアーカイブ (boot_archive)の部分)

■ OBP (Open Boot PROM) とは、

電源投入時に、PROM (Programmable Read-Only Memory) により、ハードウェアの Power On Self Test (POST) が実行され、その後 **ファームウェア (OpenBoot)** の環境が提供されます。ファームウェア上では OS 起動前のハードウェアに関する設定やテストを実行できます。

● 構成データの表示

コマンド	説明
banner	電源投入時のメッセージを表示する
devalias	全てのデバイスエイリアスを表示する
.enet-addr	Ethernet アドレスを表示する
.idprom	整形された ID PROM コンテンツを表示する
module-info	CPU 速度を表示する (2.x のみサポート)
printenv	全ての NVRAM パラメータとそのデフォルト値を表示する
show-devs	システムに接続された全てのデバイスを表示する
.speed	CPU 速度とバス速度を表示する (3.x のみサポート)
.traps	SPARC トラップのタイプを一覧表示する
.version	ブート PROM のバージョンとデータを表示する

● ブートデバイスの指定

コマンド	説明
boot	システムを起動する
boot cdrom	ローカル CD-ROM から起動する
boot disk	デフォルトハードディスクから起動する
boot floppy	フロッピーから起動する
boot net	ネットワークから起動する
boot tape	SCSI テープから起動する

● ハードウェア診断コマンド

コマンド	説明
pcia-probe-list	PCI バスをテストする
probe-scsi	内蔵 SCSI バスのデバイス接続をテストする
probe-scsi-all	全ての SCSI バスのテストをする
test-all	インストールされているデバイスのグループをテストする
test floppy	フロッピードライブをテストする
test /memory	システムメモリをテストする
test net	オンボード Ethernet インターフェースをテストする
watch-clock	システムクロックを監視する
watch-net	ネットワーク接続を監視する

システムの起動コマンド

■ boot コマンド

```
ok boot
```

OBP環境上で bootコマンドを実行すると、自動的に**マルチユーザモード**と呼ばれるシステムモードでOSが起動します。

起動時のオプションの指定が可能です。

マルチユーザモード (ランレベル:3)

システムはログインした複数のユーザがシステムとデータアクセスするために必要な全てのプロセスとサービスを実行します。

シングルユーザモード (ランレベル:1 or s,S)

システムは最小のプロセスとサービスだけを実行しており、一般ユーザはログインできません。メンテナンスモードとも呼ばれます。

● -s オプション

システムを**シングルユーザモード**で起動します。

```
ok boot -s
```

● -a オプション

システムを**対話型**で起動します。起動時に読み込むシステムファイルやカーネルを一時的に変更する場合に使用します。

```
ok boot -a
Enter filename [kernel/sparcv9/unix]:
Enter default directory for modules [/platform/FJSV,GPUS/kernel
/platform/sun4us/kernel /kernel /usr/kernel]:
Name of system file [etc/system]:
root filesystem type [ufs]:
[/pci@17,4000/scsi@3/disk@0,0:a]:
```

※ 主に起動フェーズでの障害発生時に利用します。

● -r オプション

再構成ブートを行います。新しく接続したデバイスが検出され、/devices と/dev に新しいデバイスエントリが追加されます。また、/etc/path_to_inst ファイルも更新されます。(= devfsadmコマンド)

```
ok boot -r
```

● フェイルセーフモード (Solaris 10 10/08以降)

ブートアーカイブ(boot-archive)からの起動が不可になった場合、フェイルセーフ用アーカイブ(failsafe archive)から起動し修復を行います。

```
ok boot -F failsafe
```

ランレベル

ユーザが現在使用できるサービスを定義するシステム状態を表す数字、英字。システムは常に1つのランレベルで動作しています。

ランレベル	機能
0	Solaris環境をシャットダウンし、ブートPROMのok プロンプトを表示。
1 または s,S	シングルユーザモードとして動作。最小限のサービスだけが動作。
2	マルチユーザモードとして動作。NFSサーバデーモン以外のデーモンが動作します。
3 (デフォルト)	マルチユーザモードとして動作。全てのデーモンが動作します。デフォルトのランレベルとして/etc/inittabで指定されています。
4	(実装されていません。)
5	Solaris環境をシャットダウンし、システムの電源を切断します。
6	システムをランレベル0にシャットダウンしてからマルチユーザモードに再起動します。

ランレベルの確認方法

```
# who -r
. run level 3 Jun 9 08:30 3 0 S
```

現在のランレベル

ランレベルを最後に更新した日時

現在のランレベル

一番最近リポートして依頼のランレベルの変更回数

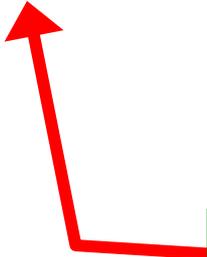
前回のランレベル

- カーネルが /sbin/initプロセスを起動し、initプロセスは、/etc/inittabファイルの記述に従いサービスの起動方法を決定します。

/etc/inittabファイルの各エントリ

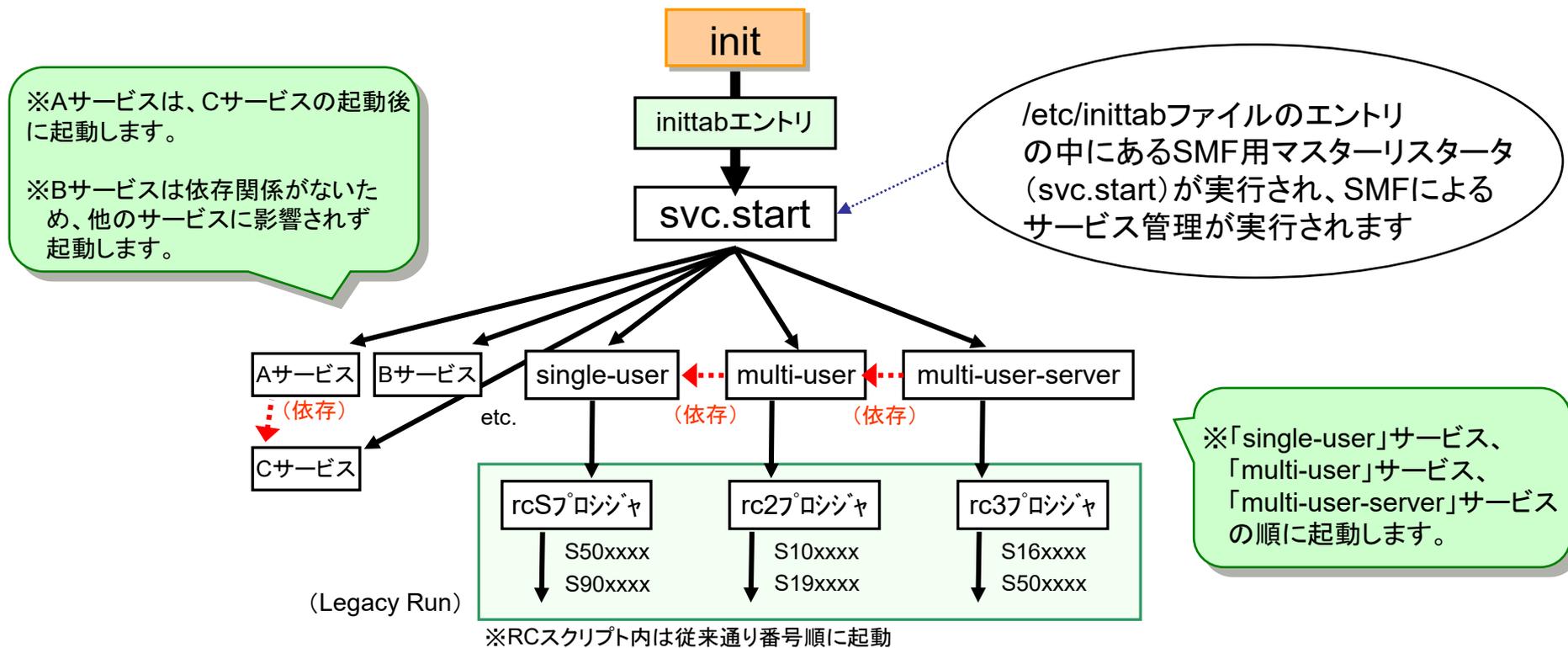
```
ap::sysinit:/sbin/autopush -f /etc/iu.ap                ... (1)
sp::sysinit:/sbin/soconfig -f /etc/sock2path            ... (2)
smf::sysinit:/lib/svc/bin/svc.startd >/dev/msglog 2<>/dev/msglog </dev/console ... (3)
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/msglog ... (4)
```

- (1)STREAMSモジュールを初期化します。
- (2)ソケット転送プロバイダを構成します。
- (3)SMF用マスターリスタータを初期化します。
- (4)電源障害の場合のシャットダウンを指定します。



サービス起動の仕組み

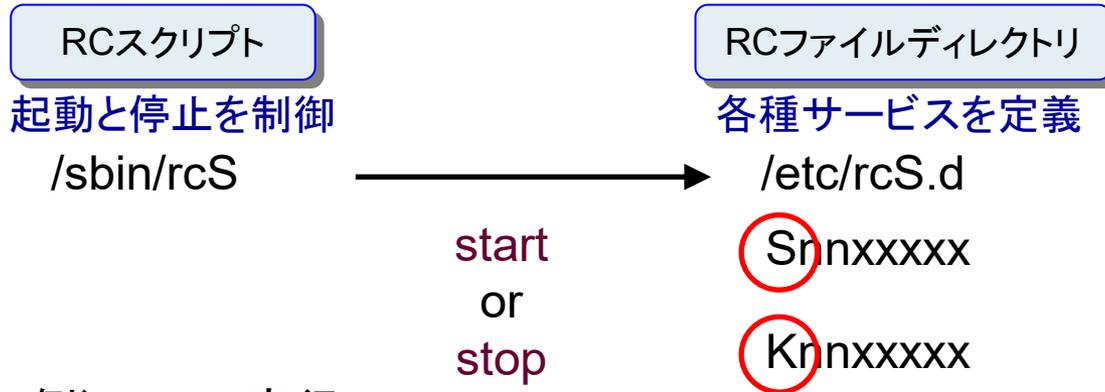
- SMFによるサービス管理の特長は、サービス間の依存関係を管理していること



- ・依存関係のないサービスは並列起動するため、従来のサービス管理よりサービス起動時間が短縮されます。
- ・サービス起動に失敗した場合などは依存関係を確認して、原因となるサービスを突き止めることが可能です。

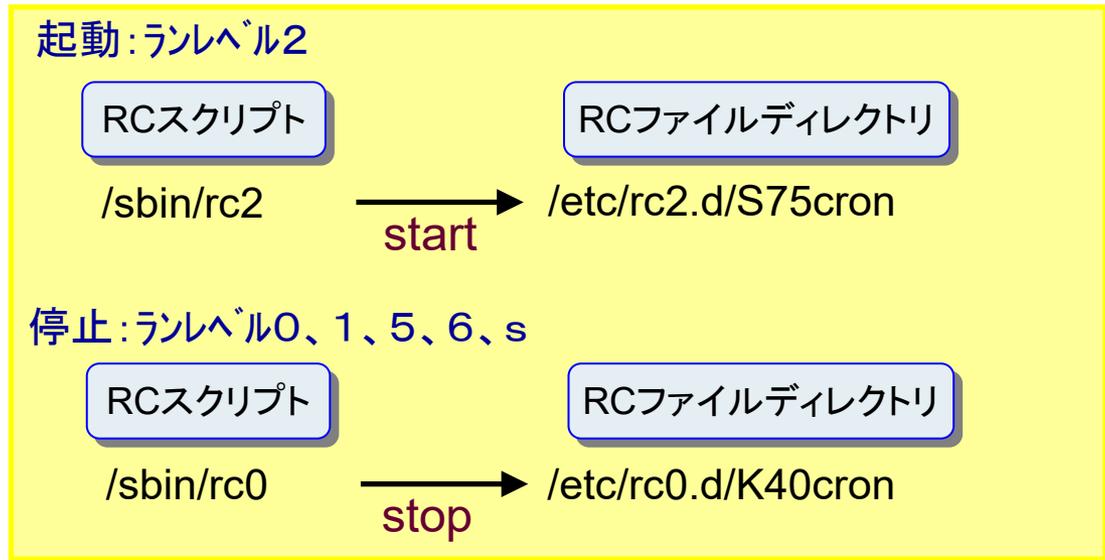
実行制御 (RC) スクリプト

- SMFにより指定されたランレベルのサービスを実行するスクリプト



S→起動 K→停止
nn : 2桁の数字(実行順番)
xxxxx : サービス名

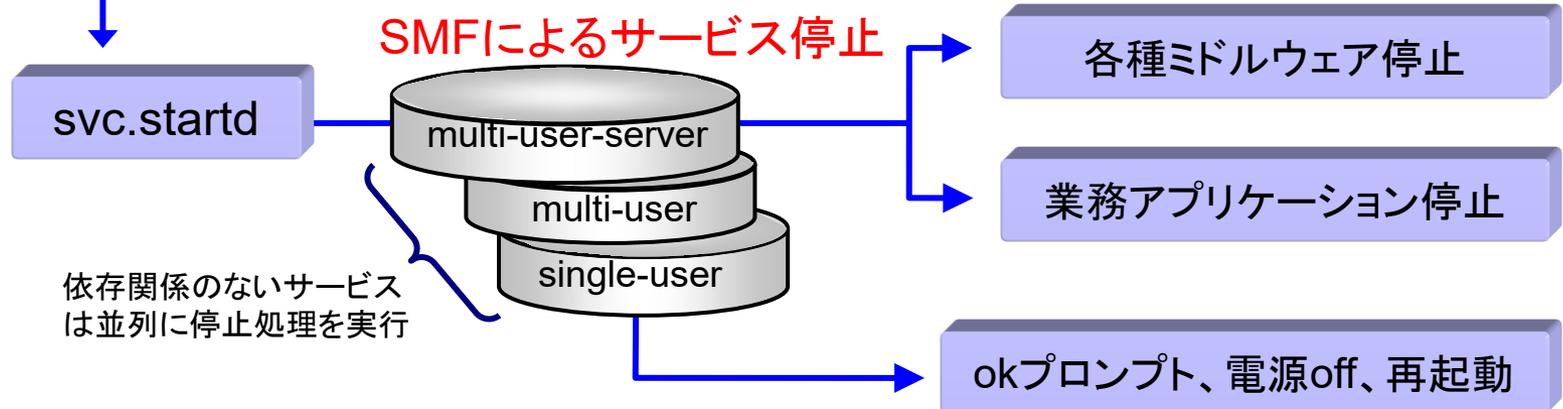
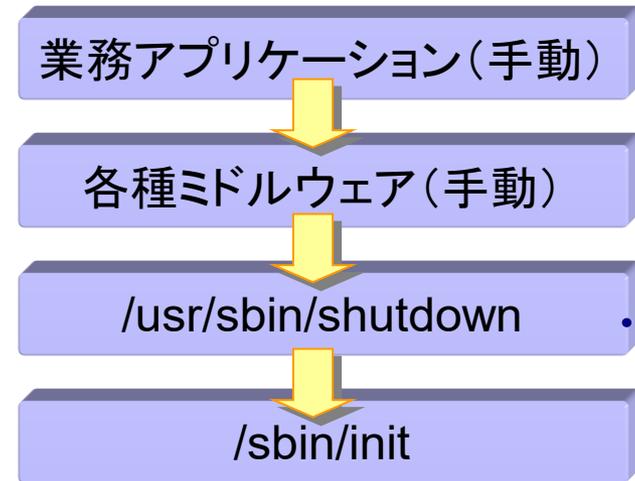
例) cronの実行



全てのRCファイルは/etc/init.dディレクトリにコピーされています。

/etc/init.d/cron start
はRCスクリプトの実行と同じ意味

●システム停止処理の流れ



通常運用時のOS停止コマンド

```
# /usr/sbin/shutdown -y -g0 -i5
```

-i: 実行レベル(ランレベル)
0: OBPモード (okプロンプト)
5: 電源切断
6: 再起動
s, S: シングルユーザモード
-g: 停止開始までの待ち合わせ時間 (秒)
-y: 確認応答なし (付けない場合、60秒後に確認が出る)

コマンド	説明	用途
shutdown	init を呼び出してシステムをシャットダウンする実行可能なシェルスクリプト。デフォルトでは、システムは実行レベル S に移行する。	実行レベル 3 で動作しているサーバーで使用する。サーバーにログインしているユーザに、サーバーが間もなくシャットダウンされることが通知される。シャットダウンされるサーバーのリソースをマウントしているシステムにも通知される。
init	すべてのアクティブなプロセスを終了し、ディスクを同期させてから実行レベルを変更する実行可能ファイル。	他のユーザが影響を受けないスタンドアロンシステムで使用する。ユーザはまもなく行われるシャットダウンについて通知されないため、シャットダウンにかかる時間が短くて済む。
reboot	ディスクを同期させ、ブート命令を uadmin システムコールに渡す実行可能ファイル。実際にプロセッサを停止するのは、uadmin システムコールである。	推奨されない。代わりに、init コマンドを使用する
halt	ディスクを同期させ、プロセッサを停止する実行可能ファイル。	/etc/rc0 スクリプトを実行しないので推奨されない。このスクリプトは、すべてのプロセスを停止し、ディスクを同期させ、すべてのファイルシステムをマウント解除する。

システムを安全に停止するために、**reboot**コマンドや **halt**コマンドによるシステムの停止・再起動は推奨しません。

詳細は、設計・運用手順書
「5. システムの起動とシャットダウン」
を参照してください。

6. バックアップ/リストア

■ バックアップの目的

データの退避・復元は、もっとも重要なシステム管理の1つです。
次のような原因によるデータの消失に備えて、定期的にシステムファイルの退避を実行する必要があります。

- ✓ ソフトウェア障害によるシステムファイルの破壊
- ✓ ハードウェアの障害や故障
- ✓ 不注意によるファイルの削除
- ✓ 天災(火災、台風、地震など)

■ バックアップ計画

管理者はシステムファイルの退避・復元をどのように実施していくか(バックアップ計画と呼ぶ)を考え、システムの長期運用に備えます。バックアップ計画で検討する内容は次のとおりです。

- ✓ バックアップを作成するファイルシステムの選定
- ✓ バックアップ方法の選択
- ✓ テープ装置の選択
- ✓ ファイルシステムの退避および復元の手順(作業マップ)

■ バックアップのタイミング

システムファイルの退避は以下のタイミングで実施してください。

- ✓ システム構築時
- ✓ 定期PTF(修正パッチ)適用時
- ✓ ソフトウェアの追加・削除の時
- ✓ システム構成(ハードウェア)の変更時

コマンド	概要	注意事項
<p>ufsdump</p>	<ul style="list-style-type: none"> ・UFSファイルシステムをバックアップします。 ・バックアップされたデータはufsdumpフォーマットで記録されます。 ・ファイル変更日付に基づいて、増分バックアップ及び、増分リストアを実行する機能があります。 ・バックアップ先として、ファイルシステム上のファイルやテープ媒体を使用することができます。 	<ul style="list-style-type: none"> ・ufsdumpでは、ディスクのパーティション構成とマウントポイントの情報はバックアップすることが出来ないため、別途記録しておく必要があります。 ・一つのテープ媒体へ複数のバックアップを行う場合は、デバイス名に「n」(テープを巻き戻さない)をつけて(例:/dev/rmt/0n)バックアップを行ってください。
<p>ufsrestore</p>	<ul style="list-style-type: none"> ・ufsdumpフォーマットで記録されたファイルやテープ媒体から、UFSファイルシステムにデータをリストアします。 ・カレントディレクトリにデータをリストアする為、バックアップ時と異なるUFSファイルシステム上にデータをリストアすることが出来ます。 ・ファイル変更日付に基づいて、増分リストアを実行する機能があります。 	<ul style="list-style-type: none"> ・リストア作業前にリストア対象のファイルシステム上に「restoresymtable」が存在していないことを確認して下さい。存在する場合は、リストア作業前に削除して下さい。 ・増分バックアップを使用してリストアする際に作成されるファイル「restoresymtable」は一連のリストア作業が全て終了するまでは削除しないで下さい。 ・リストア時に作成されるファイル「restoresymtable」はシステムのリストア終了後に削除して下さい。

●完全バックアップ(バックアップレベル0)

- ・毎回ファイルシステム全体のバックアップを取ります。

【コマンド例】

```
#ufsdump 0cuf /dev/rmt/0n /dev/rdisk/c0t0d0s0
```

↑
バックアップレベル

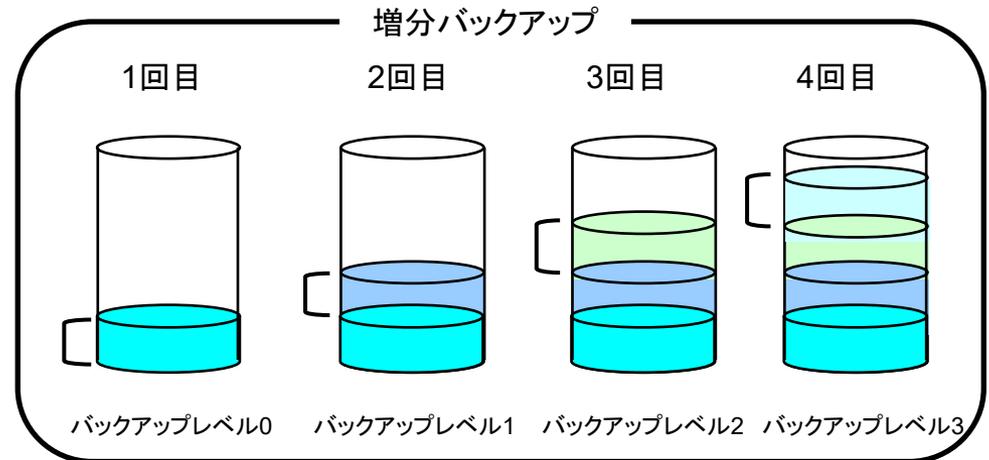
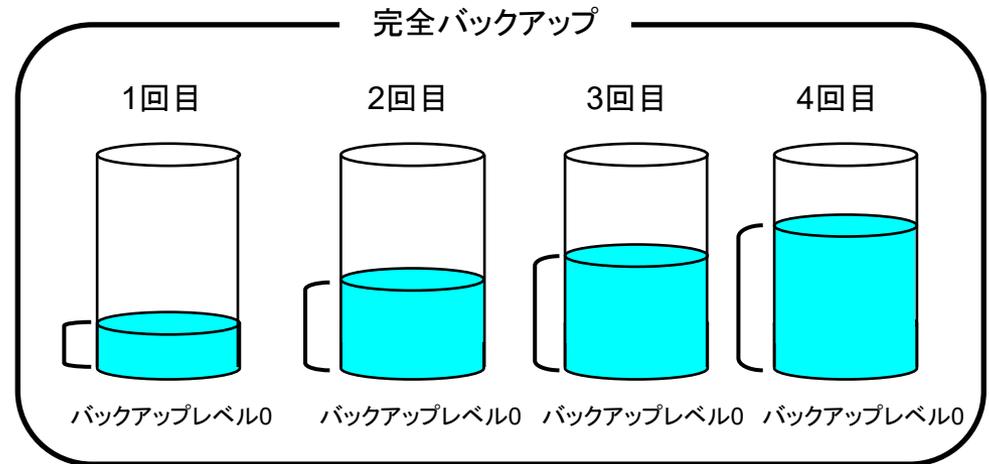
●増分バックアップ(バックアップレベル1~9)

- ・最後の低位バックアップ以降に変更があったファイルの増分バックアップを取ります。
- ・最初の1回は完全バックアップを取る必要があります。

【コマンド例】

```
#ufsdump 1cuf /dev/rmt/0n /dev/rdisk/c0t0d0s0
```

↑
バックアップレベル



コマンドの使い方と意味(1)

ufsdumpコマンド

形式

/usr/sbin/ufsdump [オプション] [引数] バックアップファイル

主なオプション	説明
0-9	バックアップレベル。レベル 0 は バックアップファイル で指定したファイルシステム全体の完全バックアップ用です。レベル 1 ~ 9 は、最後の下位バックアップ以降に変更があったファイルの増分バックアップ用です。
c	カートリッジ。カートリッジテープにバックアップを作成します。
f [dump-file]	ダンプファイル。デフォルトデバイスではなく dump-file で指定したコピー先にファイルを書き込みます。
u	ダンプレコードのアップデート。ファイルシステムの完全バックアップをとる場合には、ファイル /etc/dumpdates ファイルにエントリを追加します。このエントリは、ファイルシステムのディスクスライスのデバイス名、バックアップレベル(0 ~ 9)、日付を示します。

バックアップデバイス名

バックアップに使用するテープまたはフロッピーディスクドライブに論理デバイス名を与えて指定します。

種類	形式	説明
テープデバイス	/dev/rmt/ncn	n -- ドライブ番号(0~n) c -- 圧縮オプション。バックアップデータを圧縮します。 n -- 巻き戻しなし。省略時は巻き戻します。
フロッピーディスク	/vol/dev/rdiskette0/unlabeled	※フロッピーディスクを使用する場合は、予め「fdformat -v -U」コマンドでフォーマットをする必要があります。

コマンドの使い方と意味(2)

ufsrestoreコマンド

形式

/usr/sbin/ufsrestore [オプション] [引数] [ファイルネーム]

主なオプション	説明
r	媒体の内容全体を現在の作業ディレクトリ（ファイルシステムの最上位レベル）に復元。 ファイルシステムを完全に復元するには、このオプションを使用して完全（レベル 0）ダンプを復元してから、各増分ダンプを復元します。
v	詳細表示。各ファイルが復元されるたびに、その名前と i ノード番号が表示されます。
f [file-name]	ファイルネーム。ファイルは、デフォルトのデバイスファイル /dev/rmt/0m ではなく backup-file で指定したソースから読み取られる取られます。

使用例

ufsdumpとufsrestoreの使用例を下記に示します。

バックアップ

```
# ufsdump 0cuf /dev/rmt/0n /dev/rdsk/c0t0d0s0
.
.
# ufsdump 0cuf /dev/rmt/0n /dev/rdsk/c0t0d0s1
```

テープデバイス(/dev/rmt/0n)にスライスc0t0d0s0のバックアップを取ります。

注意: 例のようにテープデバイスの最後に「n」を付けないとc0t0d0s0のバックアップ取得後にテープが巻き戻され、c0t0d0s1のデータが上書きされてしまいます。

リストア

```
# ufsresotre rvf /dev/rmt/0n
```

カレントディレクトリにテープデバイス(/dev/rmt/0n)のデータを復元します。

<参考>バックアップのオペレーション

バックアップの流れを以下に示します。

<pre># shutdown -y -g0 -i0 ok boot -s</pre>	←システムをシングルユーザモードで起動します。
<pre># mountall -l</pre>	←ディスクをマウントします。
<pre># mt status DAT-72 tape drive: sense key(0x0)= No Additional Sense residual= 0 retries= 0 file no= 0 block no= 0</pre>	←テープの状態を確認します。
<pre># ufsdump /dev/rmt/0n /dev/rdisk/c0t0d0s0</pre>	←バックアップを開始します。 ※必要なスライスの分だけバックアップを実行します。この場合スライスc0t0d0s0の完全バックアップを取ります。
<pre># mt rewind</pre>	←テープを巻き戻します。
<pre># ufsrestore ivf /dev/rmt/0n ufsrestore> ls : ufsrestore> quit</pre>	←バックアップしたデータの確認を行います。 ※バックアップを取ったスライスの分だけ確認します。 ←バックアップデータが表示されます。
<pre># mt rewind</pre>	←テープを巻き戻します。
<pre># mt offline</pre>	←テープを取り出します。
<pre># shutdown -y -g0 -i6</pre>	←システムを再起動します。

<参考> リストアのオペレーション

リストアの流れを以下に示します。

<code>ok boot cdrom -s</code>	←システムをシングルユーザモードで起動します。
<code>SINGLE USER MODE</code>	
<code># format</code>	←formatコマンドでシステムバックアップ時のパーティション情報を作成します。
<code># newfs /dev/rdisk/c0t0d0s0</code>	←作成したパーティションにファイルシステムを作成します。
<code># fsck /dev/rdisk/c0t0d0s0</code>	←ファイルシステムのチェックを行います。
<code># mkdir /mnt/root</code>	←マウントポイントを作成します。
<code># mount -F ufs /dev/dsk/c0t0d0s0 /mnt/root</code>	←ファイルシステムをマウントします。
<code># cd /mnt/root</code>	←マウントポイントに移動してufsrestoreを実行します。
<code># cd</code>	「/dev/rmt/0n」は読み込み後にテープの巻き戻しを行いません。
<code># ufsrestore rf /dev/rmt/0n</code>	←作成されたリストア情報のファイルを削除します。
<code># rm restoresysmtable</code>	←ファイルシステムをアンマウントします。
<code># umount /mnt/root</code>	
<code># cd /</code>	
<code># installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk /dev/rdisk/c0t0d0s0</code>	↑ブートブロックをインストールします。
<code># shutdown -y -g0 -i6</code>	←システムを再起動します。

- 通常バックアップはテープデバイスに採取することが多く、基本的なテープデバイスの操作方法について理解する必要があります。

- テープデバイスの論理デバイス名

/dev/rmt/n

- mtコマンド

mtコマンドによってテープデバイス进行操作します。
下記にmtコマンドの形式、使用例を示します。

- 形式

mt [-f テープデバイス] コマンド カウント数

主なコマンド	意味	使用例
status	テープの状態を確認します。	mt -f /dev/rmt/0n status
rewind	テープを巻き戻します。	mt -f /dev/rmt/0n rewind
offline	テープを取り出します。	mt -f /dev/rmt/0n offline
fsf	カウントで指定された数のEOF分だけ進みます	mt -f /dev/rmt/0n fsf 2
bsf	カウントで指定された数のEOF分だけ後退します。	mt -f /dev/rmt/0n bsf 2
asf	カウントファイルの番号への位置へ移動します。	mt -f /dev/rmt/0n asf 2

※ EOF: End Of File。ファイルデータの最後につけられるマーク。

詳細は、設計・運用手順書
「6. バックアップ／リストア」
を参照してください。

7. 情報収集

- 構築作業においては定期的にシステム情報を採取することで、高品質なシステム構築を実現する

情報採取の必要性

- システムの正常動作の確認
- トラブル発生時の原因究明(サポートへの提示)
- 各種の設定情報の確認
- 作業エビデンス(証拠)として

各種情報収集ツールの知識(OS付属・富士通提供ツールなど)

システム構成の事前把握

正常時とトラブル時の各種情報の比較検討方法



サポートへの情報提供時にも有効となる
定期メンテナンスや**CE**作業の時間短縮にも有効

■ システム情報の個別収集

情報	確認方法および情報採取方法
システム情報	
OSのバージョン	unameコマンド
ハードウェアのレビジョン	/etc/releaseファイル
システム構成情報	prttdiagコマンド、fjprttdiagコマンド
ディスク構成情報	/etc/vfstabファイル、format、prtvtocコマンド(パーティション設定/確認)
パッチ(障害修正)情報	showrevコマンド
アプリケーション情報	pkginfoコマンド
クラッシュダンプ	dumpadmコマンド、/etc/dumpadm.confファイル
ログ情報	
起動メッセージ	dmesgコマンド
各種システムログ	/varディレクトリ(次項参照)
ネットワーク情報	
ネットワーク定義	/etc/hosts、networks、netmasks、hostname.<インターフェース名>ファイル
ネットワークインターフェース	ifconfigコマンド
ネットワーク稼働状況	netstatコマンド(インターフェース名)、nfsstatコマンド(NFSサービス)
ネットワークトレース(TCP/IP)	snoopコマンド

■ 代表的なシステムログ (/varディレクトリ)

ログファイル	説明	形式
/var/adm/messages (※)	システム起動時のメッセージ、コンソールに出力されるエラーメッセージ等が保存される。最新の起動メッセージがdmesgコマンドで表示可能	ASCII
/var/adm/sulog	su(スイッチユーザ)コマンドの履歴が保存される	ASCII
/var/adm/utmpx	システムにログイン中のユーザ情報が保存される。whoコマンドで利用される	Binary
/var/adm/wtmpx	全てのログイン、ログアウトの情報が保存される。lastコマンドで利用される	Binary
/var/adm/lastlog	ログインした各ユーザの最後のログイン時刻が保存される	Binary
/var/adm/vold.log	voldデーモン(リムーバブルメディアの管理プロセス)のメッセージが保存される	ASCII
/var/log/syslog (※)	メールシステムが起動している場合にsendmailデーモン(メール管理プロセス)のメッセージが保存される	ASCII
/var/cron/log	cronデーモン(簡易ジョブスケジューラ)のメッセージが保存される	ASCII
/var/lp/logs/lpsched (※)	スプール(プリンタ)関連のlpschedデーモンのメッセージが保存される	ASCII
/var/saf/_log	SAF機能の中でsacデーモンのメッセージが保存される	ASCII
/var/saf/ポートモニタタグ名/log	sacの子プロセスであるttymonデーモン、listenデーモンのメッセージが保存される	ASCII

※ ログローテーションによりオリジナルファイルが一定量に達した場合に別名([ファイル名].0、[ファイル名].1、...)で保存されるログ。

システム構築時におけるトラブル発生時も、上記の各種ログを確認し原因究明に役立てます。

■ fjsnap 機能概要

ハード／ソフト構成、環境設定、ログ・動作状態に関するファイルおよび各種コマンド実行結果を採取してファイルに出力します(約**200**種類、約**2000**ファイル)

● 実行方法

- 1) **fjsnap**コマンドを実行
- 2) クラッシュダンプ採取後のリブート時に自動採取

対象機種	採取コマンド	採取時間	採取サイズ	採取フォルダ	備考
SPARC Enterprise 全機種	fjsnap -a	約5分～60分	約5～100MB	/var/crash/'uname -n`	・ESFの適用必須 ・作業領域として/var/tmpを使用

※ システム環境により採取時間、サイズが異なります。

● 実行の必要性

- ・運用前の構成情報とトラブル時の構成情報を比較することで、故障箇所の迅速な特定に繋げる
- ・システム構築作業において適宜採取することで、構築作業の漏れ/ミス等のチェック用

コマンド詳細はESFのマニュアル参照
「ESFユーザーズガイド システム情報採取ツール編」

URL <http://software.fujitsu.com/jp/manual/manualindex/p11000049.html>

■ システムログ (var/adm/messages)

- エラーを記録するデーモンである syslogd は、さまざまなシステムの警告やエラーをメッセージファイルに自動的に記録します。デフォルトでは、/var/adm ディレクトリに格納されます。
- messagesファイルは/etc/syslog.conf ファイルによって管理されています。

/etc/syslog.conf ファイル

```
user.err      /dev/sysmsg
user.err      /var/adm/messages
user.alert    `root, operator`
user.emerg    *
```

● syslog ファシリティ・レベル(プライオリティ)一覧

ファシリティ(送信元)	説明
kern	カーネル
user	ユーザプロセス
mail	メールシステム
daemon	すべてのデーモン
auth	認証
lp	スプールシステム
*	全てのファシリティ

レベル(優先順位)	説明
emerg	システムの緊急事態
alert	すぐに修正が必要なエラー
crit	ハードウェア障害などの致命的なエラー
err	通常のエラー
warning	警告メッセージ
notice	エラーではないが対処が必要な状態
info	情報メッセージ
none	メッセージを送信しない
debug	プログラムをデバッグする時に使用されるメッセージ
*	noneを除く全てのレベル

ログローテーション

■ ログファイルの肥大化によるリソース枯渇への対処 (logadm)

- ログファイルは日々の運用によって増加していくため、増え続けるとディスクリソースの枯渇が発生します。
その対処として、ログファイルのローテーションを行います。
- logadm.confにローテートする対象ファイル、ファイル数、契機サイズなどの設定を記載します。

(/var/adm/messages,/var/log/syslogなどはデフォルトでローテーション対象となっています。)
/etc/logadm.conf

```

--(省略)--
/var/log/syslog -C 8 -P 'Mon May 18 18:10:00 2009' -a 'kill -HUP `cat /var/run/syslog.pid`'
/var/adm/messages -C 4 -P 'Fri May 22 18:10:00 2009' -a 'kill -HUP `cat /var/run/syslog.pid`'
/var/cron/log -c -s 512k -t /var/cron/olog
/var/lp/logs/lpsched -C 2 -N -t '$file.$N'
/var/fm/fmd/errlog -M '/usr/sbin/fmadm -q rotate errlog && mv /var/fm/fmd/errlog.0- $nfile' -N -s 2m
/var/fm/fmd/fitlog -A 6m -M '/usr/sbin/fmadm -q rotate fitlog && mv /var/fm/fmd/fitlog.0- $nfile' -N -s 10m
smf_logs -C 8 -s 1m /var/svc/log/*.log
--(省略)--

```

crontabに定期的にlogadmを実行するようにスケジューリングされています。

設定例) date.txtファイルが32byteを超えたいたら、date.txt.0 (1,2,3)という名前でローテーションさせる。

```
/tmp/date.txt -C 4 -s 32b -t '$file.$n'
```

↑ 対象ファイル
↑ ローテーション数
↑ 契機サイズ
↑ ファイル名



クラッシュダンプ生成の流れ

- システムクラッシュが発生すると自動的にダンプを生成しますが、ハングアップした場合は手動でダンプ採取が可能です

- システムクラッシュが発生

コンソールにpanic
メッセージ
表示

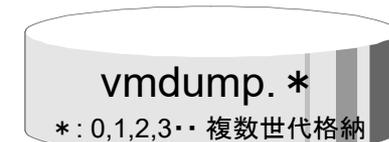


ダンプ対象メモリ
●カーネルメモリページ
●全てのメモリページ



- システム再起動後に
ダンプのセーブ処理

/var/crash/^uname -n` (ノード名)



※fjsnapも自動実行されます

Solaris 9/10から
「Fast Crash Dump」
機能をサポート

- システムハングが発生

XSCF (ILOM,ALOM)
から専用コマンドでダ
ンプ生成指示を出しま
す。



コンソール接続してROM強制割り込み後に採取

(BREAK信号送出)

send break

ok> sync

- Fast Crash DumpはSolaris 10 9/10の新機能で、従来よりダンプ採取が高速、かつ圧縮形式で採取されます。

・従来のダンプ形式
(Solaris 10 10/09以前)

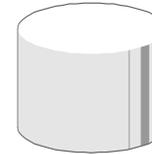


ダンプファイル名

{ unix.*
vmcore.*

*: 0,1,2,3... 複数世代格納

・新ダンプ形式
(Solaris 10 9/10)



ダンプファイル名

vmdump.*

*: 0,1,2,3... 複数世代格納

- 圧縮されないため、フルダンプの場合、メモリ容量と同等のディスク領域が必要である。
- 過去のダンプファイルの削除を忘れると、最新のダンプファイルが採取できないことがある。

- 採取時から圧縮形式のため、ディスク容量の節約が可能 (※検証環境では約10分の1以上に圧縮)
- ダンプファイルはsavecoreコマンドで解凍可能 (解凍すると、unix.*、vmcore.*ファイルが生成)

Solaris 10 9/10ではデフォルトで圧縮形式で採取する設定になっています。非圧縮に変更するにはdumpadmコマンドを使用します。(詳細はマニュアル参照)

- ダンプの採取方法は適用する機種により差異があります

	トラブル発生時のOSの状況		採取時間
	システムハング時	クラッシュ(パニック)時	
SPARC Enterprise M3000/4000/5000/ 8000/9000	XSCFからコマンド実行 XSCF> reset -d <DID> panic	boot時に自動的に採取	数分～数十分 (メモリ搭載量により変化)
SPARC T3-1/T3-2/T3-4, SPARC Enterprise T5120/5220/5140/ 5240/5440	ILOMからコマンド実行 -> send_break_action=dumpcore		

<参考>ダンプ用デバイスの準備

■ ディスク容量の確保が必要です

ダンプファイルは、swapデバイスからダンプデバイスに保存されるため、以下の容量確保が必要。^{*1}

***1 デフォルト:**
/var/crash/'uname -n'

	最小	最大
/var,swapの空き容量	512M byte	メモリ搭載容量
取得情報	Kernel部分のみ	全ダンプ情報

最大容量を
推奨(※)

搭載する物理メモリサイズとダンプ採取範囲(カーネル or フルダンプ)によって、swapサイズと/varサイズの見積りに影響します。

詳細は、設計・運用手順書
「7. 情報収集」
を参照してください。

8. セキュリティ設定

■ Solaris 10で利用できるセキュリティツール/ソフトウェア

- Kerberos
セキュリティ保護されたネットワーク経由で使用するセキュリティ認証プロトコル
- Solaris Security Toolkit (旧名称: JASS ツールキット)
システムに一括でセキュリティ変更を行うためのツール
- Solaris IP Filter
Solaris OS 用のファイアウォールソフトウェア
- Solaris Secure Shell (SSH)
すべてのネットワークトラフィックを暗号化するソフトウェア
- RBAC (Role-Based Access Control)
役割を実ユーザに割り当てることでroot権限でしか実行することのできないコマンド群を一般ユーザでも実行できるようにします。
- TCP_wrappers
外部からの接続要求をTCP/IP毎にアクセス制御を行います。

<参考> TCP_wrappersによるアクセス制限

■ TCP_wrappersとは

クライアントからのアクセスをTCP/IPサービス毎に制限するためのサービスです。

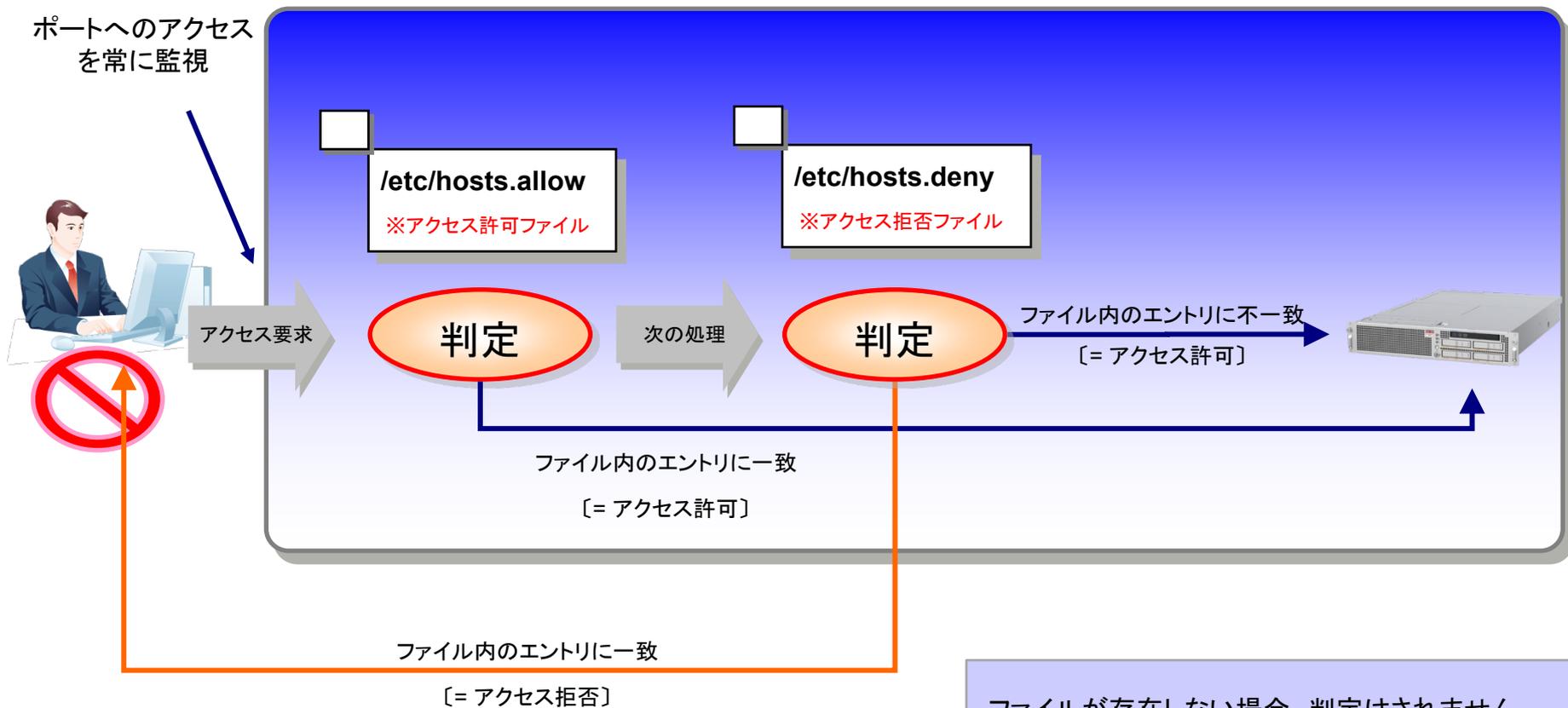
● TCP_wrappersによってアクセス制限できるサービス

サービス名	役割
inetd	サービスが使用するポートを監視・デーモンを起動するサービスです。
sshd	暗号化通信を行うサービスです。
rpcbind	RPCプログラム番号とポート番号をマッピングするサービスです。
sendmail	メールの送受信・転送を行うサービスです。

● TCP_wrappers は「SUNWtcpd」というパッケージ名で標準インストールされています。

■ TCP_wrappers のアクセス制御の仕組み

TCP_wrappersによるアクセス監視



<参考> TCP_wrappers設定 1/2

《TCP_wrappersの設定》

①アクセス許可ファイルの作成

```
# touch /etc/hosts.allow
```

②アクセス拒否ファイルの作成

```
# touch /etc/hosts.deny
```

③TCP_wrappersの設定確認

```
# inetadm -l svc:/network/telnet:default
SCOPE  NAME=VALUE
  name="telnet"
  endpoint_type="stream"
  proto="tcp6"
  isrpc=FALSE
  wait=FALSE
  exec="/usr/sbin/in.telnetd"
  user="root"
default bind_addr=""
default bind_fail_max=-1
default bind_fail_interval=-1
default max_con_rate=-1
default max_copies=-1
default con_rate_offline=-1
default failrate_cnt=40
default failrate_interval=60
default inherit_env=TRUE
default tcp_trace=FALSE
default tcp_wrappers=FALSE
default connection_backlog=10
```

← デフォルトでは、tcp_wrapperは無効 (FALSE)

④TCP_wrappersの設定を有効にします。

```
# inetadm -m telnet tcp_wrappers=TRUE
```

⑤ルールの設定確認

```
# /usr/sfw/sbin/tcpdchk -v
Using network configuration file: /etc/inet/inetd.conf

>>> Rule /etc/hosts.allow line 1:
daemons: ALL
clients: 10.20.112.
access: granted

>>> Rule /etc/hosts.deny line 1:
daemons: in.telnetd
clients: 192.168.2.5/255.255.255.0
access: denied
```

← 許可ファイル名と下記を設定する

← 対象とするサービスのデーモン名

← 許可するアクセス元IPアドレス

← granted は「許可」の意味

← 拒否ファイル名と下記を設定する

← 拒否するデーモン名

← 拒否するアクセス元IPアドレス

← denied は「拒否」の意味

詳細は、設計・運用手順書
「8. セキュリティ設定」
を参照してください。

参考情報

■ Solaris技術者向けにSolaris環境構築上必要となる、OSコマンドの使い方を逆引き集として公開

<https://www.fujitsu.com/jp/products/computing/servers/unix/sparc/technical/command-reference/index.html>

● 良く使うオプションを中心に使用例を掲載

The image shows two overlapping screenshots of the Fujitsu Solaris Command Reference website. The background screenshot displays the '逆引きコマンド一覧' (Reverse Index) page, which lists various Solaris commands and their descriptions. The foreground screenshot provides a detailed view of the 'boot' command, including its options and usage examples.

逆引きコマンド一覧

サーバ構築時の使用目的別にOracle Solaris/XSCFコマンドを分類した「逆引きコマンド集」。複数オプションが存在するコマンドは、主に使用するものを説明しています。

- XSCF操作 (522 KB)
- OBP操作
- システムの起動/停止
- システム起動環境の管理
- サービスの設定/管理
- ユーザ情報の設定/確認
- グループの設定/確認
- システム情報の設定/確認
- 環境変数の設定/確認
- カレントディレクトリ操作
- ファイルの権限/属性/種類
- ファイルの作成/削除
- ファイル内容の表示/編集
- 検索/比較/ソート
- ファイルのアーカイブ/圧縮/展開
- ネットワークの設定/表示
- ネットワークサービス操作
- バッチの適用/確認
- パッケージのインストール/確認
- プロセスの表示/操作
- デバイスやディスクの設定
- ファイルシステム関連の操作
- バックアップ/リストア
- 資源管理の設定/確認
- シェル関連
- コマンド定期実行/測定
- ヘルプ

boot [\[コマンド作成\]](#)

形式

boot オプション	説明
-s	シングルユーザーモードで起動します。
cdrom	CD、DVDから起動します。
net	ネットワークブートを行います。
-r	システムを再構成します。
-a	対話型でのブートを行います。

使用例

```
boot

システムをマルチユーザーモードで起動します。

[0] ok boot
Boot device: /pci@1917,4000/scsi@0/disk@0,0:a File and args:
.
```

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。

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。

