

# HITACHI

ソフトウェアマニュアル  
オペレーション

RPDP/S10V  
For Windows®

---

**S10V**

Programmable Controller

ソフトウェアマニュアル  
オペレーション

# RPDP/S10V For Windows<sup>®</sup>

---

**S10V**  
Programmable Controller

この製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認のうえ、必要な手続きをお取りください。  
なお、不明な場合は、弊社担当営業にお問い合わせください。

2003年 4月 (第1版) SVJ-3-133 (A) (廃版)  
2012年11月 (第2版) SVJ-3-133 (B) (廃版)  
2017年 8月 (第3版) SVJ-3-133 (C)

- このマニュアルの一部または全部を無断で転写したり複製したりすることは、固くお断りいたします。
- このマニュアルの内容を、改良のため予告なしに変更することがあります。

## 安全上のご注意

- システムの構築やプログラムの作成などは、このマニュアルの記載内容をよく読み、書かれている指示や注意を十分理解してから行ってください。誤操作により、システムが故障することがあります。
- このマニュアルは、必要なときすぐに参照できるよう、手近なところに保管してください。
- このマニュアルの記載内容について疑問点または不明点がございましたら、最寄りの弊社営業またはSEまでお知らせください。
- お客様の誤操作に起因する事故発生や損害については、弊社は責任を負いかねますのでご了承ください。
- 弊社提供ソフトウェアを改変して使用した場合に発生した事故や損害については、弊社は責任を負いかねますのでご了承ください。
- 弊社提供以外のソフトウェアを使用した場合の信頼性については、弊社は責任を負いかねますのでご了承ください。
- ファイルのバックアップ作業を日常業務に組み入れてください。ファイル装置の障害、ファイルアクセス中の停電、誤操作、その他何らかの原因によりファイルの内容を消失することがあります。このような事態に備え、計画的にファイルのバックアップを取っておってください。
- 弊社製品が故障や誤動作したりプログラムに欠陥があった場合でも、使用されるシステムの安全が十分に確保されるよう、保護・安全回路は外部に設け、人身事故や重大な災害に対する安全対策が十分確保できるようなシステム設計としてください。
- 非常停止回路、インターロック回路などはPLCの外部で構成してください。PLCの故障により、機械の破損や事故の恐れがあります。
- 運転中のプログラム変更、強制出力、RUN、STOPなどは十分安全を確認してから行ってください。誤操作により、機械の破損や事故の恐れがあります。
- このマニュアルでは、安全上の注意事項のランクを潜在危険の重大度によって、「危険」、「警告」、「注意」、「通知」と区分しています。

### 警告表示の定義



：この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こす危険の存在を示す。



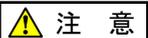
：この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こすおそれのある危険の存在を示す。



：この表示を無視して誤った取り扱いをすると、軽度の傷害または中程度の傷害を引き起こすおそれのある危険の存在を示す。



：この表示を無視して誤った取り扱いをすると、人身傷害とは関係のない損害を引き起こすおそれのある危険の存在を示す。

なお、 **注意**、 **通知** に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。どれも重要な内容を記載していますので必ず守ってください。

「重大な傷害」、「軽度の傷害または中程度の傷害」、「人身傷害とは関係のない損害」について、具体的な内容を以下に示します。

**重大な傷害**

失明、けが、やけど（高温、低温）、感電傷害、骨折、中毒などで、後遺症が残るものおよび治療のために入院、長期の通院を要するもの

**軽度の傷害または中程度の傷害**

治療のために入院や長期の通院を必要としないけが、やけど、感電傷害など

**人身傷害とは関係のない損害**

周囲の財物の損傷、弊社製品の故障や破損、データの損失など、人身傷害以外の損害

安全上の注意事項は、安全性を確保するための原則に基づいた、弊社製品における各種対策を補完する重要なものです。弊社製品やマニュアルに表示されている安全上の注意事項は、十分に検討されたものですが、それでも、予測を超えた事態が起こることが考えられます。操作するときは指示に従うだけでなく、常に自分自身でも注意するようにしてください。また、弊社製品の安全な運転および保守のために、各種規格、基準に従って安全施策を確立してください。

1.  このマニュアル内の警告表示

このマニュアルの中に書かれている警告とその記載箇所を、以下にまとめて示します。

1. 1 「通知」と表示されているもの

(第1編、第3章、1-42ページ)

通 知
RPDPを使用する際は、複数のユーザがログオンした状態で使用しないでください。 複数のユーザを使う場合は、必ず使用中のユーザをログオフしてから、別のユーザのログオンを実行してください。 複数ユーザがログオンした状態では、RPDPのコマンドが正しく実行されません。

このページは白紙です。

## はじめに

このマニュアルは、S10VのCPMS下で動作するリアルタイムプログラムをMicrosoft® Windows® 2000、Windows® XP、Windows® 7 (32bit) 、およびWindows® 10 (32bit) のもとで作成する方法について述べたものです。

- 下表に関連マニュアル（ソフトウェアマニュアル）を示します。

マニュアル番号	マニュアル名称
SVJ-3-201	ソフトウェアマニュアル CPMS概説&マクロ仕様

- 次の用語は、このマニュアルにおいて特殊な意味に用いますので注意してください。

用語	意味
開発系マシン	Microsoft® Windows® 2000、Windows® XP、Windows® 7 (32bit) 、およびWindows® 10 (32bit)
PCs	S10V

### 略称

RPDP : Realtime Program Developing Package for S10V

CPMS : Compact Process Monitor System

PCs : Programmable Controllersの略です。

S10 $\alpha$  およびS10miniシリーズ等のPLCの総称です。

PLC : Programmable Logic Controllerの略です。

プログラム内蔵方式でシーケンス制御を行う工業用電子装置のことです。

S10 $\alpha$  およびS10miniシリーズ等もPLCに該当します。

- このマニュアルは「第1編 概説」と「第2編 コマンドリファレンス」の2つの編から構成されます。

第1編は、S10V上で動作するリアルタイムプログラムの開発手順と開発に使用するコマンドの概要について説明しています。

第2編は、S10V上で動作するリアルタイムプログラムの開発に使用するコマンドのリファレンスです。コマンドごとの機能やオプション機能について説明しています。

付録は、S10V上で動作するリアルタイムプログラムを開発するうえでの注意事項やエラーメッセージ、さらにコマンド実行結果の表示フォーマットを示します。

<商標について>

- Microsoft®, Windows®は、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- SuperHは、ルネサス エレクトロニクス株式会社の登録商標です。

<記憶容量の計算値についての注意>

- $2^n$ 計算値の場合（メモリ容量・所要量、ファイル容量・所要量など）
  - 1KB（キロバイト）＝1,024バイトの計算値です。
  - 1MB（メガバイト）＝1,048,576バイトの計算値です。
  - 1GB（ギガバイト）＝1,073,741,824バイトの計算値です。
- $10^n$ 計算値の場合（ディスク容量など）
  - 1KB（キロバイト）＝1,000バイトの計算値です。
  - 1MB（メガバイト）＝1,000<sup>2</sup>バイトの計算値です。
  - 1GB（ギガバイト）＝1,000<sup>3</sup>バイトの計算値です。

# 目次

## 第1編 概説

第1章 概要 .....	1-2
1.1 RPDПについて .....	1-2
1.2 コマンド一覧 .....	1-4
第2章 プログラム開発手順 .....	1-7
2.1 全体フロー .....	1-7
2.2 サイト環境 .....	1-10
2.3 主メモリのエリア管理と領域分割 .....	1-11
2.4 タスクのエリア配置 .....	1-15
2.5 IRSUBのエリア配置 .....	1-16
2.6 プログラムのロードとタスクの生成 .....	1-17
2.7 間接リンク常駐サブプログラム .....	1-17
2.8 グローバル (GLB) .....	1-17
2.9 バリュ (VAL) .....	1-17
2.10 間接リンクグローバル .....	1-17
2.11 GLB、VAL、IRSUBプログラミングガイド .....	1-18
2.12 CPMS上のプログラム作成の制約条件 .....	1-27
2.13 名称管理について .....	1-33
第3章 インストールとRPDП実行環境 .....	1-35
3.1 インストール .....	1-35
3.2 RPDП実行環境 .....	1-36
3.3 RPDПのコマンド格納ディレクトリ .....	1-39
3.3.1 環境変数設定例 .....	1-39
3.3.2 コマンドプロンプトからの環境変数設定方法 .....	1-40
3.4 RPDП使用者アカウントの登録 .....	1-41
3.4.1 新規アカウントの登録 .....	1-41
3.4.2 既存のアカウントの所属するグループにRPDPS10Usersグループを追加 .....	1-42
第4章 コンパイラ .....	1-43
4.1 Cコンパイラオプション詳細 .....	1-43
4.2 コンパイル時の注意点 .....	1-45
4.2.1 shcコマンドを使用してコンパイルする場合 .....	1-45
4.3 mcc68kとshcの相違点 .....	1-47

4. 3. 1	コマンド行オプション	1-47
4. 3. 2	言語仕様相違点	1-51
4. 3. 3	関数の呼び出し規約	1-51
第5章 プログラミングコマンド		1-52
5. 1	プログラミングコマンドについて	1-52
第6章 システムジェネレーション		1-53
6. 1	概要	1-53
6. 1. 1	システムジェネレーションの目的	1-53
6. 1. 2	システムジェネレーションの特長	1-53
6. 1. 3	複数のPCsにおけるサイト管理方式	1-55
6. 2	システムジェネレーションコマンドとシステム生成の流れ	1-56
6. 3	PCsシステム定義情報	1-61
6. 3. 1	ユーザ設定定義情報	1-61
6. 3. 2	PCsシステム定義情報の内容	1-64
6. 3. 3	ネットワーク定義情報の内容	1-65
6. 3. 4	サイト定義情報	1-66
6. 4	PCsシステム定義情報の表示	1-67
6. 5	PCsシステム環境の複製	1-68
6. 5. 1	複製の単位	1-68
6. 5. 2	複製の範囲	1-69
6. 6	PCsシステム環境の削除	1-70
第7章 アロケータ		1-71
7. 1	分割領域の確保と削除	1-71
7. 1. 1	領域の分割の必要性	1-71
7. 1. 2	分割領域の確保	1-72
7. 1. 3	分割領域の削除	1-75
7. 1. 4	GLB、VALの名前の付け方	1-75
7. 2	バリュ (VAL) の登録、削除	1-76
第8章 ローダ		1-77
8. 1	リンク・ロードとは	1-77
8. 2	ローダの動作環境	1-78
8. 3	ライブラリのサーチパス	1-81
8. 4	リンク・ロード時の注意事項	1-81

第9章 ビルダ	1-82
9. 1 タスクの登録と削除	1-82
9. 1. 1 タスクとは	1-82
9. 1. 2 タスクの登録	1-82
9. 1. 3 タスクの削除	1-83
9. 2 常駐サブプログラムの登録と削除	1-84
9. 2. 1 間接リンクサブプログラム (IRSUB) とは	1-84
9. 2. 2 間接リンクサブプログラム (IRSUB) の登録	1-84
9. 2. 3 間接リンクサブプログラム (IRSUB) の削除	1-85
9. 3 組み込みサブルーチンの登録と削除	1-86
9. 3. 1 組み込みサブルーチンとは	1-86
9. 3. 2 組み込みサブルーチンの登録	1-86
9. 3. 3 組み込みサブルーチンの削除	1-87
第10章 マップ	1-88
10. 1 アロケータ管理テーブル情報表示の目的	1-88
10. 2 svmapコマンドのオプション指定と表示情報	1-89
10. 2. 1 マップ情報の出力対象	1-89
10. 2. 2 マップ情報の出力内容	1-89
10. 2. 3 マップ情報の出力形式	1-89
10. 3 svadmコマンドの論理アドレス指定と表示情報	1-91
第11章 立ち上げ	1-92
11. 1 概要	1-92
11. 2 CMUの状態遷移	1-93
11. 2. 1 立ち上げ操作	1-93
11. 2. 2 CMU制御操作	1-94
第12章 svdebug (オンラインデバッガ) とデバッグ支援コマンド	1-95
12. 1 概要	1-95
12. 2 PCs状態とサブコマンド実行可否	1-96
12. 3 基本的な機能	1-97
12. 4 その他の機能	1-101
12. 5 デバッグ支援コマンド	1-102
12. 5. 1 svelogコマンド	1-102
12. 5. 2 svdhpコマンド	1-103
12. 5. 3 svcpunowコマンド	1-104
12. 5. 4 svtimexコマンド	1-105

## 第2編 コマンドリファレンス

第1章 システムジェネレーション .....	2-2
svgen .....	2-2
svconf .....	2-3
svshconf .....	2-4
svsitecp .....	2-5
svsitedel .....	2-6
第2章 アロケータ .....	2-7
svdfa .....	2-7
svdla .....	2-9
svdfs .....	2-10
svdls .....	2-12
svdfv .....	2-13
svdlv .....	2-14
第3章 ローダ .....	2-15
svload .....	2-15
svdload .....	2-27
svcomp .....	2-28
第4章 ビルダ .....	2-32
svctask .....	2-32
svdtask .....	2-34
svbuild .....	2-35
svdbuild .....	2-37
svirglb .....	2-39
第5章 マップ .....	2-41
svmap .....	2-41
svadm .....	2-44
svsitectl .....	2-47
第6章 立ち上げ .....	2-48
svrpl .....	2-48
svcpuctl .....	2-49

第7章 svdebug (オンラインデバッガ) とデバッグ支援コマンド .....	2-50
svdebug .....	2-50
qu .....	2-54
ab .....	2-55
re .....	2-56
ta .....	2-57
su .....	2-60
rs .....	2-61
tm .....	2-62
ct .....	2-64
sht .....	2-65
md .....	2-66
sd .....	2-70
bs .....	2-73
bg .....	2-75
mcp .....	2-77
mmv .....	2-79
mf .....	2-81
el .....	2-83
ss .....	2-83
st .....	2-84
gt .....	2-85
br .....	2-86
rb .....	2-91
rd .....	2-92
rr .....	2-95
go .....	2-96
as .....	2-97
ac .....	2-99
ld .....	2-100
sv .....	2-108
cm .....	2-110
dr .....	2-111
ds .....	2-111
svdhp .....	2-112
svadm .....	2-112
si .....	2-113
sp .....	2-114

ps .....	2-115
pe .....	2-115
ver .....	2-116
help .....	2-117
q .....	2-118
! .....	2-118
svelog .....	2-119
svdhp .....	2-121
svcpunow .....	2-123
svtimex .....	2-124

## 付録

付録A プログラムで使用できる名称 .....	A-2
付録B ライブラリ .....	A-5
付録C サイト管理ファイル .....	A-8
付録D エラーメッセージ .....	A-12
付録E RPDP使用上の注意事項 .....	A-44
付録F svmapの表示フォーマット .....	A-46
付録G svdebug (オンラインデバッガ) md、sdの表示フォーマット .....	A-61
付録H ライブラリの使用するスタックサイズ一覧 .....	A-65

## 図目次

図 1-1	ツールの適用システム構成	1-2
図 1-2	プログラム開発手順全体フロー (サイト構築からプログラム開発までの流れ)	1-8
図 1-3	PCsのサイトディレクトリ構成	1-10
図 1-4	CPMSが管理する論理空間	1-11
図 1-5	PCsの物理メモリマップ	1-13
図 1-6	論理空間内のタスク配置	1-15
図 1-7	論理空間内のタスク配置 (マルチタスク)	1-15
図 1-8	論理空間内のIRSUB配置	1-16
図 1-9	論理空間内のIRSUB配置 (マルチエントリ)	1-16
図 1-10	書き込みの可否	1-28
図 1-11	データサイズ比較	1-31
図 1-12	データ配置順を考慮した構造体宣言例	1-32
図 1-13	空きエリアを明示的に宣言した例	1-32
図 1-14	構造体サイズを考慮した宣言例	1-32
図 1-15	PCsシステム全体構成	1-54
図 1-16	サイトディレクトリ構成	1-55
図 1-17	PCsシステム新規構築手順	1-57
図 1-18	PCsシステム改造手順	1-57
図 1-19	同一開発系マシン内のサイトコピー手順	1-58
図 1-20	開発系マシンのOS変更に伴うサイト移行手順	1-59
図 1-21	RPDPに関するディレクトリ構成とサイト構築環境定義ファイル構成	1-64
図 1-22	サイト複製機能でコピーされる範囲	1-69
図 1-23	分割領域のレイアウト	1-72
図 1-24	ロードモジュール、バックアップファイルの作成	1-77
図 1-25	ロードモジュールの構成	1-78
図 1-26	ローディング処理	1-79
図 2-1	関数呼び出しの関係とスタック使用量	2-20
図 2-2	svcomp (プログラム、サブプログラム) のフォーマット	2-30
図 2-3	svcomp (GLB) のフォーマット	2-31
図 2-4	メモリアクセス範囲	2-69
図 2-5	ダイナミック表示のオペレーション	2-69
図 2-6	メモリアクセス範囲	2-78
図 2-7	ADT設定範囲	2-98

## 表目次

表 1-1	RPDPの提供コマンド一覧	1-4
表 1-2	各論理空間の用途	1-11
表 1-3	各論理空間のアドレスとサイズ	1-12
表 1-4	GLB、VALの名称の付け方	1-18
表 1-5	GLBおよびVALの使い方	1-18
表 1-6	IRSUBの使い方	1-25
表 1-7	RPDP実行環境の環境変数一覧 (SHC Ver7)	1-37
表 1-8	RPDP実行環境の環境変数一覧 (SHC Ver9)	1-38
表 1-9	設定環境変数	1-44
表 1-10	浮動小数点数の扱い制御オプション	1-45
表 1-11	浮動小数点数の扱いと対応する標準ライブラリ	1-45
表 1-12	mcc68kとshcのコマンド行オプション比較	1-47
表 1-13	shcのオプション一覧	1-49
表 1-14	言語仕様の比較	1-51
表 1-15	コマンドの種類と機能	1-56
表 1-16	サイト定義情報の内容	1-61
表 1-17	ユーザ定義サイト構成定義情報設定項目	1-63
表 1-18	メモリに関するサイト構築定義情報	1-66
表 1-19	分割領域の用途と配置するGAREAの対応	1-72
表 1-20	ロードモジュールの条件	1-78
表 1-21	外部参照の組み合わせ	1-80
表 1-22	出力内容と指定可能出力形式の組み合わせ	1-90
表 2-1	svdfaのオプションの組み合わせ	2-8
表 2-2	stypeに指定する値とアライン数の関係	2-11
表 2-3	svdfsのオプションの組み合わせ	2-11
表 2-4	スタックサイズの計算例	2-20
表 2-5	オプションの組み合わせと出力内容	2-43
表 2-6	svdebug機能一覧	2-52
表 2-7	シグナル受信時の動作	2-53
表 2-8	タスクの状態	2-58
表 2-9	ステータスビットの構成	2-58
表 2-10	id、t、cyctの説明	2-63
表 2-11	指定できる値とオプションの組み合わせ	2-68
表 2-12	オプションの組み合わせによる表示フォーマット	2-68
表 2-13	リソースの管理状態	2-104
表 A-1	ライブラリの指定条件	A-5

表A-2	シグナルの扱い .....	A-22
表A-3	エラーメッセージ .....	A-22
表A-4	リアルタイムリソースの管理状態 .....	A-48

このページは白紙です。

## 第1編 概説

## 第1章 概要

### 1. 1 RPDPIについて

リアルタイムプログラム開発パッケージ (S10V RPDP) とは、S10VのリアルタイムOS (CPMS) 上で動作するプログラムを開発するツールです。このツールは、Windows® 2000、Windows® XP、Windows® 7 (32bit) 、またはWindows® 10 (32bit) 搭載マシン (以降、開発系マシンと表記します) 上で動作します。以下にこのツールを使用するシステム構成を示します。

PP型式	開発系マシン前提OS
S-7895-10P	Microsoft® Windows® 2000およびWindows® XP
S-7895-63P	Microsoft® Windows® 7 (32bit) およびWindows® 10 (32bit)

- S10V RPDP: Realtime Program Developing Package for S10V
- CPMS: Compact Process Monitor System

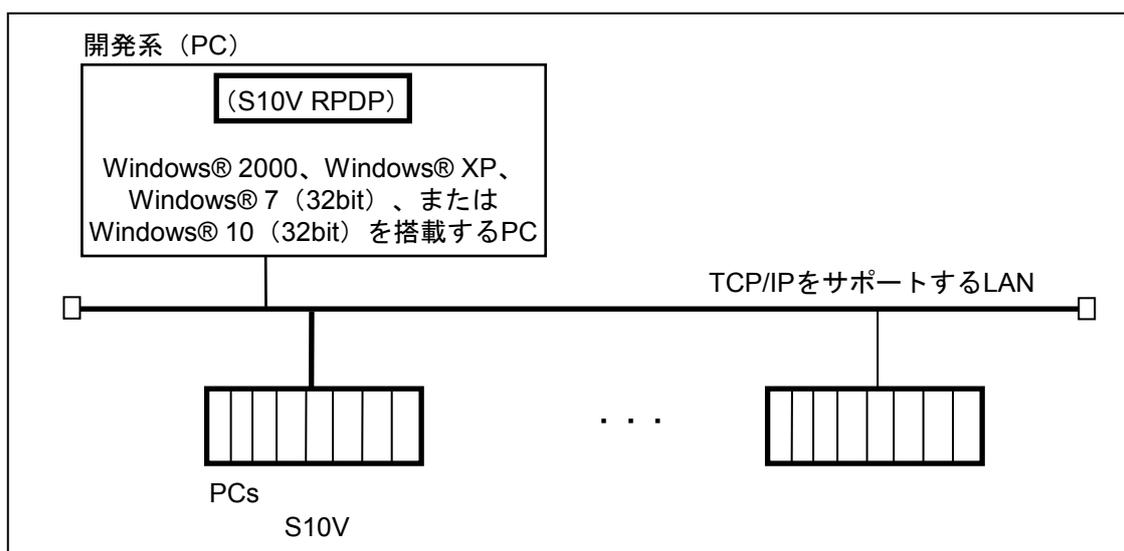


図 1 - 1 ツールの適用システム構成

CPMS上で動作するリアルタイムプログラムの開発は、専用の開発システムS10V RPDPを用いて行います。このマニュアルではS10V RPDPをRPDPと呼びます。RPDPを用いてプログラムを開発することにより、CPMS上で動作するプログラムは、リアルタイム処理を高速で行えるようにするための属性や機能を使用できます。

以下にRPDPのプログラム開発手順とサポート機能について示します。

サイト構築 (システムジェネレーション機能)	PCsごとにサイト名称を割り当てて管理します。システムジェネレーション機能を使用してサイト名称に対応したサイト環境を構築します。
↓	
プログラミング (コンパイル、ライブラリ生成)	クロスコンパイラ (*) やライブラリアン (*) を使用して開発系マシン上でPCsで動作するプログラムオブジェクトやライブラリを生成します。
↓	
メモリの割り付け (アロケータ)	RPDPのアロケータを用いてPCsメモリ上のタスク、サブルーチン (IRSUB)、グローバルデータ (GLB) を格納する領域を割り付けます。
↓	
ローディングと登録、登録内容表示 (ローダ・ビルダ・管理ツール (マップ))	ローダを使用してアロケータにより確保した領域にタスク、サブルーチン (IRSUB)、グローバルデータ (GLB) を格納します (ローディング)。ローディングしたものをタスクやサブルーチン (IRSUB) として登録するのがビルダです。登録内容はマップ機能による一覧表示により確認できます。
↓	
PCsへのダウンロード (立ち上げ)	立ち上げ機能によりサイト環境下のバックアップファイルをPCsのメモリへダウンロードして立ち上げます。
↓	
デバッグ (デバッガ)	タスクやIRSUBの動作を確認するためにデバッガを使用します。デバッガはタスクの動作を制御する機能を持っています。また、バックアップファイルからPCsメモリへのタスク、IRSUB、グローバルを個別にローディングする機能も持っています。

(\*) 開発系マシンの前提OSがMicrosoft® Windows® 2000およびWindows® XPの場合には、クロスコンパイラ・ライブラリアンは、「日立マイクロコンピュータ開発環境システムSuperH RISC engine C/C++コンパイラパッケージVer.7」を使用します。

開発系マシンの前提OSがMicrosoft® Windows® 7 (32bit) およびWindows® 10 (32bit) の場合には、クロスコンパイラ・ライブラリアンは、「日立マイクロコンピュータ開発環境システムSuperH RISC engine C/C++コンパイラパッケージVer.9」を使用します。

1.2 コマンド一覧

RPDPが提供するコマンド一覧を表1-1に示します。

表1-1 RPDPの提供コマンド一覧

(1/3)

分類	コマンド	機能概要	参照ページ
システムジェネレーション	svgen	サイト構築環境の設定	2-2
	svconf	システム定義情報の登録 IPアドレス、メモリサイズの指定を行う	2-3
	svshconf	システム定義情報の表示	2-4
	svsitecp	サイトのコピー	2-5
	svsitedel	サイトの削除	2-6
コンパイラ	shc	クロスコンパイラ (コンパイラパッケージの一部)	(*)
プログラミング コマンド	optlnk	ライブラリアン (コンパイラパッケージの一部)	(*)
	optlnk	リンカ (コンパイラパッケージの一部)	(*)
アロケータ	svdfa	分割領域の確保、バックアップファイルの生成	2-7
	svdla	分割領域の削除、バックアップファイルの削除	2-9
	svdfs	細分割領域の確保	2-10
	svdls	細分割領域の削除	2-12
	svdfv	VALの登録	2-13
	svdlv	VALの削除	2-14
ローダ	svload	リソースのバックアップファイルへの格納と管理情報への登録	2-15
	svdload	リソースの管理情報からの削除	2-27
	svcomp	登録済みリソースとの比較	2-28
ビルダ	svctask	タスクの生成	2-32
	svdtask	タスクの削除	2-34
	svbuild	間接リンクサブプログラムの登録	2-35
		組み込みサブルーチンの登録	2-36
	svdbuild	間接リンクサブプログラムの削除	2-37
		組み込みサブルーチンの削除	2-38
svirglb	IRGLBの登録/削除	2-39	

(\*) スタート ボタンから下記方法により、マニュアルを起動し、SuperH RISC engine C/C++コンパイラ、アセンブラ、最適化リンケージエディタユーザーマニュアルの“C/C++コンパイラ操作方法”、“最適化リンケージ操作方法”を参照してください。

(1) SHC Ver.7のマニュアル起動方法

- ① [Hitachi Embedded Workshop2] – [Online Manuals [SuperH] – [Japanese (Ver-Rev) ] ]
- ② [Renesas High-performance Embedded Workshop] – [Online Manuals [SuperH] – [Japanese (Ver-Rev) ] ]

(2) SHC Ver.9のマニュアル起動方法

- ① [Renesas] – [High-performance Embedded Workshop] – [Manual Navigator]

(2/3)

分類	コマンド	機能概要	参照ページ		
オンライン デバッガ	svdebug	タスク起動/ 停止	qu	タスクの起動要求	2-54
			ab	タスクの起動禁止	2-55
			re	タスクの起動禁止解除	2-56
			ta	タスクの状態表示	2-57
			su	タスクの実行抑止	2-60
			rs	タスクの実行抑止解除	2-61
			tm	タスクの周期起動	2-62
			ct	タスクの周期起動解除	2-64
			sht	タスクの周期起動表示	2-65
			si	スタック初期化	2-113
			sp	スタック使用量の表示	2-114
		メモリプリン ト/パッチ	md	アドレス指定によるメモリ内容の表示/ 変更	2-66
			sd	名称指定によるメモリ内容の表示/ 変更	2-70
			bs	指定ビットへのデータ設定	2-73
			bg	指定ビットのデータ表示	2-75
			mcp	メモリ内容のコピー	2-77
			mmv	メモリ内容の移動	2-79
			mf	メモリへのパターン値設定	2-81
		ブレークポイ ント	br	ブレークポイントの設定/表示	2-86
			rb	ブレークポイントの解除	2-91
			rd	レジスタの表示	2-92
			rr	レジスタの内容変更	2-95
			go	ブレークポイントからの実行再開	2-96
		システム エラー表示	el	エラーログの表示	2-83
			ss	システムの状態表示	
		現在時刻設定 /表示	st	現在時刻の設定	2-84
			gt	現在時刻の表示	2-85
		アップ/ダウ ンローディ ング、コン ペア	ld	リソースの個別ダウンロード	2-100
			sv	リソースの個別バックアップ	2-108
			cm	バックアップファイルとPCsメモリの内容 比較	2-110
		DHP記録許可 /禁止	dr	DHP記録許可	2-111
			ds	DHP記録禁止	
		ADT	as	ADTの設定/表示	2-97
			ac	ADTの解除	2-99
		その他	svdhp	DHPの表示	2-112
			svadm	アドレスに対するリソース名称の表示	
			ps	デバッグ文の表示開始	2-115
			pe	デバッグ文の表示終了	
			ver	CMUのバージョン表示	2-116
			help	サブコマンド一覧表示	2-117
			q	デバッガの終了	2-118
			!	svdebug実行時の開発系マシン上のコマン ドの実行	

分類	コマンド	機能概要	参照ページ
管理ツール	svmap	マップ情報表示	2-41
	svadm	アドレスに対する名称の表示	2-44
	svsitectl	サイト状態の制御と状態の表示	2-47
立ち上げ	svrpl	リモートローディング	2-48
	svcpuctl	リモート状態制御	2-49
稼働管理	svcpunow	PU負荷率の表示	2-123
	svtimex	タスク稼働率表示	2-124
エラーログ、DHP 表示	svelog	エラーログ情報出力	2-119
	svdhp	DHPトレース情報の表示	2-121

## 第2章 プログラム開発手順

### 2.1 全体フロー

プログラム開発手順の全体フローを図1-2に示します。

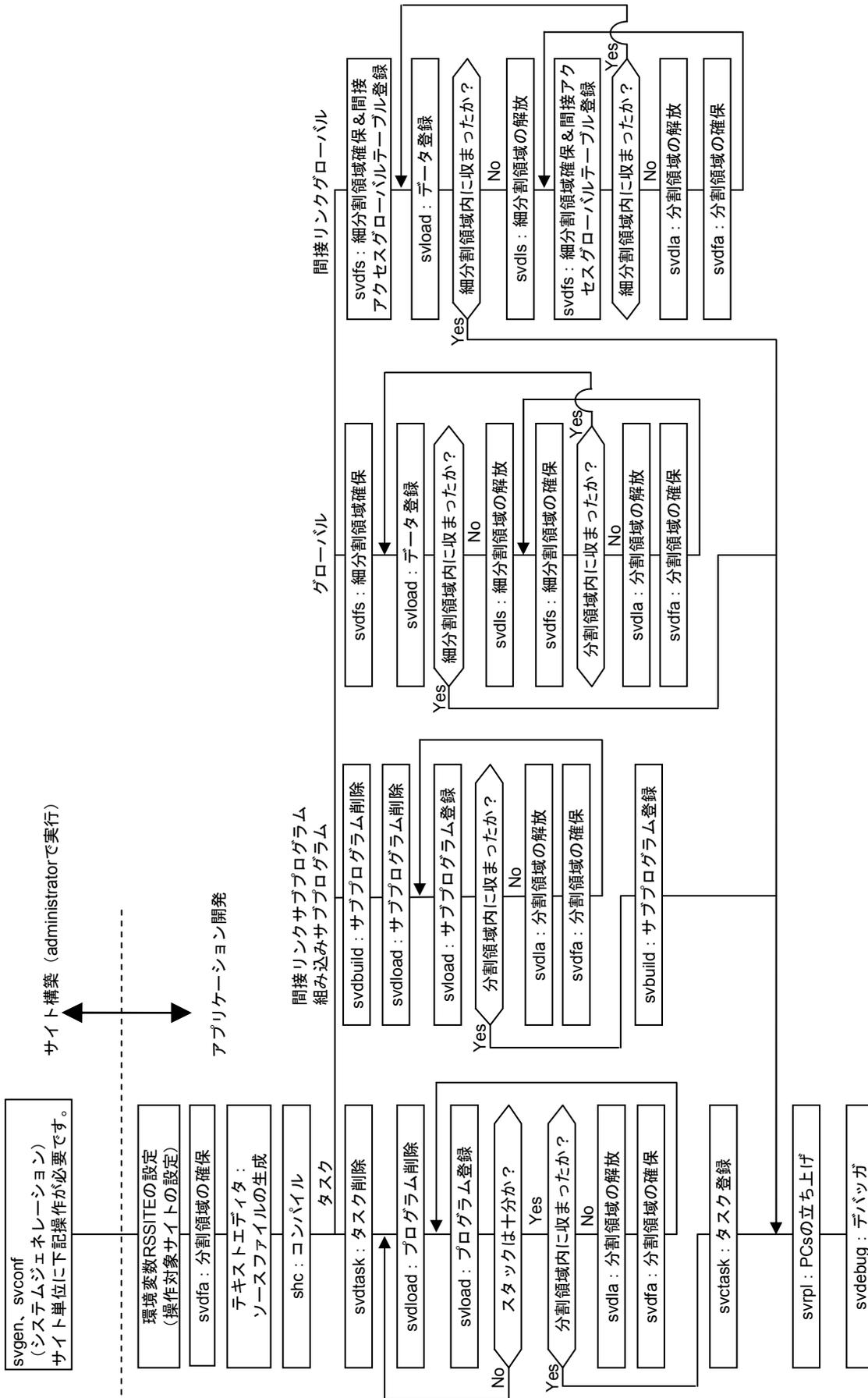


図 1-2 プログラム開発手順全体フロー (1/2) (サイト構築からプログラム開発までの流れ)



2.2 サイト環境

RPDPはPCsごとにサイト名称を割り当て、PCs単位にCPMS上で動作するタスク、サブプログラム、グローバルなどを管理します。

サイト名称はPCs単位のユニークな名称です。これはシステムジェネレーション時にユーザが決定します。

RPDPのシステムジェネレーションは、サイトごとにサイト名称のディレクトリ（これをサイトディレクトリと呼びます）を作成し、サイトディレクトリ下にサイト単位に所有する管理ファイルを配置します。この管理ファイルとは、PCs内メモリの初期値データファイルであるバックアップファイルや、バックアップファイル内に格納されているタスク、サブルーチン、グローバルなどを管理するファイルです。

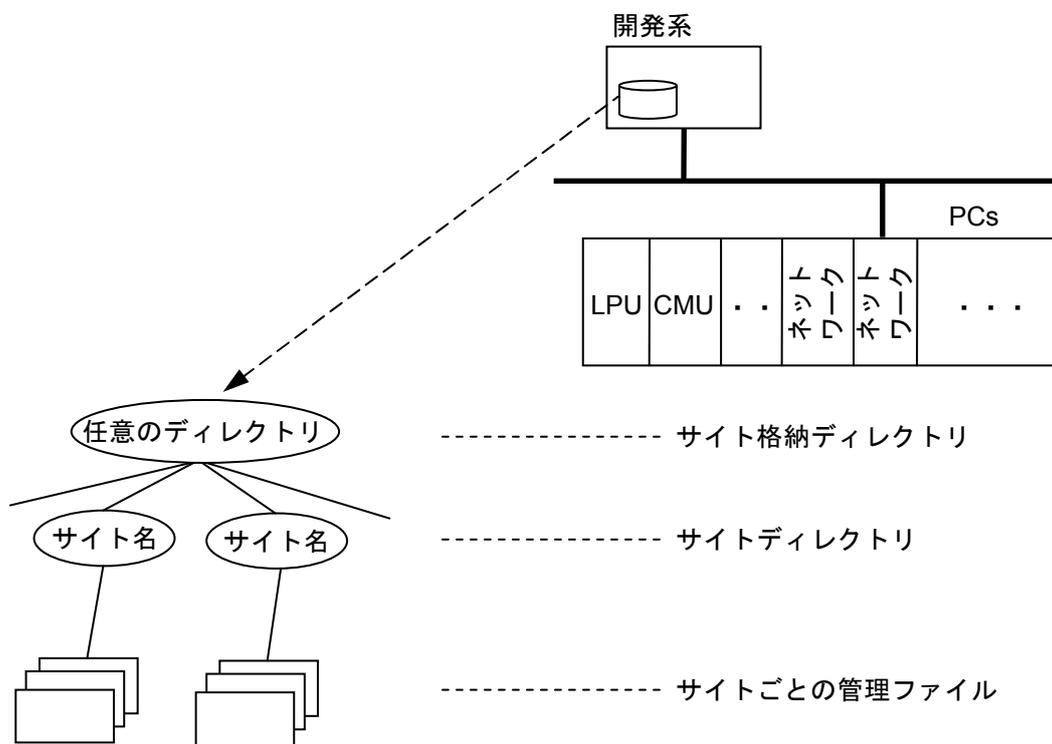


図1-3 PCsのサイトディレクトリ構成

2. 3 主メモリのエリア管理と領域分割

RPDPは開発系マシン上でPCs主メモリの管理を行います。エリア管理の目的は、プログラム、サブプログラム、データを主メモリ上に重複することなく効率よく配置することにあります。RPDPのエリア管理の対象となるメモリ空間は、物理メモリとPCs上でCPMSが管理する論理空間です。物理メモリは、システムジェネレーションで用途ごとにGAREAを定義したサイズで論理空間の先頭からマッピングされます。

図1-4にCPMSが管理する論理空間を、表1-2に論理空間の用途を示します。

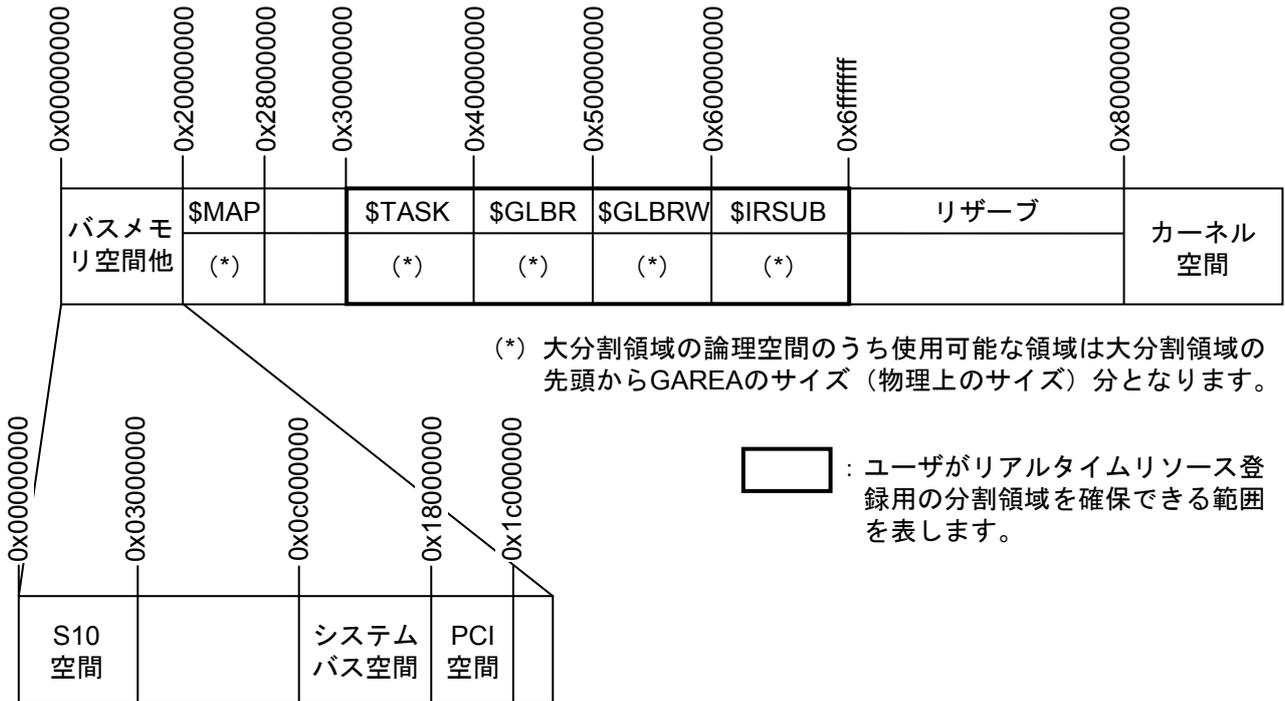


図1-4 CPMSが管理する論理空間

表1-2 各論理空間の用途

論理空間の大分割領域名	用途
\$TASK	タスク（プログラム）を格納するエリアです。
\$GLBR	読み出し専用GLBを格納するエリアです。
\$GLBRW	読み書き両用GLBを格納するエリアです。
\$IRSUB	サブプログラムを格納するエリアです。
\$MAP	RPDPが管理する下記のテーブルを格納します。 ・ IRSUBの間接リンクテーブル (IRSUBT) ・ タスクコントロールブロック (TCB) ・ IRGLBの間接リンクテーブル (IRGLBT) ・ 組み込みサブのテーブル (USLCB) PCsメモリ上の管理情報も格納します。 マップ情報（アロケータ管理テーブル）、パッケージのバージョン情報を格納します。

図1-4に示すCPMSが管理する論理空間のアドレスとサイズを表1-3に示します。

表1-3 各論理空間のアドレスとサイズ

S10V			
論理空間の大分割 領域名 (用途)	アドレス	サイズ	SH4の仮想 アドレス 空間領域名
リザーブ	0x00000000-0x0000ffff	64KB	P0領域 (U0領域)
S10空間	0x00010000-0x01ffffff	32MB-64KB	
リザーブ	0x02000000-0x0ffffff	8MB	
バスメモリ空間	0x0c000000-0x17ffffff	192MB	
PCI空間	0x18000000-0x1bffffff	64MB	
リザーブ	0x1c000000-0x1ffffff	64MB	
\$MAP	0x20000000-0x27ffffff	128MB	
CPMS領域	0x28000000-0x2ffffff	128MB	
\$TASK	0x30000000-0x3ffffff	256MB	
\$GLBR	0x40000000-0x4ffffff	256MB	
\$GLBRW	0x50000000-0x5ffffff	256MB	
\$IRSUB	0x60000000-0x6ffffff	256MB	
リザーブ	0x70000000-0x7ffffff	176MB	
カーネル空間	0x80000000-0x9ffffff	512MB	P1領域
	0xa0000000-0xbffffff	512MB	P2領域
リザーブ	0xc0000000-0xdffffff	512MB	P3領域
	0xe0000000-0xffffffff	512MB	P4領域

図1-5に物理メモリマップの詳細を示します。



図1-5 PCsの物理メモリマップ

SPM : CPMSのエディションデータです。

HKP : Hardware KROM ProgramがROMからコピーされて動作する空間です。

CPMS : OSのプログラム本体でROMからコピーされます。

タスク、サブプログラム、GLB用エリア :

タスク、サブプログラム、GLBがダウンロードされるエリアです。

システムジェネレーション時にGAREAごとのサイズをユーザが指定します。

マップ情報、TCB、IRSUBT、IRGLBT、USLCBもこのエリアにダウンロードされます。

OSワーク : OSが使用するバッファエリアです。DHPエリア、エラーログエリア、ネットワークバッファ等から構成されます。

● タスク、サブプログラム、GLB用エリア詳細

以下にタスク、サブプログラム、GLB用エリアの詳細を示します。

システムジェネレーション時に各エリア（GAREA）のサイズを指定してサイト構築を行ってください。

（\$MAPの容量は固定（約0.7MB）です。）

タスク、サブプログラム、GLB用エリア				
\$MAP	\$TASK	\$GLBR	\$GLBRW	\$IRSUB

\$MAP：PCsメモリ上の管理情報を格納するエリアです。

\$TASK：タスク（プログラム）を格納するエリアです。

\$GLBR：読み出し専用GLBを格納するエリアです。

\$GLBRW：読み書き両用GLBを格納するエリアです。

\$IRSUB：サブプログラムを格納するエリアです。

CPMS上で動作するタスクやサブプログラム、またはそれらが使用するデータを生成する場合には、初めにタスクやサブプログラム、データを格納する大分割領域（GAREA）に分割領域（area）を確保します。グローバル（GLB）の場合は、さらに分割領域を細分割領域（sarea）に分けます。タスクやサブプログラムは細分割領域名称を指定してデータをアクセスします。

（1）分割領域

分割領域（area）はsvdfaで確保し、svdlaで解放します。システムジェネレーションで定義した大分割領域（GAREA）内に複数の分割領域を確保できます。

分割領域を確保すると、確保したサイズ分のバックアップファイルが生成されます。

（2）細分割領域

svdfaで確保した1つの分割領域の中には複数のリソースが配置できます。タスクおよびサブプログラムはsvloadで分割領域内に配置し、svdloadで解放します。GLBの細分割領域（sarea）はsvdfsで確保し、svdlsで解放します。

分割領域（area）を解放するときは、svdlaで解放してください。

2. 4 タスクのエリア配置

CPMS上で動作するすべてのタスクは1つの論理空間内に配置されます。タスクは、\$TASKに確保した分割領域内に格納します。1つの分割領域内に複数のタスクを格納することができます。

textはページ境界（4KB境界）に、data/bssとstackは8バイト境界に配置します。

また、同一タスクのtext/dataとstack/bss、およびOSワークは別ページに配置します。

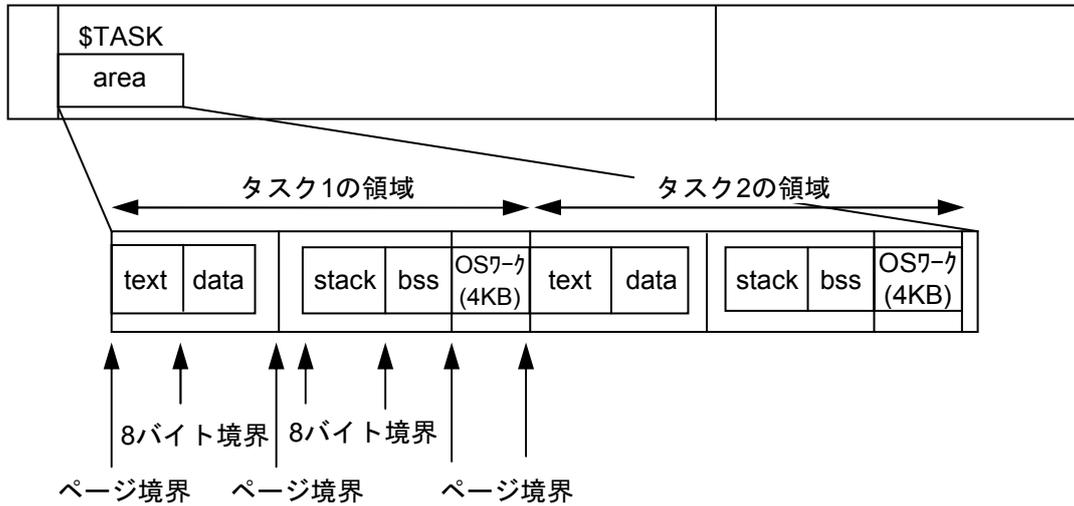


図 1 - 6 論理空間内のタスク配置

● マルチタスクのスタック配置

マルチタスクのときはstackとbssを別ページに配置します。

stackとOSワークはマルチタスクの個数分ロード時に確保し、どこを使用するかはタスク生成時にユーザが指定します（下図は、タスク1とタスク2がマルチタスクの場合を示します）。

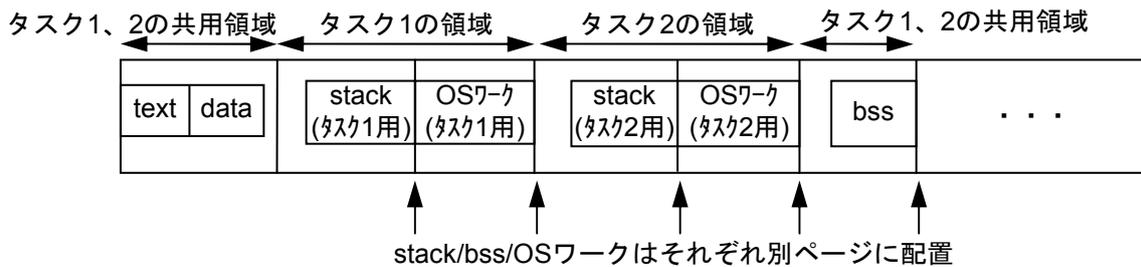


図 1 - 7 論理空間内のタスク配置（マルチタスク）

マルチタスクとは、プログラムのメモリ容量削減を目的とし、1つのプログラムに対してn個のタスクとして生成したものです。マルチタスクは、ロードしたプログラムのtext/data/bss部を共用し、stack部をタスクごとに別エリアに配備することで実現しています。

このため、注意事項としては、マルチタスクを組む1つのタスクで、bss部にデータを書き込むとマルチタスクを組む他のタスクにもその情報が伝わるため、初期状態bssを期待し動作するタスクの動作保証ができません。したがって、マルチタスクの場合は、bssにデータを書き込まないでください。

2.5 IRSUBのエリア配置

CPMS上で動作するすべてのIRSUBは1つの論理空間内に配置されます。IRSUBは、\$IRSUBに確保した分割領域内に格納します。1つの分割領域内に複数のIRSUBを格納することができます。textは32バイト境界に、dataは8バイト境界に配置します。

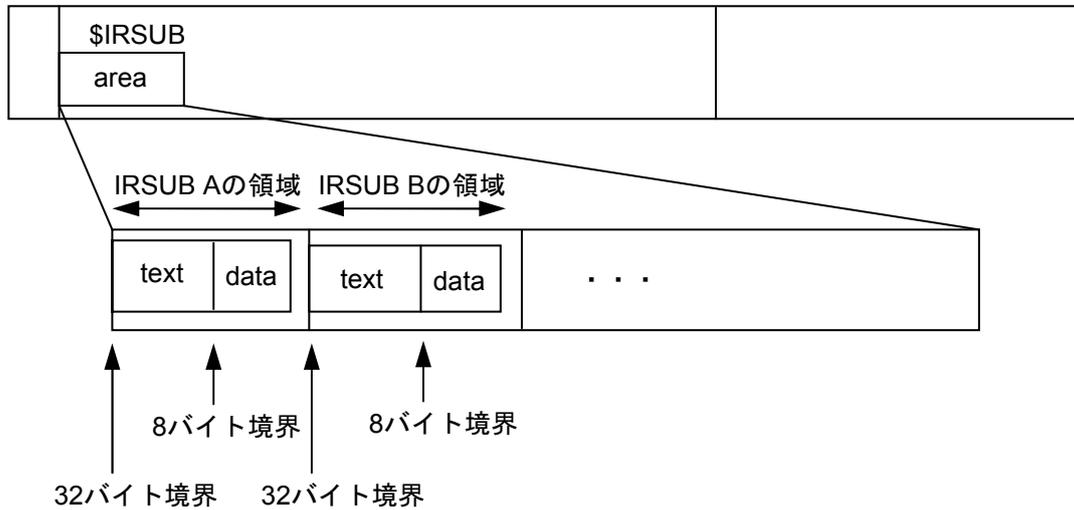


図1-8 論理空間内のIRSUB配置

● マルチエントリ配置

IRSUBマルチエントリでも、text/dataの配置は変わりません。

RPDPの管理ファイル上にマルチエントリされたエントリ名称と相対エントリアドレスを持ち管理します。図1-9にマルチエントリIRSUB（エントリ名称A、B、C）の配置を示します。

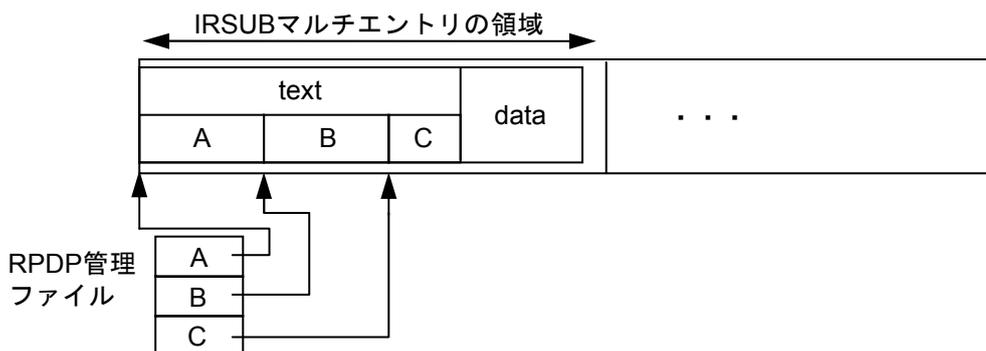


図1-9 論理空間内のIRSUB配置（マルチエントリ）

## 2. 6 プログラムのロードとタスクの生成

アロケータによって決定された管理情報に基づいて、ローダ (svload) はプログラムやデータを area、sareaにロードします。

ローダはグローバルなどのCPMSの資源情報をエリア管理情報から引き出し、それらを設定しながら実行可能モジュールを生成します。生成した実行可能モジュールは、開発系マシンのバックアップファイルに格納されます。

プログラムとしてロードした実行可能モジュールは、ビルダ (svctask) を用いてタスクとして登録します。svctaskはCPMSが管理するタスク制御ブロック (TCB: Task Control Block) にそのタスクの属性を設定します。

## 2. 7 間接リンク常駐サブプログラム

タスクがいくつかのサブプログラムから構成されているとします。このサブプログラムのうち、タスク本体に組み込まれているサブプログラムを内部サブプログラム (ISUB) と呼びます。タスク本体とは別な場所にあり、他のタスクからも共有できるように主メモリ上に常に存在するサブプログラムを常駐サブプログラム (RSUB) と呼びます。

このRPDPでは間接リンクRSUB (IRSUB) をサポートしています。IRSUBはタスクからリンクする管理テーブルを設け、この管理テーブルをリンクするタスクを変更することなく、IRSUB本体の変更を容易に行うことができます。

IRSUBの本体を更新するのはローダ (svload) です。間接リンク用の管理テーブルの更新はビルダ (svbuild) が行います。

## 2. 8 グローバル (GLB)

CPMSではタスク間の主メモリの共有方法としてGLBを提供しています。これは、アロケータを使ってあらかじめCPMSタスクの論理空間内のGLB空間に領域を確保して名前を付けておき、その領域を複数のタスクやサブルーチンで共同に使用するものです。この領域をsvdfaを用いてareaに分割し、さらにsvdfsを用いてsareaに細分割して使用します。

## 2. 9 バリュ (VAL)

ユーザは、プログラム間で共通に使用する定数を外部名称として登録し使用できます。これをバリュ (VAL) と呼びます。VALの登録はsvdfv、削除はsvdlvを用いて行います。

VALはロードモジュールをバックアップファイルにロードするときにローダが設定します。したがって、VALを参照するタスク、サブプログラムをロードする前にVALを登録しておいてください。

## 2. 10 間接リンクグローバル

RPDPでは間接リンクグローバルをサポートしています。間接リンクグローバルは、グローバル本体にリンクする管理テーブルを設定し、このリンク管理テーブルの内容を更新することにより、グローバル本体の更新を容易にしています。

グローバルの本体を更新するのはアロケータ (svdfs)、ローダ (svload) です。svdfsでエリアを定義し、svloadで初期値をロードします。間接リンク用の管理テーブルの更新はビルダ (svirglb) が行います。

2. 11 GLB、VAL、IRSUBプログラミングガイド

プログラム、サブプログラムから使用するGLB、VAL、IRSUBのコーディング方法、リンク方法について示します。

(1) GLB、VALの名称の付け方

表 1-4 GLB、VALの名称の付け方

項目	仕様
最大文字数	14文字（_g、_vは除く）
名称規則	半角英数字、_（アンダーライン） ただし、先頭は英字に限ります。 最終文字は属性を表し、以下の形式にしてください。 GLBのとき_g VALのとき_v
名称のユニーク性	同一名称は使えません。

(2) GLB、VALの使い方

表 1-5 にGLBおよびVALの使い方を示します。

表 1-5 GLBおよびVALの使い方

No.	項目	C言語
1	GLBの宣言（参照側）	extern long name_g[size]; [説明] name : GLB名称 size : GLBのサイズ
2	GLBの参照 （細分割領域指定）	extern long name_g[size]; main(){ long i; i = name_g[index]; } [説明] name : GLB名称 size : GLBのサイズ
3	GLBの宣言（被参照側）	宣言する必要はありません。 No.4に示すように初期値を設定してください。
4	GLBへの初期値設定	long name_g[size] = {1,2,3,...}; [説明] name : GLB名称 size : GLBのサイズ
5	VALの参照	extern long name_v; long y = (long)&name_v; main(){ long x; x = y; } [説明] name : VAL名称

## (3) GLBデータ参照時の注意事項

プログラム作成時にGLBデータを参照する場合、参照するGLBが同一プログラム上で初期値を定義しているかどうかによってデータの取り扱いが異なります。そのため、以下に示す点に注意してGLBデータを参照するプログラムを作成してください。

## ① 参照するGLBが同一プログラム内で定義されていない場合

上記の条件は、ソースプログラムまたはsvloadで結合されるオブジェクトファイルの基となるソースプログラム内で参照するGLBが表1-5のNo.3、No.4に示すような定義をしていない場合です。

この場合、以下の(a)、(b)について注意してください。

## (a) GLBの宣言

GLBの宣言は、表1-5のNo.1に示すように各名称に対する容量宣言が行えます。コンパイラやアセンブラは、この容量に対して、svdfsコマンドにて確保した領域の大きさとの合理性チェックは行いません。したがって、プログラムが実際の領域を超えたアドレスを参照してもエラーとなりません。

(例) 宣言したエリアを超えたアドレスの参照

<アロケータ>

```
svdfs usrresp0 glb2 100
```

<c>

```
extern long glb2_g[100];
      ⋮
      :
      glb2_g[100] = ……;
```

} エラー検出されません。

## (b) 相対アドレスの参照

GLBは、名称± $\alpha$  ( $\alpha$ は相対バイトアドレスを示します。)の形式で参照できます。その場合の範囲は、 $-2^{31} \leq \alpha \leq 2^{31}-1$ です。

## ② 参照するGLBが同一プログラム内で定義されている場合

上記の条件は、ソースプログラムまたはsvloadで結合されるオブジェクトファイルの基となるソースプログラム内で参照するGLBが、表1-5のNo.3、No.4に示すような定義をしている場合です。

この場合、以下の(a)～(c)について注意してください。

## (a) GLB名称だけの参照

プログラム中に初期値定義のあるGLBの名称だけを参照する場合は、特に制限事項はありません。

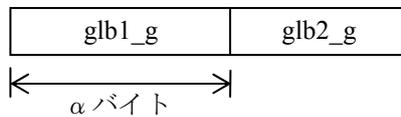
(例) 名称だけの参照

```
<c>
extern long   glb2_g ;
long   glb1_g[3] = { (long)&glb2_g , 0 , 100 } ;

      ⋮
      ⋮
      ⋮
      ⋮
      glb2_g = glb1_g[0] ;
```

(b) GLB先頭から相対アドレスを参照

名称 +  $\alpha$  の形式で参照する場合、 $\alpha$  の値は定義した範囲を超えてはいけません。ただし、範囲外チェックエラーは検出しませんので注意してください。



すなわち、 $glb1\_g + \beta$  と記述する場合、 $0 \leq \beta < \alpha$  でなければなりません。

(例) 相対アドレス参照

```
<c>
int   glb1_g[3] = { 1, 2, 3 } ;
int   glb2_g[2] = { (long)&glb1_g[0], 0 } ; ... 相対アドレスは範囲内

a = glb1_g[3] + glb2_g[4] ; ..... 相対アドレスは範囲外で
                                     す。
```

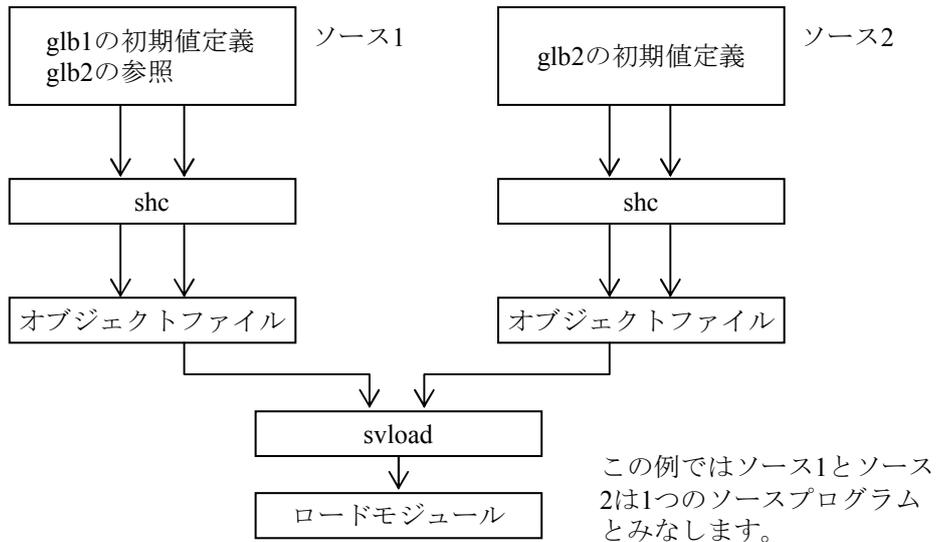
(c) svload操作

例えば、text部とGLB初期データを持つソースプログラムから作成されたロードモジュールをsvloadでロードするとき、単にプログラムまたはサブプログラムとしてロードしたのでは初期値データはロードされません。初期値データをロードするためには、改めて+Dオプションを付けてsvloadを実行してください。つまり、1つのロードモジュールに対してオプションを変えて2度svloadを行うこととなります。初期値データだけのファイルを作成し、別々に作業してください。1つのソースプログラムで複数のGLB初期値を定義できます。

③ リンク上の注意事項

すでに①、②でも述べましたが、svloadでオブジェクトファイルを複数結合する場合、例えそれぞれが別のソースファイルであっても、結合した結果1個のファイルであるものとします。

(例) 2つのオブジェクトファイルの結合

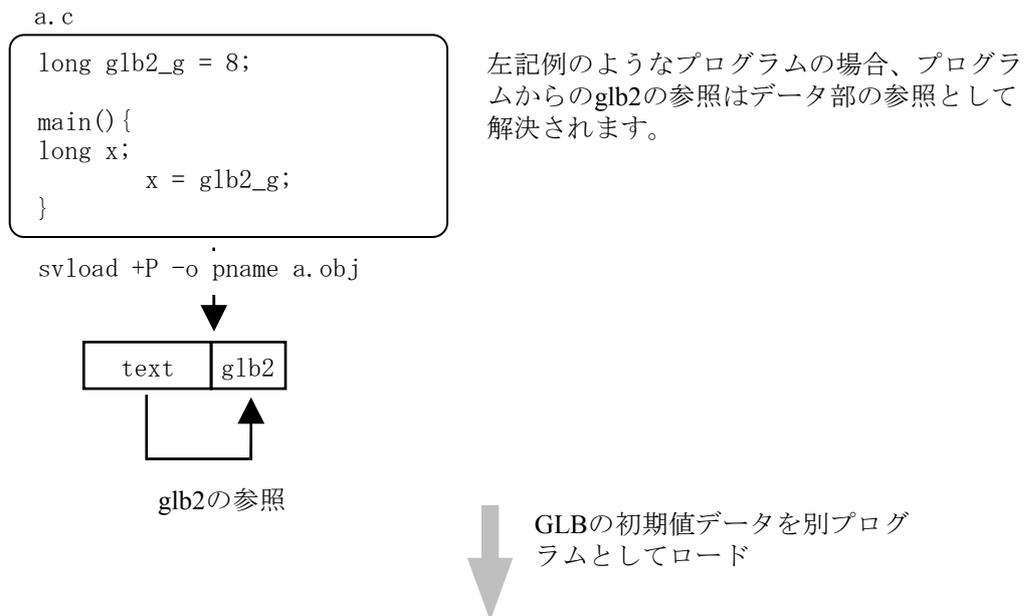


④ 参照するGLBの初期値を同一プログラム上で定義している場合

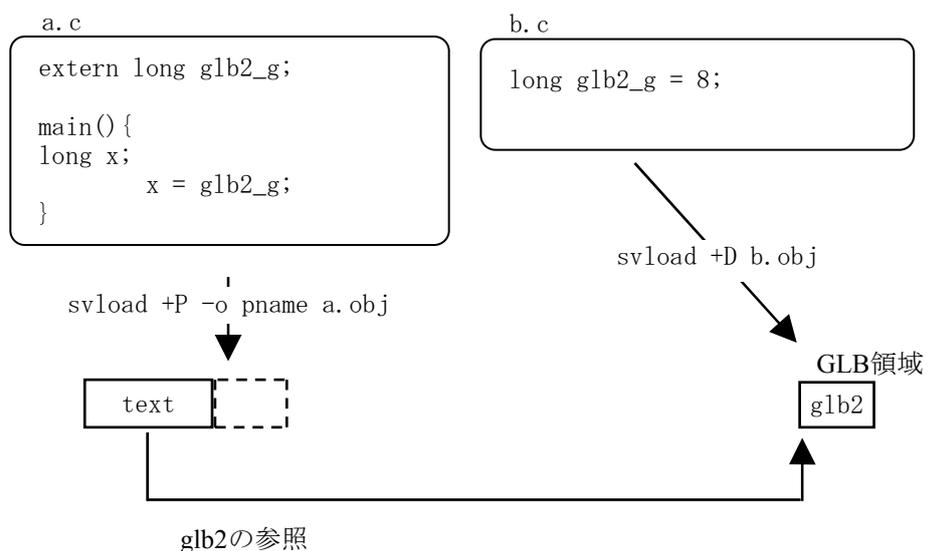
プログラム作成時にGLBデータを参照する場合、参照するGLBの初期値を同一プログラム上で定義しないでください。

GLBの初期値定義と参照が同一プログラム内にある場合、参照はローカルなデータの参照として解決されます。

(例) ■ プログラムとGLBデータを同一のプログラムに記述した場合



■ プログラムとGLBデータを分離した場合



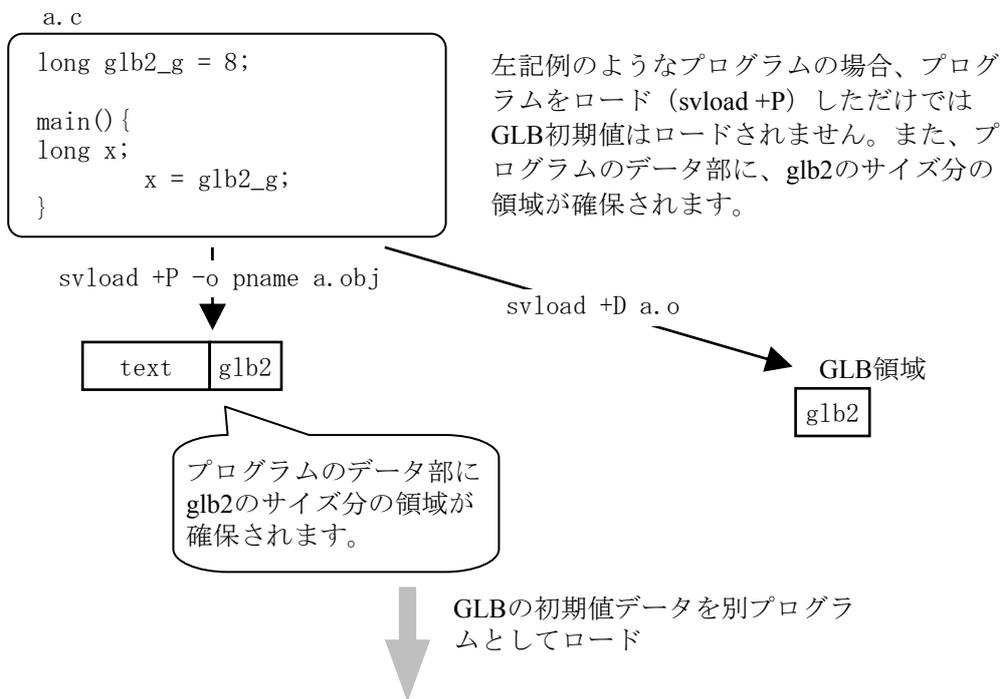
(4) GLBデータロード時の注意事項

① プログラムとGLBデータの混在

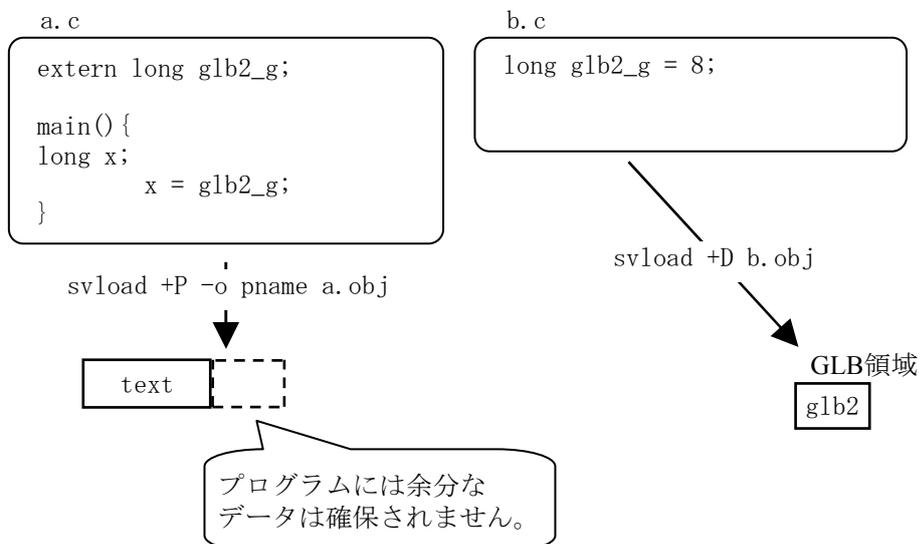
テキスト部を持つプログラム、サブプログラムと、GLBの初期値データを同一のプログラムに記述した場合、ローダ (svload) でプログラム (+P)、サブプログラム (+I) としてロードしても、GLBの初期値データはロードされません。GLBの初期値データをロードするためには、改めて+Dオプションを指定してロードしなければなりません。つまり1つのプログラムをオプションを変えて2回ロードすることになります。

また、プログラム、サブプログラムのデータ部にはGLB初期値データのサイズ分のエリアが確保されてしまいます。このため、GLBの初期値データを定義するプログラムは、GLBに初期値を定義するだけのプログラムとしてください。

(例) ■ プログラムとGLBデータを混在させた (同一のプログラムに記述した) 場合



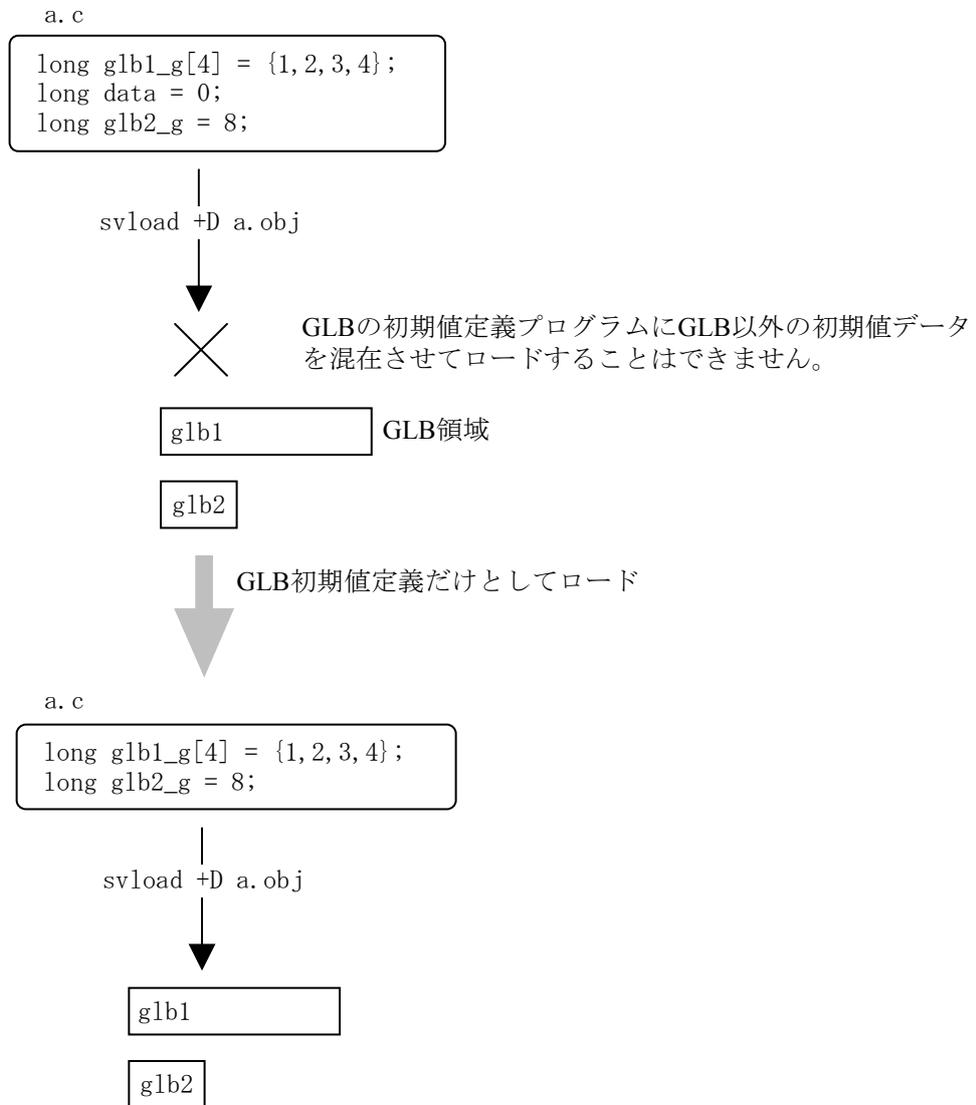
■ プログラムとGLBデータを分離した場合



② GLB初期値とGLB以外のデータの混在

1つのプログラムに複数のGLB初期値データを定義することができます。ただし、GLBの初期値定義プログラムにはGLB初期値以外のデータを定義することはできません。

(例) 同一のプログラムにGLB初期値定義とGLB以外のデータを混在させた場合



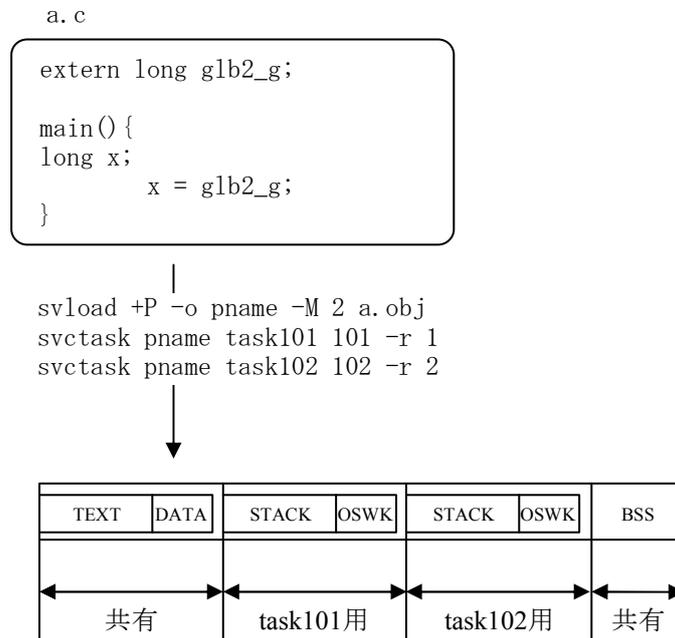
## (5) IRSUBの使い方

表 1 - 6 IRSUBの使い方

項目	C言語
IRSUBの参照 (IRSUB名称指定)	<pre>main(){     name(); }</pre> <p>[説明] name : IRSUB名称</p>
IRSUBの参照 (irsubad関数指定)	<pre>void *irsubad(); main(){     long no;     long (*adr)();     no = xxx;     adr = irsubad(no);     if(adr != 0){         (*adr)();     }else{         /* IRSUB未登録処理 */     } }</pre> <p>[説明] xxx : IRSUB番号 irsubad関数を使用して間接リンクテーブルからアドレスを参照して使用します。</p>

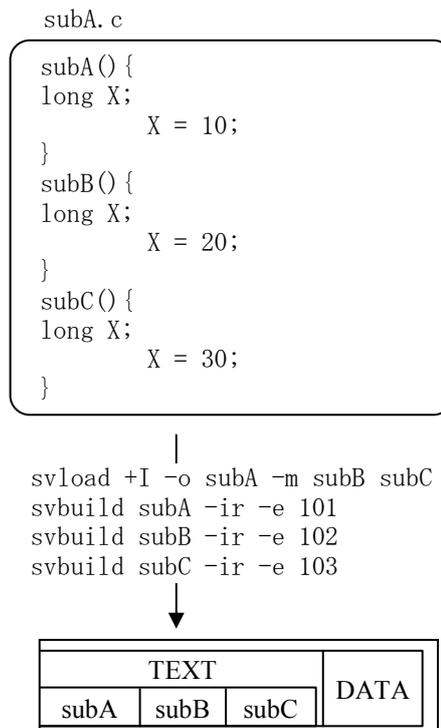
(6) マルチタスクの生成方法

1つのプログラムを2つのマルチタスクとして生成する場合、下記のようにsvload/svctaskコマンドを実行します。



(7) マルチエントリIRSUBの生成方法

サブプログラムを3つのマルチエントリIRSUBとして生成する場合、下記のようにsvload/svbuildコマンドを実行します。1つのソースファイル内に3つのモジュールがある場合を以下に示します。



## 2. 12 CPMS上のプログラム作成の制約条件

CPMS上で動作するリアルタイムプログラム作成に当たっては、以下に示す制約事項があります。

### (1) オーバレイ構造不可

CPMSはタスク、常駐サブプログラムのオーバレイ構造は許していません。したがって、タスク、常駐サブプログラムを作成するときは、そのプログラムが大きくなりすぎないように注意してください。

### (2) バルクサブルーチン未サポート

CPMSは補助メモリ上にサブルーチンを置いておき、必要に応じて主メモリに乗せて動作させるバルクサブルーチンはサポートしていません。間接リンク常駐サブプログラム (IRSUB) とするか、タスク内に組み込んだ内部サブプログラム (ISUB) としてください (コントローラには補助メモリを付けることはできません)。

### (3) 間接リンク常駐サブプログラム (IRSUB) 作成上の注意

IRSUBは主記憶装置上に常駐し、複数のメインプログラムより共通に使用されます。そのため、IRSUBはこれを使用するメインプログラムとは独立な主記憶装置の領域を占有します。また、同一時刻に複数のメインプログラムにより使用されるのでリエントラントにしてください。

リエントラント性のないプログラムはIRSUBにはできません。リエントラントとは、あるメインプログラムがそのIRSUBを使用中でも、他のメインプログラムがその同じIRSUBを使用できることを言います。

以下に正しいIRSUBの作成方法について説明します。

リエントラントなIRSUBは処理手続き部 (text部)、データ部 (data部) からなる不変部分と作業エリアからなる可変部分の2つに分離されます。不変部分は複数のメインプログラムが共有します。可変部分は各メインプログラムが各メインプログラムの可変部分に確保し、IRSUBはメインプログラムに確保された可変部分を使用します。したがって、IRSUBが使用する可変部分はスタックエリアを参照するようにプログラミングしてください。IRSUBは初期値なし作業エリア (bss部) を使用できません。

## 第2章 プログラム開発手順

リエントラントなIRSUBを作成する場合、下記の3点に注意してください。

- (a) 作業エリアはすべてスタックとする。
- (b) IRSUBが複数のプログラムで構成されている場合、プログラム間共通エリアを使用しない。
- (c) 静的変数の初期値を定義している場合、その値を変更しない。

上記(a)、(b)は、コンパイルリスト、またはリンケージマップリスト中のセクション情報で、Bセクションのサイズが0になっていることで確認できます。

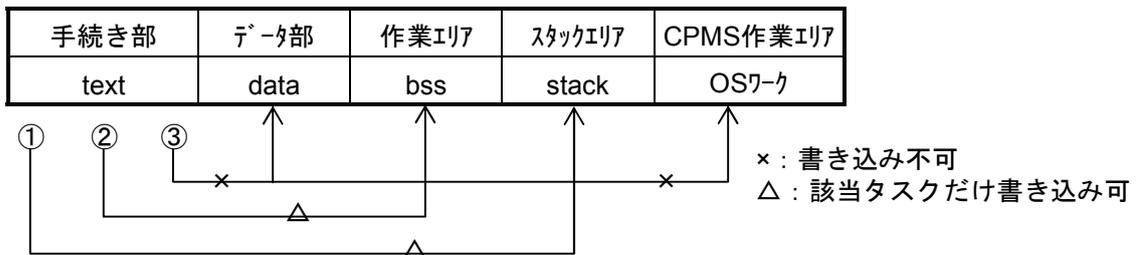


図 1-10 書き込みの可否

- ① : スタックエリアへの書き込みを示します。該当タスクはスタックエリアへ書き込むことができます。
- ② : 作業エリアへの書き込みです。通常IRSUBでは作業エリアを確保せず、また書き込みもしないでください。該当タスクは作業エリアへ書き込むことができます。
- ③ : データ部への書き込みです。タスクはデータ部へ書き込めません。

以下に各言語でのリエントラントなIRSUBの作成時の注意点を示します。

<Cプログラム例>

```

int b1 ; ..... ①
int d1 = 10 ; ..... ②
static int b2 ; ..... ③
static int d2 = 100 ; ..... ④
ex() {
static int b3 ; ..... ⑤
static int d3 = 1000 ; ..... ⑥
int s1 ; ..... ⑦
int s2 = 20 ; ..... ⑧

    :
    :
    :
}

```

①で宣言したb1に対して、書き込みを行うプログラムはノンリエントラントとなります。

②で宣言したd1に対して、書き込みを行うプログラムはノンリエントラントとなります。

③で宣言したb2に対して、書き込みを行うプログラムはノンリエントラントとなります。

④で宣言したd2に対して、書き込みを行うプログラムはノンリエントラントとなります。

⑤で宣言したb3に対して、書き込みを行うプログラムはノンリエントラントとなります。

⑥で宣言したd3に対して、書き込みを行うプログラムはノンリエントラントとなります。

⑦、⑧で宣言したs1、s2に対して、書き込みを行ってもプログラムのリエントラント性は損われません。IRSUBとして使用する場合には、⑦、⑧のような変数だけとしてください。

以下に各変数が割り当てられている領域について説明します。

b1は通常bss領域に割り当てられます。(注)

b2はbss領域に割り当てられます。

b3はbss領域に割り当てられます。

d1はdata領域に割り当てられます。

d2はdata領域に割り当てられます。

d3はdata領域に割り当てられます。

s1はstack領域に割り当てられます。

s2はstack領域に割り当てられます。

(注) 他のプログラムでb1に初期値を設定している場合にはdata領域に割り当てられます。

## 第2章 プログラム開発手順

(4) プログラムの再配置はできない。

プログラム、サブプログラムの再配置はありません。一旦動作エリアが確定したプログラム、サブプログラムを別のエリアでそのまま動かすことはできません。プログラム、サブプログラムを別のエリアに動かすにはプログラム、サブプログラムの削除・再登録を行ってください。

(5) 名称は14文字まで

プログラム、サブプログラムの名称の文字数は、14文字以内としてください。GLB、VALの名称も14文字以内としてください。C言語でのGLB、VALの表現は、これらの名称の後に各々\_g、\_vを付けた16文字以内としてください。

(6) GLB、VALの名称

\_g、\_vで終わる名称は外部名として宣言すると、GLB、VALとして扱われます。したがって、GLB、VALを使わないプログラムでは、名称の末尾が\_g、\_vとならないように名称を付けてください。また、\_bで終わる名称も将来拡張用として予約されていますので使用しないでください。

(7) 外部名称はユニークにしてください。

外部名称はシステム内のGLB名、プログラム名、サブプログラム名、VAL名全体でユニークに付けてください。

(8) 使用してはいけない名称があります。

プログラム作成において使用できない名称や、使用上注意の必要な名称があります。詳細については、「付録A プログラムで使用できる名称」を参照してください。

(9) プログラムの構造

CPMSのもとで動くプログラムの構造は次のようになっています。

text	プログラムのプロシジャが入るエリア
data	プログラムの初期データが入るエリア
stack	タスクが使う動的なワークエリア
bss	プログラムが使う静的なワークエリア
OSワーク	OSが使う動的なワークエリア

これらのエリアの大きさは各々4バイトの整数に補正され、かつ各エリアの先頭アドレスも8または4096の倍数になるように配置されます。配置の詳細については「2. 5 IRSUBのエリア配置」を参照してください。

(10) 先頭アドレスの制約

GLBエリアはデフォルト値では4の倍数になるようにアロケータで補正します。

(11) 初期値の取り扱い

初期値の取り扱いは、下記のようになりますので注意してください。

領域	CPMS
data	プログラムされた値
bss	不定
stack	不定

(12) GLB初期値設定データの大きさ

オブジェクトファイルのデータの大きさは、コンパイラのアラインメント処理により、ソースプログラムで定義した以上の容量となる場合がありますので注意してください。以下に具体例を示します。

CPMSでは、データアクセスの高速化を目的とし、データ型などに合わせ配置（アドレス）を固定化するナチュラルアラインメント方式を採用しています。データの配置は、コンパイラやリンカが自動的に行うため、ユーザは、アラインメントを意識せずコーディングすることができます。しかし、GLBの初期値設定データの実際のサイズは、以下に示すようにコーディングした構造体のサイズより大きくなる場合がありますので注意してください。

<コーディングの大きさ : 16バイト>    <初期値設定データのメモリ上の大きさ : 24バイト>

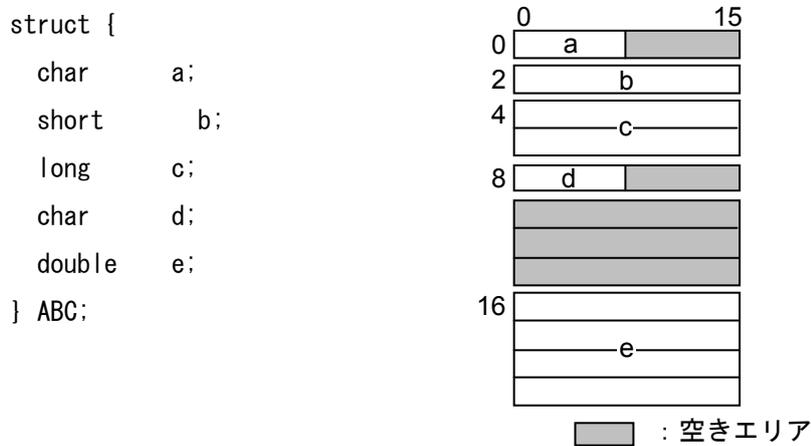


図 1-11 データサイズ比較

<対処方法>

(1) 構造体内の空きエリアをなくすような構成（データ配置順）とすること。

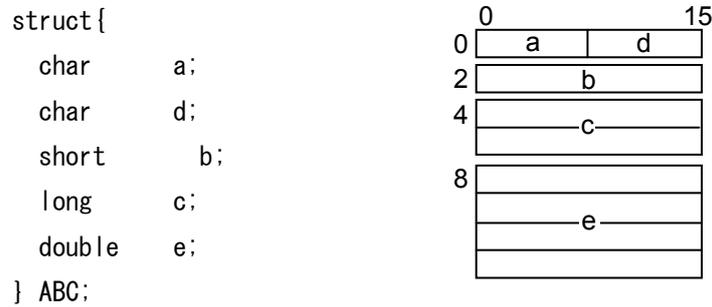


図 1-12 データ配置順を考慮した構造体宣言例

(2) やむを得ず空きエリアが入る場合は、構造体内に空きエリアであることを明示的に宣言すること（明示的にしないとマシンによって空きエリアが確保される場合と確保されない場合があります）。

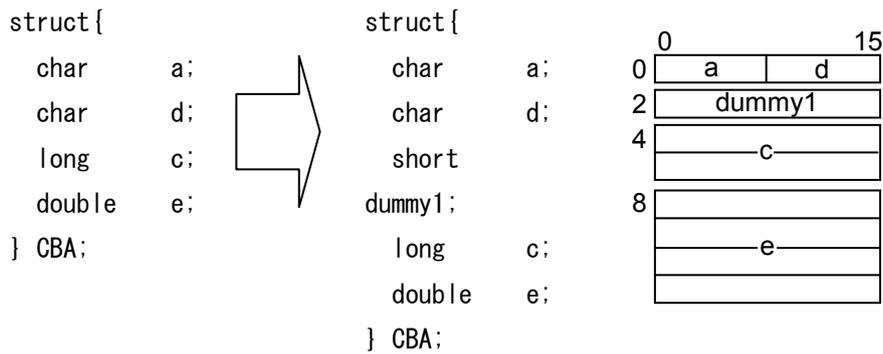


図 1-13 空きエリアを明示的に宣言した例

(3) 構造体の大きさ（バイト数）は、構造体内の最大型の倍数となるようにすること。構造体の配列を確保するとき、2つ目以降の構造体先頭アドレスが、アクセス可能なアドレスに割り当てられるようにする。

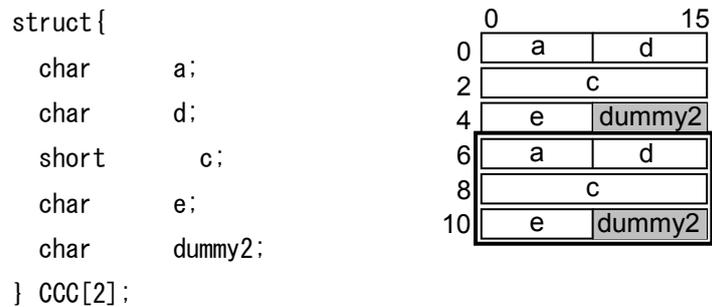


図 1-14 構造体サイズを考慮した宣言例

## 2. 13 名称管理について

## &lt;Windows®のファイル名称仕様&gt;

Windows®上でファイルやディレクトリを扱う場合、新規に作成するときは大文字／小文字が区別されて作成されます（指定した名称そのままで作成されます。）が、それ以降の参照、削除などの各種操作は、大文字／小文字の区別なしで扱われます。以下に例を示します。

- ファイル名“A”を作成 → “A”が作成されます。“a”があれば“a”を上書きし、ファイル名は“a”のままです。
- ファイル名“a”を作成 → “a”が作成されます。“A”があれば“A”を上書きし、ファイル名は“A”のままです。
- ファイル名“A”を参照 → “A”または“a”を参照します。
- ファイル名“a”を参照 → “A”または“a”を参照します。
- ファイル名“A”を削除 → “A”または“a”を削除します。
- ファイル名“a”を削除 → “A”または“a”を削除します。

## &lt;S10V RPDPの名称管理仕様&gt;

S10V RPDPでは、プログラムやサブプログラムなどはファイルとして作成しますが、大文字／小文字の扱いにおいてWindows®の仕様上ファイル名称のコンフリクトが発生するため、以下の扱いとします。

- ・ユーザ指定の名称を基にファイルやディレクトリを作成するコマンド（svloadとsvgen）においては、新規登録時はユーザが指定した名称そのままにファイルやディレクトリを作成、管理し、以降の登録では大文字／小文字区別なしで二重定義をチェックし（“AB”で登録してあった場合、“AB”、“Ab”、“aB”、“ab”による登録は二重定義エラー）、ファイルやディレクトリの上書きを防止します。

該当するのは下記の名称です。

## ■ svloadコマンド

タスク用プログラム名、サブプログラム（IRSUB、ULSUB）名

## ■ svgenコマンド

サイト名、ユニット名

## ■ svdebugコマンド

-o、-iオプションで指定するファイル名

svサブコマンドの-fオプションで指定するファイル名

以上の名称に関して、従来大文字／小文字の違いだけで区別していた場合は見直しが必要となります。

- ・上記以外のコマンドも大文字／小文字を区別しません。

[ “svload +P -o AB AB.obj .... ” にてプログラム登録してあった場合の例]

(1) svmapコマンドでのプログラム名表示

ABと表示します。登録時にユーザが指定した名称そのままです。

(2) svloadコマンドによるプログラム登録

svload +P -o AB AB.obj  
svload +P -o Ab AB.obj  
svload +P -o aB AB.obj  
svload +P -o ab AB.obj

いずれも二重定義エラーとなります。

(3) svdloadコマンドによるプログラム削除

svdload AB +P  
svdload Ab +P  
svdload aB +P  
svdload ab +P

いずれもプログラムABを削除します。

(4) svctaskコマンドによるプログラムを指定してのタスク登録

svctask AB タスク名称 タスク番号  
svctask Ab タスク名称 タスク番号  
svctask aB タスク名称 タスク番号  
svctask ab タスク名称 タスク番号

いずれも同一プログラム名でタスク登録します。

## 第3章 インストールとRPDP実行環境

### 3.1 インストール

RPDPを含むP.P. (プログラムプロダクト: P.P.名称 S10V\_RPDP) をインストーラにてインストールします。インストールは、Administratorでログインしてから実施してください。

S10V\_RPDPのディスクをドライブに挿入し、ファイルマネージャ (またはWindows®エクスプローラ) から、DISK1フォルダ内の“setup.exe”を実行してください。

セットアップ画面が表示されますので、画面の指示に従ってください。

インストール後はログインし直してください。

### 3. 2 RPDP実行環境

RPDPを使用するためには、以下に示す環境変数の設定が必要です。

環境変数のデフォルト値は、RPDPインストール時に自動で設定されます。ただし、RPDPインストール時にすでに設定済みの環境変数はPATHとHLNK\_DIRを除き、上書きされません。

型式：S-7895-10Pをインストール時

表1-7 RPDP実行環境の環境変数一覧（SHC Ver7）に示す環境変数のデフォルト値が設定されます。

型式：S-7895-63Pインストール時

表1-8 RPDP実行環境の環境変数一覧（SHC Ver9）に示す環境変数のデフォルト値が設定されます。

システムで共通に使用する環境変数は、[コントロールパネル] - [システム] の [環境] タブから、システムの環境変数に設定してください。

shcコンパイラの動作に必要な環境変数の意味および設定内容の詳細は、shcコンパイラパッケージのマニュアルを参照してください。

表 1-7 RPDP実行環境の環境変数一覧 (SHC Ver7)

環境変数	設定内容	デフォルト値	設定単位	備考
SHCPU	CPU種別を設定します。 “SH4” を設定してください。	SH4	システムで 共通	
SHC_INC	コンパイラのインクルード ファイル格納ディレクトリを 設定します。	%SystemDrive%\Hew2\Tools \Hitachi\Sh\7_1_0\include	システムで 共通	以下に示すバージョン以外の shcコンパイラパッケージがイ ンストールされている場合
SHC_LIB	コンパイラのインストール ディレクトリを設定します。	%SystemDrive%\Hew2\Tools \Hitachi\Sh\7_1_0\bin	システムで 共通	は、設定を行いません。 7.0B, 7.0.01, 7.0.02, 7.0.03, 7.1.00, 7.1.01
SHC_TMP	コンパイラがテンポラリファ イルを作成するディレクトリ を設定します。	%SystemRoot%\renix\tmp	システムで 共通	
HLNK_DIR	svloadコマンドのライブラリの サーチパスを設定します。	%SystemRoot%\renix\s10v \lib;%HLNK_DIR%	システムで 共通+ユー ザ単位に追 加可能	%SystemRoot%\renix\ s10v\libを先頭に追加します。
HLNK_TMP	リンカージェディタがテンポ ラリファイルを作成するディ レクトリを設定します。	%SystemRoot%\renix\tmp	システムで 共通	
PATH	コンパイラおよびRPDPのコマ ンド格納ディレクトリを設定 します。	%SystemRoot%\renix\s10v \bin;%SystemDrive%\Hew2 \Tools\Hitachi\Sh\7_1_0\bin; %PATH%	システムで 共通	%SystemRoot%\renix \s10v\bin;%SystemDrive\ \Hew2\Tools\Hitachi\Sh \7_1_0\binを先頭に追加しま す。 以下に示すバージョン以外の shcコンパイラパッケージがイ ンストールされている 場合は、設定を行いません。 7.0B, 7.0.01, 7.0.02, 7.0.03, 7.1.00, 7.1.01
RSSITE	操作対象サイト名称を設定し ます。	なし	ユーザ単位 に設定可能	
RSUTYP	使用者種別を設定します。 (s: システム、u: ユーザ) ALBコマンドの-Sオプション を省略した場合のデフォルト の処理モードとなります。	なし	ユーザ単位 に設定可能	

表 1-8 RPDP実行環境の環境変数一覧 (SHC Ver9)

環境変数	設定内容	デフォルト値	設定単位	備考
SHCPU	CPU種別を設定します。 “SH4” を設定してください。	SH4	システムで 共通	
SHC_INC	コンパイラのインクルード ファイルの格納ディレクトリ を設定します。	%SystemDrive%\Program Files \Renesas\Hew\Tools \Renesas\Sh\9_4_0\include	システムで 共通	
SHC_LIB	コンパイラのインストール ディレクトリを設定します。	%SystemDrive%\Program Files \Renesas\Hew\Tools \Renesas\Sh\9_4_0\bin	システムで 共通	
SHC_TMP	コンパイラがテンポラリファ イルを作成するディレクトリ を設定します。	%SystemRoot%\renix\tmp	システムで 共通	
HLNK_DIR	svloadコマンドのライブラリの サーチパスを設定します。	%SystemRoot%\renix\s10v \lib;%HLNK_DIR%	システムで 共通+ユー ザ単位に追 加可能	%SystemRoot%\renix\s10v \libを先頭に追加します。
HLNK_TMP	リンケージエディタがテンポ ラリファイルを作成するディ レクトリを設定します。	%SystemRoot%\renix\tmp	システムで 共通	
PATH	コンパイラおよびRPDPのコマ ンド格納ディレクトリを設定 します。	%SystemRoot%\renix\s10v \bin;%SystemDrive%\ \Program Files\Renesas\Hew \Tools\Renesas\Sh\9_4_0\bin; %PATH%	システムで 共通	%SystemRoot%\renix\s10v \bin;%SystemDrive%\ Program Files\Renesas\Hew \Tools\Renesas\Sh\9_4_0 \bin;を先頭に追加します。
RSSITE	操作対象サイト名称を設定し ます。	なし	ユーザ単位 に設定可能	
RSUTYP	使用者種別を設定します。 (s : システム、u : ユーザ) ALBコマンドの-Sオプション を省略した場合のデフォルト の処理モードとなります。	なし	ユーザ単位 に設定可能	

### 3. 3 RPDPのコマンド格納ディレクトリ

RPDPのコマンド格納ディレクトリは、%windir%\¥renix¥s10v¥binディレクトリです。

#### 3. 3. 1 環境変数設定例

以下に環境変数の設定例を示します。

shcコンパイラパッケージ実行環境設定例 (S-7895-10P)

```
SHCPU=SH4
SHC_INC=C:\¥Hew2¥Tools¥Hitachi¥Sh¥7_1_1¥include
SHC_LIB=C:\¥Hew2¥Tools¥Hitachi¥Sh¥7_1_1¥BIN (1)
SHC_TMP=%windir%\¥renix¥tmp
HLNK_DIR=%windir%\¥renix¥s10v¥lib;C:\¥users¥subsyst¥lib (2)
HLNK_TMP=%windir%\¥renix¥tmp
PATH=%PATH%;%windir%\¥renix¥s10v¥bin;C:\¥Hew2¥Tools¥Hitachi¥Sh¥7_1_0¥BIN (3)
```

- (1) shcコンパイラパッケージ (Ver7.1) をデフォルトの設定でインストールした場合のコンパイラの格納ディレクトリC:\¥Hew2¥Tools¥Hitachi¥Sh¥7\_1\_1¥binを設定しています。
- (2) RPDPの標準ライブラリ格納ディレクトリC:\¥WINNT¥renix¥s10v¥libと、ユーザ独自のライブラリ格納ディレクトリC:\¥users¥subsyst¥libを設定しています。
- (3) S10V RPDPのコマンド格納ディレクトリC:\¥WINNT¥renix¥s10v¥binと、shcコンパイラの格納ディレクトリC:\¥Hew2¥Tools¥Hitachi¥Sh¥7\_1\_1¥binを設定しています。

shcコンパイラパッケージ実行環境設定例 (S-7895-63P)

```
SHCPU=SH4
SHC_INC=C:\¥Program Files¥Renesas¥Hew¥Tools¥Renesas¥Sh¥9_4_0¥include
SHC_LIB=C:\¥Program Files¥Renesas¥Hew¥Tools¥Renesas¥Sh¥9_4_0¥BIN (4)
SHC_TMP=%windir%\¥renix¥tmp
HLNK_DIR=%windir%\¥renix¥s10v¥lib;C:\¥users¥subsyst¥lib (5)
HLNK_TMP=%windir%\¥renix¥tmp
PATH=%PATH%;%windir%\¥renix¥s10v¥bin;C:\¥ProgramFiles¥Renesas¥Hew¥Tools¥Renesas¥Sh¥9_4_0¥BIN (6)
```

- (4) shcコンパイラパッケージ (Ver9) をデフォルトの設定でインストールした場合のコンパイラの格納ディレクトリC:\¥Program Files¥Renesas¥Hew¥Tools¥Renesas¥Sh¥9\_4\_0¥binを設定しています。
- (5) RPDPの標準ライブラリ格納ディレクトリ%windir%\¥renix¥s10v¥libとユーザ独自のライブラリ格納ディレクトリC:\¥users¥subsyst¥libを設定しています。
- (6) S10V RPDPのコマンド格納ディレクトリ%windir%\¥renix¥s10v¥libとshcコンパイラの格納ディレクトリC:\¥Program Files¥Renesas¥Hew¥Tools¥Renesas¥Sh¥9\_4\_0¥binを設定しています。

3. 3. 2 コマンドプロンプトからの環境変数設定方法

ログオン後、コマンドプロンプトから環境変数を変更するためにはsetコマンドを使用してください。

(例)

```
set RSSITE=site1 (1)  
set RSUTYP=s (2)
```

- (1) RSSITEにサイト名“site1”を設定します。
- (2) RSUTYPに使用者種別“s”（システム）を設定します。

### 3. 4 RPDPユーザーアカウントの登録

RPDPを一般ユーザログインで使用するためには、専用のグループ（RPDPS10Usersグループ）に所属したアカウントでログオンしてください。新規にRPDPS10Usersに所属するアカウントを作成することも、既存のアカウントをRPDPS10Usersに所属させることもできます。

#### 3. 4. 1 新規アカウントの登録

新規にRPDPS10Usersに所属するアカウントを作成するための手順を以下に示します。

- (1) Administratorでログインします。
- (2) [マイコンピュータ] を右クリックし [管理(G)] を選択すると [コンピュータの管理] 画面が表示されます。
- (3) “ローカルユーザーとグループ” の “ユーザ” をクリックします。
- (4) 以下の手順で、専用のアカウントを登録します。
  - ① [操作(A)] メニューの [新しいユーザー(N)] を選択します。
  - ② [新しいユーザー] ダイアログボックスが表示されたら、ユーザ名等必要な項目を入力し、新しいアカウントを登録してください。ユーザ名やパスワードは任意です。
  - ③ 作成したユーザを選択し [所属するグループ] タブをクリックします。 **追加(D)** ボタンをクリックします。
  - ④ [グループの選択] 画面が表示されたら、 **詳細設定** ボタンをクリックします。
  - ⑤ さらに [グループの選択] 画面が表示されるので **(今すぐ) 検索** ボタンをクリックし、検索結果欄の “RPDPS10Users” をダブルクリックし追加します。  
“RPDPS10Users” グループは、S10V RPDPインストール時に自動的に登録されています。
  - ⑥ **OK** ボタンをクリックし終了します。

3. 4. 2 既存のアカウントの所属するグループにRPDPS10Usersグループを追加

既存のアカウントをRPDPS10Usersに所属させるための手順を以下に示します。

- (1) Administratorでログインします。
- (2) [マイコンピュータ] を右クリックし [管理(G)] を選択すると [コンピュータの管理] 画面が表示されます。
- (3) “ローカルユーザーとグループ” の “ユーザ” をクリックします。
- (4) 以下の手順で、アカウントの所属するグループに “RPDPS10Users” を追加します。
  - ① “RPDPS10Users” グループに所属させたいユーザを選択します。
  - ② [所属するグループ] タブをクリックします。  ボタンをクリックします。
  - ③ [グループの選択] 画面が表示されたら、  ボタンをクリックします。
  - ④ さらに [グループの選択] 画面が表示されるので  ボタンをクリックし、検索結果欄の “RPDPS10Users” をダブルクリックし追加します。

“RPDPS10Users” グループは、S10V RPDPインストール時に自動的に登録されています。
  - ⑤  ボタンをクリックし終了します。

### 通 知

RPDPを使用する際は、複数のユーザがログオンした状態で使用しないでください。  
複数のユーザを使う場合は、必ず使用中のユーザをログオフしてから、別のユーザのログオンを実行してください。  
複数ユーザがログオンした状態では、RPDPのコマンドが正しく実行されません。

## 第4章 コンパイラ

この章では、PCs用Cコンパイラおよびアセンブラの詳細について説明します。コマンドリファレンスについては、「第2編 コマンドリファレンス」を参照してください。

OSがMicrosoft® Windows® 2000およびWindows® XPの開発系マシンで、コンパイラ・アセンブラを使用する場合には、「日立マイクロコンピュータ開発環境システムSuperH RISC engine C/C++コンパイラパッケージ Ver.7」を、OSがMicrosoft® Windows® 7 (32bit) およびWindows® 10 (32bit) の開発系マシンでコンパイラ・アセンブラを使用する場合、「日立マイクロコンピュータ開発環境システムSuperH RISC engine C/C++コンパイラパッケージVer.9」（以下、shcコンパイラと表記します）を使用することを前提としています。

### 4. 1 Cコンパイラオプション詳細

以下にshcを使用したコンパイルの方法と、コンパイル時の注意点を示します。

shcの仕様の詳細は、shcコンパイラ付属のマニュアルを参照してください。

- コマンド形式

```
shc [△<オプション>...][△<ファイル名>[△<オプション>...]...]
```

（例）shc△test1.c △test2.c [Enter]

- 環境変数の設定

shc使用時には表 1 - 9 に示す環境変数を設定してください。

表 1-9 設定環境変数

環境変数	設定内容
path	<p>環境変数pathにインストールしたコンパイラパッケージの実行ファイルの格納ディレクトリを追加してください。</p> <p>コンパイラ (shc)、アセンブラ (asmsh)、最適化リンケージエディタ (optlnk) のパスの設定が必要です。</p> <p>この環境変数の指定は必須です。</p> <p>指定フォーマット： path=&lt;実行ファイルパス名&gt;[;&lt;既存パス名&gt;;...]</p>
SHC_LIB	<p>コンパイラのロードモジュールおよびシステムインクルードファイルを格納したディレクトリを指定してください。</p> <p>この環境変数の指定は必須です。</p> <p>指定フォーマット： set SHC_LIB=&lt;実行ファイルパス名&gt;</p>
SHCPU	<p>対象とするCPU種別を指定します。</p> <p>このシステムでは、SHCPU=SH4を指定してください。</p> <p>この指定を省略した場合、CPU種別はSH1として扱われるので注意してください。</p> <p>CPU種別は-cpuオプションでも指定することができます。</p> <p>指定フォーマット： set SHCPU=&lt;CPU&gt;</p>
SHC_INC	<p>コンパイラのインクルードファイル格納ディレクトリを指定してください。システムインクルードファイルの検索順序は、includeオプション指定ディレクトリ、SHC_INC指定ディレクトリ、システムディレクトリ (SHC_LIB) となります。</p> <p>ユーザインクルードファイルの検索順序は、カレントディレクトリ、includeオプション指定ディレクトリ、SHC_INC指定ディレクトリとなります。</p> <p>指定フォーマット： set SHC_INC=&lt;インクルードパス名&gt; [;&lt;インクルードパス名&gt;;...]</p>
HLNK_DIR	<p>最適化リンケージエディタの入力ファイル格納ディレクトリを指定します。</p> <p>Inputオプション、libraryオプションで指定したファイルの検索順序は、カレントディレクトリ、HLNK_DIR指定ディレクトリになります。</p> <p>ローダのライブラリサーチパスもHLNK_DIRの設定に従います。</p> <p>指定フォーマット： set HLNK_DIR=&lt;入力ファイルパス名&gt;[;&lt;入力ファイルパス名&gt;;...]</p>
SHC_TMP	<p>コンパイラがテンポラリファイルを作成するディレクトリを設定します。</p> <p>set SHC_TMP=&lt;ディレクトリ&gt;</p>
HLNK_TMP	<p>リンケージエディタがテンポラリファイルを作成するディレクトリを設定します。</p> <p>set HLNK_TMP=&lt;ディレクトリ&gt;</p>

## 4. 2 コンパイル時の注意点

## 4. 2. 1 shcコマンドを使用してコンパイルする場合

## ● 浮動小数点数の扱い

shcでは浮動小数点数の非正規化数と丸めの扱いをコンパイルオプションで制御することができます。

ただし、それぞれの扱いによってロード時にリンクする標準ライブラリが異なりますので注意してください。

以下に非正規化数の扱いと丸め方を制御するオプションと対応する標準ライブラリを示します（ロード時にライブラリを指定しないとローダはlibsh4nbmdn.libをリンクします）。

libsh4nbmzz.libを使用する場合は、“rpd.h”をインクルードしてコンパイルを行ってください。“rpd.h”をインクルードしてコンパイルを行うことにより、ロード時にリンクするライブラリを誤った際にエラーメッセージが出力されるようになります。

出力されるメッセージについては「第2編 コマンドリファレンス」の「第3章 ローダ」svloadの<ライブラリの整合性チェック>を参照してください。

表 1-10 浮動小数点数の扱い制御オプション

	仕様	オプション	デフォルト
非正規化数の扱い	0として扱う	-denormalization=off	0として扱う
	非正規化数として扱う (*)	-denormalization=on	
結果の値の丸め方	有効数字を超える部分を切り捨てる	-round=zero	切り捨て
	有効数字を超える部分を四捨五入する	-round=nearest	

(\*) S10VのCPUであるSH4 (SH7751) は非正規化数をサポートしていないため、実行時には0として扱われます。

表 1-11 浮動小数点数の扱いと対応する標準ライブラリ

	-denormalization	-round	標準ライブラリ
指定オプション	off	zero	libsh4nbmzz.lib
	on	zero	—
	off	nearest	—
	on	nearest	libsh4nbmdn.lib

● コンパイルリストの生成と保存 (shc)

タスクの使用するスタックサイズの算出等で必要となるため、コンパイルリストを生成して保存しておいてください。コンパイルリストを生成するためには、以下に示すオプションを指定します。

-listfileオプションは、コンパイルするCソースファイルより前に指定してください。

Cソースファイルより後ろに指定した場合、最後の1ファイルだけコンパイルリストが生成されます。

■ コンパイルリスト生成指定

```
-listfile [= <リストファイル名>] -show=source, object
```

リストファイル名の指定を省略した場合、ソースファイル名と同じファイル名に拡張子“.lst”を付加したファイルを生成します。

(例)

```
◆ shc △-listfile △test1.c △test2.c [Enter]
```

「test1.c」、 「test2.c」ともlistfileオプションが有効となります。

```
◆ shc △test1.c △test2.c △-listfile [Enter]
```

listfileオプションは「test2.c」だけに対して有効になります。

## 4. 3 mcc68kとshcの相違点

## 4. 3. 1 コマンド行オプション

表 1-12にmcc68kとshcのコマンド行オプションの比較を示します。

表 1-12 mcc68kとshcのコマンド行オプション比較 (1/2)

書式		SHC Version		意味
mcc68k	SHC	V7	V9	
-c	-Code = { <u>Machinecode</u> }	○	○	リンクをしません。オブジェクトモジュールを生成します。
-Dname -Dname=def	-DEFine = <sub>[,...] <sub> : <マクロ名> [= <文字列>]	○	○	nameを定義します。 nameをdefに定義します。
-E	—	—	—	コンパイルしません。プリプロセッサの結果を標準出力に出力します。
-g	-DEBbug	○	○	デバッグ情報を生成します。
-Idir	-Include = <パス名>[,...]	○	○	インクルードファイルのサーチディレクトリを追加します。
-O -On	-OPTimize = { 0   1   Debug_only }	○	●	最適化レベルを設定します。 <mcc68k> n=b, c, e, g, I, j, I, R, r, s, t <shc V7、V9> optimize=0 : 最適化なし optimize=1 : 最適化あり -Speed、-NOSpeed、-Sizeで最適化の方法を選択できます。 <shc V9> optimize=Debug_only :
	-SPeed	○	○	文単位の削除に関する最適化も完全に抑止
	-NOSPeed	○	○	し、ローカル変数の情報を常に参照できるようになります。
	-SIze	○	○	
-P	-PREProcessor [= <ファイル名>]	○	○	<mcc68k> プリプロセッサだけ実行し、結果を.iファイルに格納します。 <shc> プリプロセッサだけ実行し、結果を.pファイルに格納します。
-S	-Code = { Asmcode }	○	○	アセンブラソースを生成します。 アセンブラ、リンカを起動しません。
-Uname	—	—	—	nameを未定義にします。

shcのオプションの表記で、英大文字は短縮形指定時の文字を示します。  
又、下線部 ( ) は省略時解釈を示します。

## V7、V9凡例

- : 対応するオプションあり
- : V9で変更あり
- : 対応するオプションなし

表 1-12 mcc68kとshcのコマンド行オプション比較 (2/2)

書式		SHC Version		意味
mcc68k	SHC	V7	V9	
—	-Listfile -SHow =Souce,OBJectで代用可	○	○	アセンブラソースにソースファイルの行をいれます。
—	-Endian=Big	○	○	Big-endianモードでコンパイルします。 (デフォルトはbig)
—	-Endian=Little	○	○	Little-endianモードでコンパイルします。
—	-Sjis(デフォルト)	○	○	漢字 (シフトJIS) をサポートします。 K&R仕様時だけ指定できます。
—	-Show=length=n	○	○	ソースリストの1ページの行数を指定します。
—	-Listfile -Listfile=filename	○	○	ソースリストを表示します。
デフォルト でANSI準拠	デフォルトでANSI準拠	○	○	ANSI Standard Cに適合したプログラムだけコンパイルします。

shcのオプションの表記で、英大文字は短縮形指定時の文字を示します。  
又、下線部 ( ) は省略時解釈を示します。

V7、V9凡例

- : 対応するオプションあり
- : V9で変更あり
- : 対応するオプションなし

表 1-13にshcのオプションの一覧を示します。

表 1-13 shcのオプション一覧 (1/2)

項目	指定形式	意味
マイコン/動作モード	-CPu={ SH4 }	作成するオブジェクトプログラムのマイコン種別を指定します。
リストファイル	-Listfile [= <ファイル名>]	リストファイルの出力有無を指定します。
リスト内容と形式	-SHow = <sub>[,...] <sub>: {Source NOSource  Object NOObject  Statistics NOSTatistics  Include NOInclude  Expansion NOExpansion}	コンパイラが出力するリストの内容とその形式、および出力の解除を指定します。 Source : ソースプログラムリストの有無。 Object : オブジェクトプログラムリストの有無。 Statistics : 統計情報リストの有無。 Include : インクルードファイル展開した後のソースプログラムリストの有無。 Expansion : マクロ展開した後のソースプログラムリストの有無。
非正規化数の扱い	-DENormalize={ OFF   ON }	offを指定した場合、非正規化数を0として扱います。 onを指定した場合、非正規化数を非正規化数として扱います。 このオプションの省略時解釈は、denormalize=offです。
丸め方式	-Round={ Zero   Nearest }	zeroを指定した場合、Round to Zeroで丸めます。 nearestを指定した場合、Round to Nearestで丸めます。 このオプションの省略時解釈は、round=zeroです。
FPSCRレジスタの切り替え	-FPScr={ Safe   Aggressive }	FPSCRレジスタの精度モードを関数呼び出し前後で保証するかどうか指定します。 Safeを指定した場合、関数呼び出しから戻ったときのFPSCRレジスタの精度モードは常に単精度モードになります。 Aggressiveを指定した場合、関数呼び出しから戻ったときのFPSCRレジスタの精度モードの値は保証しません。 このオプションの省略時解釈は、fpscr=aggressiveです。
デフォルトインクルードファイル	-PREInclude=<ファイル名>[,...]	指定したファイルの内容をコンパイル単位の先頭に取り込みます。 ファイル名が複数ある場合にはカンマ (,) で区切って指定することができます。

表 1-13 shcのオプション一覧 (2/2)

項目	指定形式	意味
オブジェクト形式	-Code={Machinecode Asmcode}	machinecodeを指定した場合、リロケータブルオブジェクト（機械語）プログラムを出力します。 asmcodeを指定した場合、アセンブリプログラムを出力します。
マクロ名の定義	-DEFine=<sub>[,…] <sub> : <マクロ名>[=<文字列>]	ソースファイル内で記述する#defineと同等の効果を得ます。 <マクロ名>=<文字列>と記述することで<文字列>をマクロ名として定義できます。
デバッグ情報	-DEBug	デバッグ情報を生成します。
インクルード ファイルフォルダ	-Include=<パス名>[,…]	インクルードファイルの存在するパス名を指定します。
最適化レベル	-OPTimize={0 1 Debug_only}	オブジェクトプログラムの最適化レベルを指定します。 0を指定した場合、オブジェクトプログラムの最適化を行いません。 1を指定した場合、最適化を行います。 Debug_onlyを指定した場合、文単位の削除に関する最適化も完全に抑止し、ローカル変数の情報を常に参照できるようになります。オブジェクト性能が低下する可能性がありますので、デバッグ時に一時的に使用することを推奨します。 このオプションの省略時解釈は、optimize=1です。
プリプロセッサ展開	-PREProcessor=[<ファイル名>]	プリプロセッサ展開後のソースプログラムを出力します。 <ファイル名>を指定しない場合は、ソースファイル名と同じファイル名で拡張子が「p」のファイルが作成されます。 (入力ソースファイルがCプログラムの場合)

## 4. 3. 2 言語仕様相違点

表 1-14にmcc68kとshcの言語仕様の相違点と移行時の注意事項を示します。

比較は、仕様上相違のあるものについてだけ示します。また、mcc68k、shc特有の拡張機能についても省略します。mcc68k特有の拡張機能を使用していたプログラムは見直しが必要です。

表 1-14 言語仕様の比較

比較項目	mcc68k	shc	移行時の注意事項
識別子の有効文字数 (外部)	510文字	8189文字	
識別子の有効文字数 (内部)	512文字	8191文字	
アライン	2バイトアライン	ナチュラルアライン ただしdoubleは4バイト アライン	構造体のサイズ、配置を定 数で意識しているプログラ ムは修正してください。

## 4. 3. 3 関数の呼び出し規約

関数の呼び出し規約は、スタート ボタンから下記方法により、マニュアルを起動し、SuperH RISC engine C/C++コンパイラ、アセンブラ、最適化リンケージエディタユーザマニュアルの“プログラミング”を参照してください。

## (1) SHC Ver.7のマニュアル起動方法

- ① [Hitachi Embedded Workshop2] -  
[Online Manuals [SuperH] - [Japanese (Ver-Rev) ] ]
- ② [Renesas High-performance Embedded Workshop] -  
[Online Manuals [SuperH] - [Japanese (Ver-Rev) ] ]

## (2) SHC Ver.9のマニュアル起動方法

- ① [Renesas] - [High-performance Embedded Workshop] - [Manual Navigator]

## 第5章 プログラミングコマンド

### 5.1 プログラミングコマンドについて

S10Vのプログラミングコマンドでは、ライブラリアン、リンカはそれぞれoptlnkとsvloadを用意します。optlnkについては [スタート] メニューから下記方法により、マニュアルを起動し、SuperH RISC engine C/C++コンパイラ、アセンブラ、最適化リンカージェディタユーザマニュアルの“最適化リンカージェディタ操作方法”を、svloadについては「第8章 ローダ」を参照してください。

#### (1) SHC Ver.7のマニュアル起動方法

- ① [Hitachi Embedded Workshop2] –  
[Online Manuals [SuperH] – [Japanese (Ver-Rev) ] ]
- ② [Renesas High-performance Embedded Workshop] –  
[Online Manuals [SuperH] – [Japanese (Ver-Rev) ] ]

#### (2) SHC Ver.9のマニュアル起動方法

- ① [Renesas] – [High-performance Embedded Workshop] – [Manual Navigator]

#### ■ optlnkオプション一覧

項目	指定形式	意味
ライブラリファイル	-library=<ファイル名>[,…]	ライブラリファイルを指定します。複数ある場合にはカンマ (,) で区切って指定します。
プレリンカ	-noprelink	プレリンカの起動を抑止します。
出力形式	-form={absolute   relocate}	出力形式を指定します。 absoluteアブソリュートファイルを出力します。 relocateリロケータブルファイルを出力します。
出力ファイル	-output=[<サブオプション>] <サブオプション>:<ファイル名>	出力ファイル名を指定します。
リストファイル	-list=[<ファイル名>]	リストファイル出力およびリストファイル名を指定します。
リスト内容	-show=[<sub>] <sub>: { symbol }	モジュール内シンボル名一覧を出力します。
スタック情報ファイル	-stack	スタック使用量情報ファイルを出力します。
コピーライト	-nologo	nologoオプション指定時はコピーライト表示出力を抑止します。

## 第6章 システムジェネレーション

### 6.1 概要

#### 6.1.1 システムジェネレーションの目的

S10Vのリアルタイムプログラムの開発においてユーザのシステム形態、運用形態に合わせてシステムを作成する作業をシステムジェネレーションと呼びます。

このシステムジェネレーションを支援するツールが、システムジェネレーション用コマンドです。これによりPCsシステム導入時において開発系マシン上でPCsのリアルタイムプログラムを開発できる環境を構築できます。さらに、システム構築後のPCsの拡張が容易に行えます。

#### 6.1.2 システムジェネレーションの特長

システムジェネレーションは、次に示す特長を持っています。

- システム（サイト）構築環境の作成

S10Vの開発環境（サイト）の枠組みを生成し、サイト管理情報に登録します。

- 構成定義情報の生成

PCsを構築するために必要な定義情報をサイト単位に設定し、開発環境を構築します。

- 構成定義情報の表示

定義済みのPCsの定義情報を確認するための機能です。

- 既存サイトの複製を生成

PCsの構築手順を簡略化するために、既存サイトの複製を生成する機能です。複製の生成後、構成定義情報を変更し、新しいサイトを構築することができます。

- サイトの削除

不要となったサイトを削除する機能です。

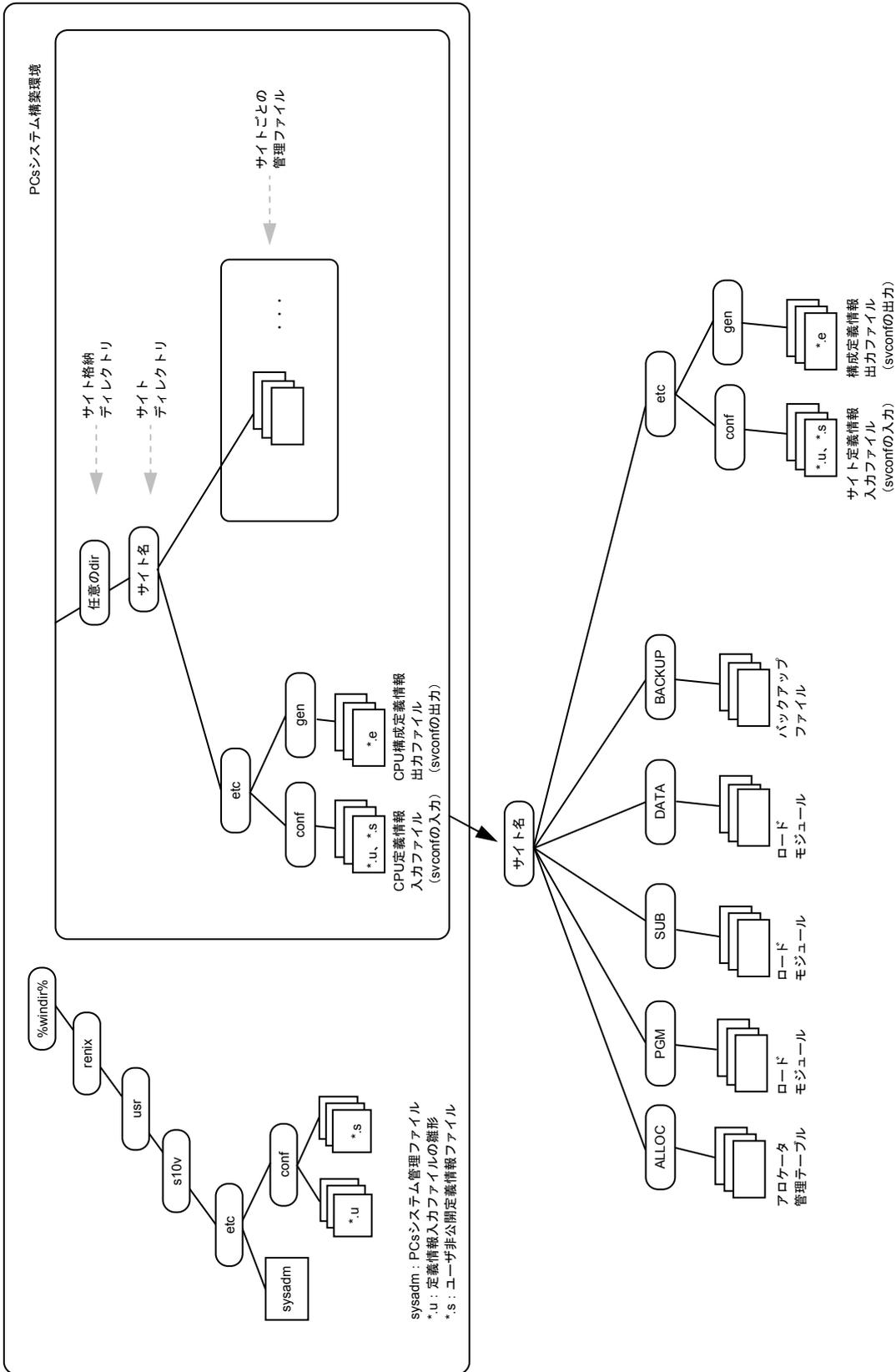


図1-15 PCsシステム全体構成

## 6. 1. 3 複数のPCsにおけるサイト管理方式

RPDPにより、1つの開発系マシンの環境下で複数のPCsのリアルタイムプログラム開発環境を管理できます。

RPDPは、管理対象のPCsに対して“サイト名称”と呼ぶ名称を付けて管理します。

## ● サイト名称

PCsに対し1つのサイト名称を付けます。サイト名称の規定を以下に示します。

- ・文字数は、1～14文字です。
- ・使用できる文字は、半角英数字、“\_”です。ただし、先頭は英字だけとします。

## ● サイト管理方式

各サイトの情報は、開発系マシンのファイルシステム上で次のように管理されます。

RPDPに関するディレクトリ構成を機種別に図1-16に示します。

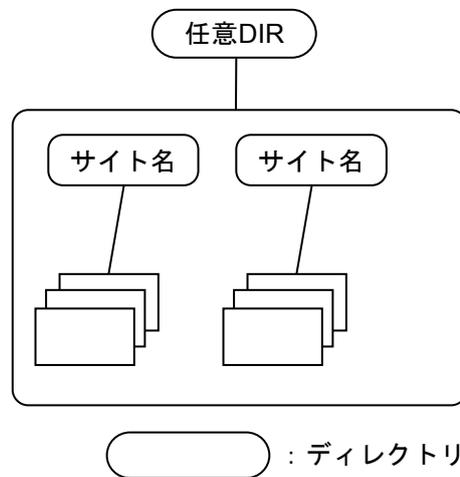


図1-16 サイトディレクトリ構成

システムジェネレーションコマンドは、以下のディレクトリ構成を開発マシン上に作成します。

- ・svgenコマンドに指定する、ユーザ定義ファイル中のSUBSYS\_DIR識別子で指定した、任意のディレクトリの下に各サイト用のディレクトリを作成します。
- ・各サイト用のディレクトリ下には、RPDPで使用するディレクトリ、ファイルを作成します。

それには、PCsのシステム構成に対応した情報が含まれます。

6. 2 システムジェネレーションコマンドとシステム生成の流れ

(1) コマンドの種別

RPDPのシステムジェネレーションコマンドを表1-15に示します。

コマンドはadministratorで実行してください。

ログオン中のユーザからadministratorに切り替える際は、必ず一度ログオフしてからadministratorにログオンしてください。

システムジェネレーション終了後も同様に、administratorをログオフし元のユーザにログオンしなおしてください。

※ログオンしたままユーザの切り替えを行うとRPDPのコマンドが正しく実行されません。

表1-15 コマンドの種類と機能

コマンド	機能	内容
svgen	PCs構築環境の作成	開発系マシン上にPCsのサイトを管理するために必要なディレクトリやサイトを構築します。
svconf	構成定義情報の生成	サイト構成定義情報入力ファイルの内容を基に構成定義情報出力ファイルを生成します。
svshconf	構成定義情報の表示	構築済みサイトの構成定義情報を表示します。
svsitecp	既存サイトの複製を生成	指定サイトのサイト環境の複製を生成し、新規サイトを生成します。 ただし、構成定義情報の複製は行いません。
svsitedel	サイトの削除	指定サイトのサイト環境ファイルを開発系マシンから削除します。

(2) ジェネレーションの流れ

システムジェネレーション作業は、新規と改造の2つに分けられます。

新規：管理対象となるPCsシステムを新規に生成する作業です。

改造：生成済みPCsシステムの定義情報を変更し、システムの改造を行う作業です。

(3) PCsシステムのシステム構成に関する新規構築手順

PCsシステムのシステム構成に関する新規構築手順を図1-17に示します。

手順番号	作業の流れ	説明
1	▽ エディタを使用してサイト構築環境定義ファイルを作成	サイト構築環境を作成するために、サイト構築環境定義ファイルを作成してください。 (「6. 3. 1 ユーザ設定定義情報」参照)
2	svgenを使用してシステム(サイト)構築環境を作成	svgenコマンドを実行するとサイト構築環境定義ファイルを基に、開発系に必要なディレクトリやファイルを作成します。
3	エディタを使用して各種サイト構成定義情報ファイルを作成	PCsシステムを構築するために、サイト構成定義情報ファイルを作成します。 (「6. 3. 3 ネットワーク定義情報の内容」、「6. 3. 4 サイト定義情報」参照)
4	svconfを使用してコントローラシステム環境を構築 △	svconf sitenameを実行し、サイトを構築してください。

図1-17 PCsシステム新規構築手順

(4) PCsシステムのシステム構成に関する改造手順

PCsシステムのシステム構成に関する改造手順を図1-18に示します。

(a) PCsシステム改造

手順番号	作業の流れ	説明
1	▽ エディタを使用して各種サイト構成定義情報ファイルを修正	PCsシステムを再構築するために、サイト構成定義情報入力ファイルを修正します。 (「6. 3. 3 ネットワーク定義情報の内容」、「6. 3. 4 サイト定義情報」参照)
2	svconfを使用してコントローラシステム環境を再構築 △	svconf sitenameを実行し、サイトを再構築してください。

図1-18 PCsシステム改造手順

(5) 他のサイトをコピーしてサイト構築

(a) 同一開発系マシン内のサイトコピー

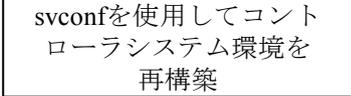
手順番号	作業の流れ	説明
1		既存サイトをコピーします。
2		コントローラシステムを再構築するために、サイト構成定義情報ファイルを修正します。 (「6. 3. 3 ネットワーク定義情報の内容」、「6. 3. 4 サイト定義情報」参照)
3	 <p style="text-align: center;">△</p>	svconf sitenameを実行し、サイトを再構築してください。

図 1-19 同一開発系マシン内のサイトコピー手順

(6) 開発系マシンのOS変更に伴うサイト移行手順

開発系マシンのOSをWindows® 2000、Windows® XPから、Windows® 7 (32bit)、Windows® 10 (32bit) に移行する場合、移行先の開発系マシンにおいてサイト構築をやり直す必要があります。開発系マシンのOS変更に伴うサイト移行手順を図1-20に示します。

手順番号	作業の流れ	説明
1	<p style="text-align: center;">▽</p> <p>【移行元開発系マシンで実施】</p> <p>サイト構築環境定義ファイル、サイト構築環境をバックアップ</p>	<p>サイト構築環境定義ファイル (svgenコマンドの入力ファイル) とサイト構築環境の各種サイト構成定義情報ファイル (サイトのetc¥conf下の全てのファイル) をバックアップします。</p>
2	<p>【移行先開発系マシンで実施】</p> <p>エディタを使用してサイト構築環境定義ファイルを修正</p>	<p>サイト構築環境定義ファイルの「サイト格納ディレクトリ」を移行先開発系マシンで変更が必要な場合は、修正します。</p> <p>変更が必要ない場合は、そのまま使用します。</p>
3	<p>svgenを使用してサイト構築環境を生成</p>	<p>手順番号2で修正したサイト構築環境定義ファイルを指定してsvgenコマンドを起動し、サイト構築環境を生成します。</p>
4	<p>バックアップした移行元PCの各種サイト構成定義情報をコピー</p>	<p>手順番号1でバックアップした移行元開発系マシンの各種サイト構成定義情報ファイル (サイトのetc¥conf下のファイル) を移行先開発系マシンのサイト構成定義情報ファイルへ上書きコピーします。</p>
5	<p>svconfを使用して構成定義情報を生成</p>	<p>svconf sitenameを実行し、構成定義情報を生成してください。</p>
6	<p>SHCコンパイラV9でコンパイル</p>	<p>移行元開発系マシンより移行先開発系マシンへ移動したソースファイルをSHCコンパイラV9でコンパイルし、オブジェクトファイルを生成してください。</p> <p>移行元開発系マシンのSHCコンパイラV7で作成したオブジェクトファイルは使用しないでください。</p>
7	<p>リアルタイムプログラムリソースの割り付け</p>	<p>手順番号6で生成したオブジェクトファイルを使用して、移行元開発系マシンと同じ手順で、アロケータ、ローダ、ビルダコマンドを実行し、バックアップファイルを生成してください。</p>
8	<p>移行後のバックアップファイルをダウンロード</p> <p style="text-align: center;">△</p>	<p>svrplコマンドを実行し、手順番号7で生成したバックアップファイルをPCsの主メモリへダウンロードしてください。</p>

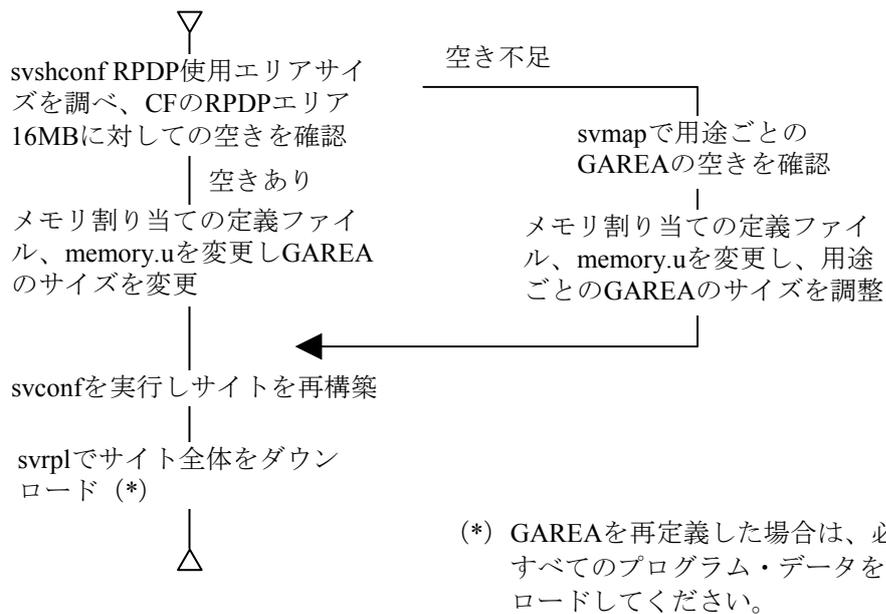
図1-20 開発系マシンのOS変更に伴うサイト移行手順

<領域分割に関する注意事項>

(1) GAREA不足時の拡張

S10V RPD Pのエリア管理では、タスクやGLBなどの用途ごとに使用できる物理メモリのサイズを、サイト構築時に用途ごとのGAREAとして割り当ててください。サイト構築後にGAREAが不足した場合は、物理メモリに空きがあればGAREAを拡張することができます。物理メモリに空きがない場合は、他の用途に割り当てたGAREAの容量を減らすなどの調整が必要です。

GAREAのサイズを変更した場合は、svrplでサイト全体を再ダウンロードしてください。サイト構築後にGAREAを拡張する手順を以下に示します。



(2) GAREAの断片化の防止

S10V RPD Pのエリア管理では、物理メモリと論理空間は1対1にマッピングされています。このため、タスクやGLBなどの用途ごとの論理空間に分割領域を確保できる範囲は、各論理空間の先頭からサイト構築時に定義したGAREAのサイズの範囲となります。したがって、GAREAの範囲内に分割領域を割り当てていない小さな空きエリアが多数存在しても、連続した空きがないと新しい分割領域を割り当てることはできません。

GAREA内の空きエリアの断片化を解消するためには、分割領域を切り直し再配置しなければなりません。分割領域を再配置するためには、分割領域内のプログラム・サブプログラムの再ロード、GLBの場合は細分割領域の再定義、初期値データの再ロードが必要です。さらに、GLBの分割領域を再配置した場合は、それを参照しているプログラム・サブプログラムの再ロード、GLBの場合は細分割領域の再定義、初期値データの再ロードが必要です。さらに、GLBの分割領域を再配置した場合は、それを参照しているプログラム・サブプログラムの再ロードも必要です。

このため、GAREA内の空きエリアには断片化が発生しないように使用してください。分割領域のサイズを変更して解放/再確保をするとGAREAの断片化が発生しやすくなります。断片化発生防止のために、分割領域の解放・再確保をしなくても良いように、分割領域は将来拡張分のサイズを含めて確保することを推奨します。

### 6. 3 PCsシステム定義情報

PCsシステムは、開発系マシンとPCs、およびそれらを接続するネットワークやPI/Oから構成されます。PCsにサイトを割り当てこれを構築します。サイト定義情報の内容を表1-16に示します。

表1-16 サイト定義情報の内容

種別	定義情報	管理単位
サイト構築環境定義情報	開発系で管理するPCsシステムの定義情報であり、サイト構築環境定義ファイルで指定する、サイト格納ディレクトリ、サイトディレクトリが該当します。	開発系で管理するPCs全体が対象

#### 6. 3. 1 ユーザ設定定義情報

ユーザ設定定義情報は、PCs構築時にユーザが設定しなければならない定義情報です。ユーザ設定定義情報には大別して下記2つの項目があります。

- サイト構築環境定義情報

svgenコマンドでサイト構築環境を生成するために必要な情報であり、ユーザが任意のファイルに記述します。

- サイト構成定義情報

サイト構成定義情報は、svconfコマンドでサイト環境下にPCsシステムのリアルタイム開発環境を生成するために必要な情報であり、svgenコマンドでのサイト環境作成時にサイト格納ディレクトリの下に雛形ファイルが生成されています。

##### (1) サイト構築環境定義情報

svgenコマンドでサイト構築環境を生成するために必要な情報であり、ユーザが任意のファイルに記述します。

- 設定項目

項目	設定内容
サイト格納ディレクトリ	サイトディレクトリを格納するディレクトリのパスを指定します。 ドライブレターを含めたフルパスで指定してください。
サイト名	サイト名称を指定してください。

■ 記述形式

```
SUBSYS_DIR=△サイト格納ディレクトリ  
SITE_NAME=△サイト名
```

太字・斜体の部分はキーワードを表します。キーワードを含めて記述してください。  
△は1つ以上の空白またはタブを表します。

・ 定義例

以下にシステム定義例を示します。

```
SUBSYS_DIR= c:%sitedir  
SITE_NAME= pcs10
```

■ サイト名の規則

サイト名は、14文字以内とします。文字は、半角英数字または“\_”（アンダーライン）を使用できます。

(2) サイト構成定義情報

svconfコマンドでサイト環境を生成するために必要な情報であり、サイト構成定義情報ファイルに記述します。サイト構成定義情報ファイルは、svgenでのサイト構築環境生成時に、サイトディレクトリ下のetc%conf下に雛形ファイルが生成されています。

表1-17にサイト構成定義情報の一覧を示します。

詳細については、「6.3.3 ネットワーク定義情報の内容」、「6.3.4 サイト定義情報」参照してください。

表 1-17 ユーザ定義サイト構成定義情報設定項目

種別	定義情報	設定項目	識別子	入力属性 (範囲)	デフォルト値	入力ファイル名
ネットワーク 定義情報	アダプタ 情報	ネットワークアダプタの定 義情報ですが、固定値とな ります。	SLOT	10進数 (0固定)	—	adapter.u
	インタ フェース 起動情報	ネットワークアダプタのイ ンタフェース単位の定義情 報を設定してください。	STATION_NO IF_COUNT UNO IF_NUM IP_ADDR	10進数 (1固定) 10進数 (1固定) 10進数 (1固定) 10進数 (1固定) X1.X2.X3.X4形式のIP アドレス	— — — — —	
プロセスサ定 義情報	メモリ 情報	サブネットマスク	NETMASK	X1.X2.X3.X4形式のIP アドレス	—	memory.u
		ブロードキャストアドレス	BRD_ADD	X1.X2.X3.X4形式のIP アドレス	—	
		タスク用GAREAサイズ (KB)	TASKSZ	10進数 (16の倍数)	—	
		読み出し専用GLB用GAREAサイズ (KB)	GLBSZ	10進数 (16の倍数)	—	
		読み書き両用GLB用GAREAサイズ (KB)	GLBWSZ	10進数 (16の倍数)	—	
		サブプログラム用GAREAサイズ (KB)	SUBSZ	10進数 (16の倍数)	—	

<注意事項>

- S10V実機のIPアドレスはS10Vツールで設定してください。デフォルトのIPアドレスは、192.192.192.1で設定されています。  
また、192.168.0.1はR600用E-portアドレスのため使用できません。
- NCP-F、LANCP経由でRPDPコマンドは使用できません。RPDPコマンドは、CMU経由のEtherだけ使用できます。
- RPDPで定義可能なメモリサイズの合計は16MBまでです。

6. 3. 2 PCsシステム定義情報の内容

(1) PCsシステム定義情報

PCsシステム定義情報は、svgenコマンドのパラメータであるサイト構築環境定義ファイルを基に作成、更新されます。ユーザによりサイト構築環境定義ファイル内に設定された“サイト格納ディレクトリ名”、“サイトディレクトリ名”は、(%windir%\¥renix¥usr¥s10v¥etc下の) sysadmファイル内に格納されます。

【設定項目】

項目	設定内容
サイト格納ディレクトリ	サイトディレクトリを格納するディレクトリのパスを指定します。 ドライブレターを含めたフルパスで指定してください。
サイト名	サイト名称を指定してください。

【設定例】

開発系マシン上に下図に示すディレクトリ構成を作成する場合のサイト構築環境定義ファイルの設定例を以下に示します。

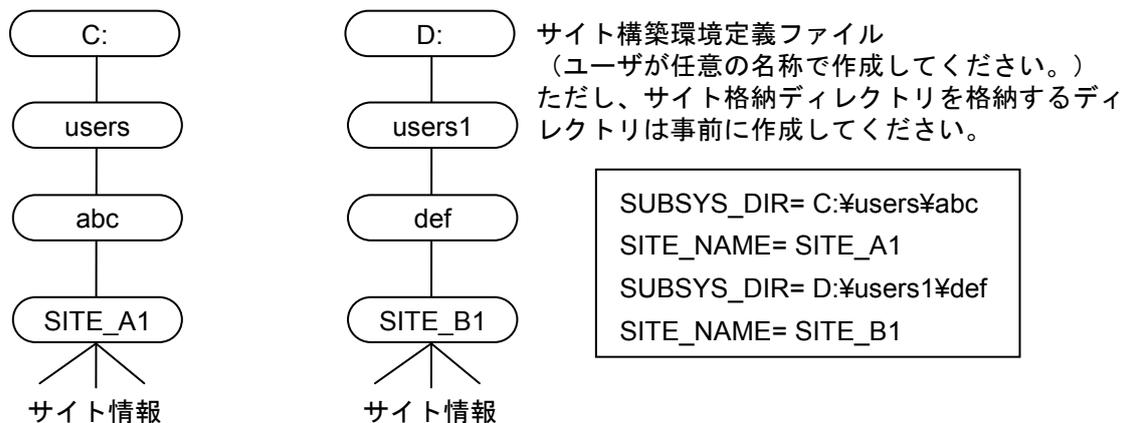


図 1-21 RPDPIに関するディレクトリ構成とサイト構築環境定義ファイル構成

## 6. 3. 3 ネットワーク定義情報の内容

## (1) アダプタ情報 (adapter.u)

CMUを定義する項目です。

## 【設定項目】

- SLOT (スロット番号)  
ネットワークアダプタを定義するスロット番号を設定します。  
CMUの場合、0固定となります。  
デフォルト値はないため必ず設定してください。
- STATION\_NO (ステーション番号)  
ハードウェアのステーション番号のロータリスイッチ情報を設定します。  
CMUの場合、1固定となります。  
デフォルト値はないため必ず設定してください。
- IF\_COUNT (インタフェース数)  
使用するインタフェース (チャンネル) 数を設定します。(デフォルト値は1)  
CMUの場合、1固定となります。

## (2) インタフェース起動情報 (adapter.u)

ネットワークアダプタのインタフェース単位に定義する項目です。

下記の設定項目を1組設定します。

## 【設定項目】

- UNO (UCB番号)  
インタフェース単位に定義するUCB番号を設定します。CMUの場合、1固定となります。  
デフォルト値はないため必ず設定してください。
- IF\_NUM (インタフェース番号)  
使用するインタフェース番号を設定します。(デフォルト値は1)  
CMUの場合、1固定となります。
- IP\_ADDR (IPアドレス)  
自IPアドレスをドット表記で設定します。  
デフォルト値はないため必ず設定してください。
- NETMASK (サブネットマスク)  
ネットワークをサブネットワークに分割するために、アドレスをどこまで占有するかをドット表記で設定します。  
デフォルト値はないため必ず設定してください。
- BRD\_ADD (ブロードキャストアドレス)  
ネットワークにブロードキャストを示すために使用するアドレスをドット表記で設定します。  
デフォルト値はないため必ず設定してください。

<サブネットワークを指定しない場合の設定例>

アドレス体系	IPアドレス	サブネットマスク	ブロードキャストアドレス
クラスA	0~126.X1.X2.X3	255. 0. 0. 0	0~126.255.255.255
クラスB	128~191.X1.X2.X3	255.255. 0. 0	128~191. X1.255.255
クラスC	192~223.X1.X2.X3	255.255.255. 0	192~223. X1. X2.255

X1、X2、X3は0~255

クラス別IPアドレスの説明

- ・クラスAの場合、0~126がネットワークアドレス部、X1.X2.X3がホストアドレス部となります。
- ・クラスBの場合、128~191.X1がネットワークアドレス部、X2.X3がホストアドレス部となります。
- ・クラスCの場合、192~223.X1.X2がネットワークアドレス部、X3がホストアドレス部となります。

#### 6. 3. 4 サイト定義情報

##### (1) メモリに関する情報

S10Vでは、“サイト格納ディレクトリ¥サイト名¥etc¥conf” 下のmemory.uファイル内に定義します。ここにはメモリの使用の内訳に関する情報を設定します。

##### 【設定項目】

下表に設定項目を示します。

表 1-18 メモリに関するサイト構築定義情報

種別	定義情報	設定項目	識別子	入力属性 (範囲)	デフォルト値	入力ファイル名
プロセスサ定義情報	メモリ情報 サイトが使用するメモリの情報を設定してください。	タスク用GAREAサイズ (KB)	TASKSZ	10進数 (16の倍数)	—	memory.u
		読み出し専用GLB用GAREAサイズ (KB)	GLBRSZ	10進数 (16の倍数)	—	
		読み書き両用GLB用GAREAサイズ (KB)	GLBRWSZ	10進数 (16の倍数)	—	
		サブプログラム用GAREAサイズ (KB)	SUBSZ	10進数 (16の倍数)	—	

(注) メモリ情報のサイズの合計は最大16MBです。

#### 6. 4 PCsシステム定義情報の表示

PCsシステム定義情報の表示機能は、svgen、svconfコマンドで生成したPCsシステム定義情報を表示します。

表示オペレーションおよび表示内容については、svshconfコマンドのリファレンスを参照してください。

## 6.5 PCsシステム環境の複製

### 6.5.1 複製の単位

PCsシステム環境の複製機能は、サイト単位でPCsシステム環境のコピーを生成します。

サイトをコピーしても構成定義情報はコピーされません。必ず定義情報を修正し、再構築してください。

サイトのコピーのオペレーションは、`svsitecp`コマンドのリファレンスを参照してください。

6.5.2 複製の範囲

以下にサイトの複製機能でコピーされる範囲を示します。

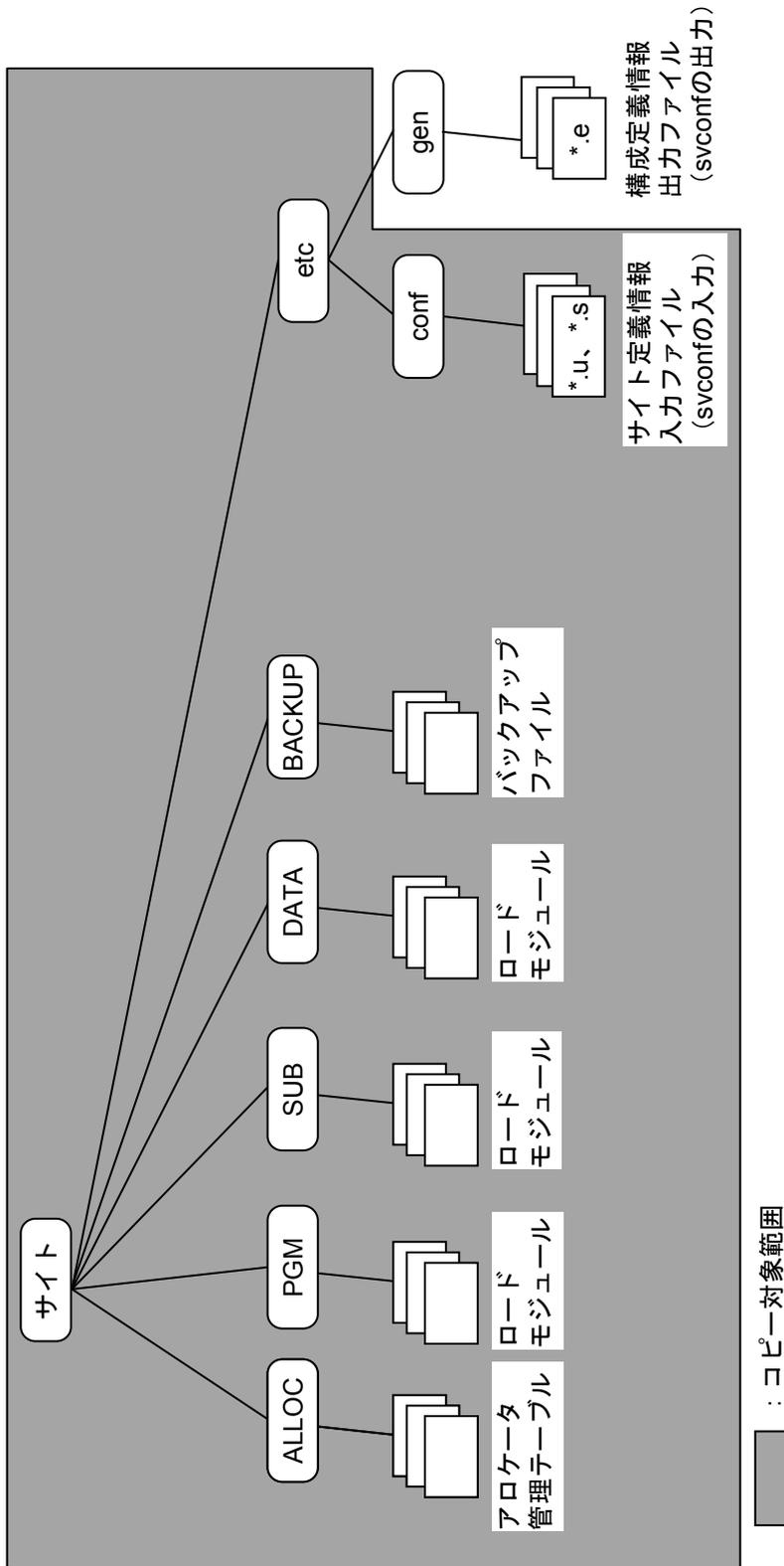


図1-22 サイト複製機能でコピーされる範囲

## 6. 6 PCsシステム環境の削除

PCsシステム環境の削除機能は、指定したサイト環境下のファイルを削除し、コントローラシステム管理ファイルから管理情報を削除します。

サイトの削除オペレーションは、`svsitedel`コマンドのリファレンスを参照してください。

## 第7章 アロケータ

### 7. 1 分割領域の確保と削除

#### 7. 1. 1 領域の分割の必要性

リアルタイムプログラムの開発に先立って、あらかじめリアルタイムプログラムが使用するタスク、サブプログラム、GLBなどの共有資源用の格納領域を確保（アロケート）しておく必要があります。

リアルタイムシステムでは処理の高速性を追求するため、さまざまな資源に対するアクセスをその資源の格納アドレスを使って行います。このため、プログラム実行時資源の格納アドレスが不定であってはならないため、アロケータで計算機内のエリアをユーザの指定領域に分割し、これが指定アドレスからずれないように管理しています。システム設計時にはこのシステムの適用対象から、どれくらいのデータが必要かを調べてGLBの大きさ、配置を決めてください。リアルタイムシステムでは多くの場合、個々のプログラム作りよりも、このデータに関する設計が重要でシステムの総合性能に大きく影響します。

割り当ては、次の2段階で行います。

- (1) タスク、サブプログラム、GLB用に分割領域（AREA）を定義します。
- (2) GLBについてはsvdfsを使用してAREAを分割し、細分割領域（SAREA）を定義できます。

アロケータで定義した分割領域、細分割領域については、その名前と属性、位置、サイズなどが領域管理情報に記録され、リアルタイムプログラム中からは、ここで定義した名前を用いてGLBなどの共有資源を参照したり、呼び出したりできます。

このように共有資源用の領域を断層的に確保するのは、共有資源用の領域が通常の資源とは別に確保されていて、いったん確保した分割領域を再定義した際に、なるべく他の分割領域を定義し直さなくても良いように配慮しているためです。

7. 1. 2 分割領域の確保

分割領域はその用途によって決められた大分割領域（GAREA）内に配置されます。

表 1-19に分割領域の用途と配置するGAREAの対応を示します。

定義する分割領域の用途は、分割領域の確保時にsvdfaコマンドのオプションで指定します。

表 1-19 分割領域の用途と配置するGAREAの対応

分割領域の用途	配置するGAREA	svdfaのオプション
タスク（プログラム）	\$TASK	-p
読み出し専用GLB用のデータ	\$GLBR	-gr
読み書き両用GLB用のデータ	\$GLBRW	-gi、-gw
サブプログラム	\$IRSUB	-s

分割領域を確保すると、指定サイズ分のエリアが大分割領域内に確保されます。ただし、確保した分割領域をダウンロードするまでは、PCsメモリには反映されません。

また、分割領域を確保すると、分割領域内に配置するリアルタイムリソースの初期値設定用のファイル（バックアップファイル）を生成します。

ただし、初期値なしGLBの分割領域はバックアップファイルを生成しません。

バックアップファイルの内容は0で初期化されています。

分割領域の確保に用いるコマンドを次に示します。

svdfa 分割領域（AREA）の確保

svdfs 細分割領域（SAREA）の確保

分割領域のレイアウトを図 1-23に示します。

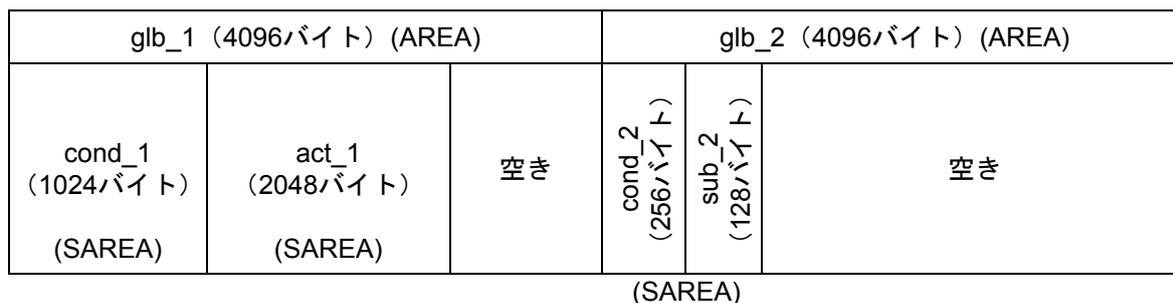


図 1-23 分割領域のレイアウト

上記レイアウトの確保実行例

```
#svdfa glb_1 4096 -gw
#svdfs glb_1 cond_1 1024
#svdfs glb_1 act_1 2048

#svdfa glb_2 4096 -gw
#svdfs glb_2 cond_2 256
#svdfs glb_2 sub_2 128
```

図1-23に基づいて分割領域を確保します。

```

$svdfa glb_1 4096 -gi
$svdfa glb_2 4096 -gi
$svdfs glb_1 cond_1 1024
$svdfs glb_1 act_1 2048
$svdfs glb_2 cond_2 256
$svdfs glb_2 sub_2 128
$svmap -g -a -e
** allocator map **
2002/11/19 09:15:56

site name = test2

< area >
garea/aname          raddr   size    laddr   kind  bkupfile
:
:
$GLBRW/glb_1        + s 00000000 00001000 50000000 glbi  glb_1.bkf
$GLBRW/glb_2        + s 00001000 00001000 50001000 glbi  glb_2.bkf
$GLBRW/              00002000 000fe000 50002000

< sarea >

:
:
garea/aname/sname          raddr   size    laddr
$GLBRW/glb_1/cond_1      + s 00000000 00000400 50000000
$GLBRW/glb_1/act_1      + s 00000400 00000800 50000400
$GLBRW/glb_1/           00000c00 00000400 50000c00
$GLBRW/glb_2/cond_2      + s 00000000 00000100 50001000
$GLBRW/glb_2/sub_2      + s 00000100 00000080 50001100
$GLBRW/glb_2/           00000180 00000e80 50001180
:
:
** map output end **
$

```

リアルタイムプログラムの中では、定義した細分割領域の名前、`cond_1`、`act_1`、`cond_2`、`sub_2`を使用することによって、これらの共有資源を利用できます。

```
$notepad sample.c
```

```
extern int cond_1_g[256];
extern int act_1_g[512];
extern char cond_2_g[256];
extern short sub_2_g[64];
main()
{
  short abc;
  cond_1_g[10] = 0;
  act_1_g[20] = 30;
  cond_2_g[255] = 'A';
  abc = sub_2_g[0];
}
```

なお、いったん確保した分割領域のサイズや属性を変更する場合、細分化して定義した細分割領域の合計より小さくならないように注意してください。

細分割領域の合計より小さい場合、細分割領域を指定分割領域内に確保できなくなります。

## 7. 1. 3 分割領域の削除

分割領域の削除に用いるコマンドを次に示します。

```
svdla    分割領域 (AREA) の削除
svdls    細分割領域 (SAREA) の削除
```

さきほど確保したGLB部のglb\_1およびglb\_2を削除する例を示します。削除しようとする分割領域にさらに細分割領域があれば、その細分割領域も同時に解放されます。glb\_2の場合、細分割領域のcond\_2、sub\_2を定義してありますが、glb\_2を削除すると、cond\_2、sub\_2も自動的に削除されます。

```
$svdls cond_1
$svdls act_1
$svdla glb_1
$svdla glb_2
$svmap -g
** allocator map **                                2002/04/07 15:19:13

site name = test_CP
:
< global, irglb >
:

** map output end **
$
```

## 7. 1. 4 GLB、VALの名前の付け方

GLB、VALはシステム内でユニークな名前にしてください。

- 名前の文字数 (バイト数) は、最大14文字
- 先頭文字は半角英字またはアンダーライン ( \_ )
- 2文字目以降は、半角英数字、アンダーラインの組み合わせ

ただし、これらの名前を利用するリアルタイムプログラムでは、使用するC言語に次のような名前の制約がありますので注意してください。

<C言語の制約>

- 名前の文字数 (バイト数) は、最大14文字
- 先頭文字は英字
- 2文字目以降は、半角英数字、アンダーラインの組み合わせ
- 名前の後に次のようなサフィックスを付加します。サフィックスの文字数は、名前の文字数に含みません。

GLBのとき \_g

VALのとき \_v

- GLB、VALの宣言は、必ずextern指定による外部変数の宣言としてください。

## 7. 2 バリュ (VAL) の登録、削除

アロケータは、バリュ (VAL) と呼ばれる全プログラム間共通の定数の登録、削除を行います。VALの登録、削除は、それぞれsvdfv、svdlvコマンドで行います。

## 第8章 ローダ

## 8.1 リンク・ロードとは

shcコマンドを用いて作成したオブジェクトモジュール(.objファイル)を、svloadコマンドを用いてリンクし(1つのプログラムにまとめ)、GLB、IRSUBなどのアロケータ管理情報を用いてロードし、ロードモジュールをバックアップファイルに書き込みます(図1-24参照)。

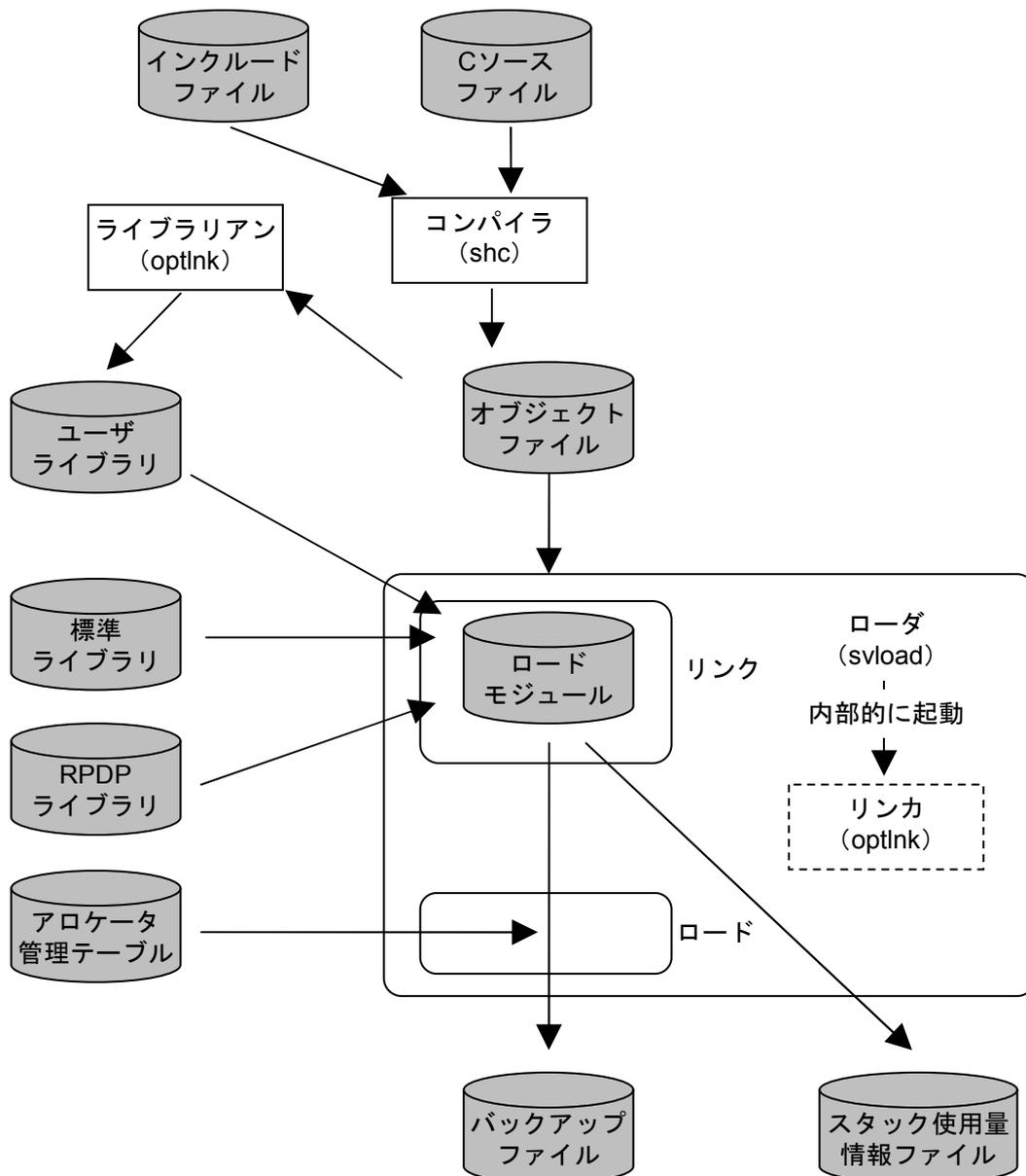


図1-24 ロードモジュール、バックアップファイルの作成

8. 2 ローダの動作環境

ローダに入力するプログラムは、ローダでの結合後のロードモジュールが、表1-20の条件を満たすようなプログラムとしてください。

表1-20 ロードモジュールの条件

オプション	ロードモジュール		
	TEXT	DATA	BSS
プログラム登録	>0	—	—
組み込みサブプログラムの登録	>0	—	— (*)
データの登録	—	>0	—

TEXT：実行できる部分 DATA：初期値ありデータ BSS：初期値なしエリア

—：サイズ=0、>0いずれでも処理できます。

>0：サイズ>0以外はエラーとします。

(\*) サブプログラムのBSS部は、書き込み禁止です。

サブプログラムは、BSSを持たないでください。

ローダが生成するロードモジュールの構成を図1-25に示します。

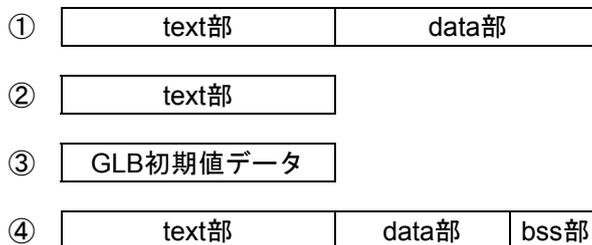


図1-25 ロードモジュールの構成

<図1-25の説明>

- ① text部とdata部を持つプログラムまたはサブプログラムのロードモジュールです。プログラムまたはサブプログラムとしてローディングできます。
- ② text部だけからなるプログラムまたはサブプログラムのロードモジュールです。①と同様にローディングします。
- ③ GLBの初期値設定プログラムのロードモジュールです。データとしてローディングできます。
- ④ text部、data部とbss部を持つプログラムのロードモジュールです。プログラムとしてローディングできます。サブプログラムには、bss部を持たせないでください。

## (1) ローダの処理

図1-25に示したロードモジュールのうち③、④の構成を例に取り、ローダのローディング処理について説明します。

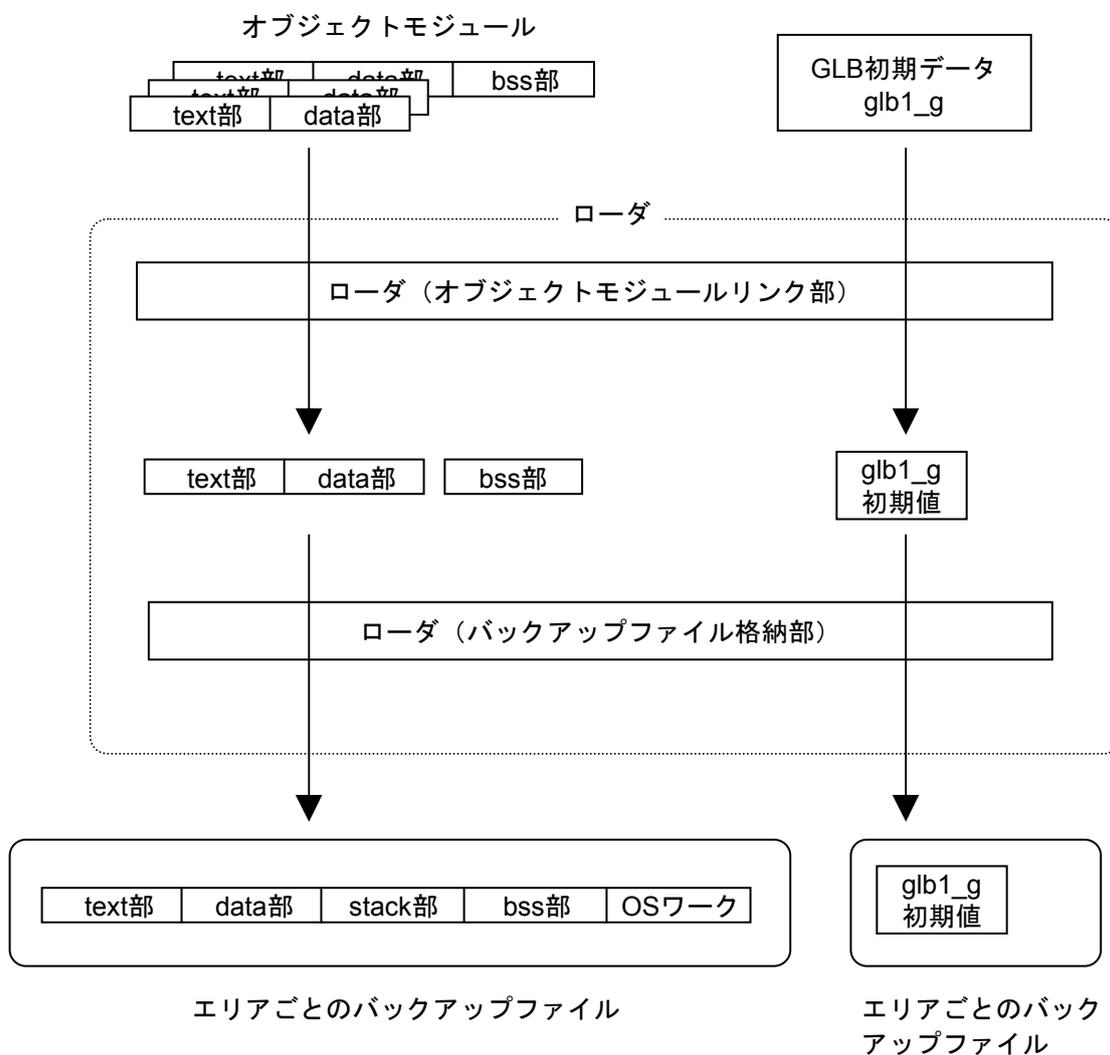


図1-26 ローディング処理

<図1-26の説明>

- ① グローバルの初期値データ部分はアロケータの管理するテーブルにsvdfsで登録した細分割領域に対応する個所にローディングされます。
- ② プログラムは実行モジュールとしてローダコマンドで指定した領域にローディングされます。ローダは、実行モジュールに指定されたスタック領域のエリアとOSワークを付加してバックアップファイルに格納します。

(2) 名称のユニーク性

system/user間でプログラム名称、サブプログラム名称、組み込みサブルーチン名称、グローバル名称、バリュ名称はユニークな名称にしてください。

(3) system/user外部参照チェック

システムからユーザの参照はできません。

ユーザからシステムのサブプログラムだけ参照できます。表1-21に参照可能な組み合わせを示します。

表1-21 外部参照の組み合わせ

参照 \ 被参照		サブプログラム		グローバル		バリュ	
		S	U	S	U	S	U
プログラム	S	○	×	○	×	○	×
	U	○	○	×	○	×	○
サブプログラム	S	○	×	○	×	○	×
	U	○	○	×	○	×	○
グローバル	S	○ (*1)	×	○ (*2)	×	○ (*3)	×
	U	○	○ (*1)	×	○ (*2)	×	○ (*3)

○ : 参照可 S : システム × : 参照不可 U : ユーザ

(\*1) グローバル内にIRSUB番号が埋め込まれます。

したがって、サブプログラムはbuildされているIRSUBでなければなりません。

(\*2) グローバル内に被参照グローバルのアドレスが埋め込まれます。

(\*3) グローバル内にVAL値が埋め込まれます。

### 8. 3 ライブラリのサーチパス

ローダのライブラリのサーチパス (-lオプションで指定されたライブラリのサーチ順) は、shcコンパイラパッケージの最適化リンケージエディタの入力ファイルの検索順序に従います。

最適化リンケージエディタの入力ファイルの検索順は下記となります。

- (1) カレントディレクトリ
- (2) 環境変数HLNK\_DIRで指定されたディレクトリ

環境変数HLNK\_DIRには複数のパスが設定できます。

複数のパスを設定する場合は、セミコロンで区切って指定してください。

### 8. 4 リンク・ロード時の注意事項

リアルタイムプログラムをリンク、ロードする際には、プログラム中で使用している、GLB、VALの割り当てを済ませておいてください。また、IRSUBを使用しているときにはIRSUBのビルドまで済ませておいてください。

## 第9章 ビルダ

## 9. 1 タスクの登録と削除

## 9. 1. 1 タスクとは

ローダ (svload) で作成したロードモジュールは、ビルダのsvctaskによってタスクとして生成し、稼働の準備を整えます。svctaskは、ロードモジュールに対応するタスク名、タスク番号、タスクの実行レベルなどを作り、OSが管理しているタスク情報に登録します。

## 9. 1. 2 タスクの登録

プログラムをタスクとして登録する例を以下に示します。

```

$ svdfa areal 0x3000 -p
$ shc sample.c
$ svload +P -w 1024 1024 -a areal -o sample sample.obj
$ svmap -p -t
** allocator map **                                     2002/04/07 15:20:38

site name = test_CP

< task-program >
tn  tname          tnox rmtn lvl  sp    pname          st mtn  texttop
lastaddr tsize  dsize  ssize (part ) bsize  extra  oswork
:
                                sample          + s ls 0001 30000000
30003000 000068 000000 000400(000400) 000000 000000 001000
:
** map output end **
$ svctask sample sample_1 10 -l 25
$ svmap sample_1 -t
** allocator map **                                     2002/04/07 15:21:38

site name = test_CP

< task-program >
tn  tname          tnox rmtn lvl  sp    pname          st mtn  texttop
lastaddr tsize  dsize  ssize (part ) bsize  extra  oswork
:
10 sample_1          + u 000a 0001 19 30002000 sample          + s cs 0001 30000000
30003000 000068 000000 000400(000400) 000000 000000 001000
:
** map output end **
$

```

svctaskで指定しているsampleは、タスクの資源となるロードモジュールの名前です。タスクの起動時には、ここで指定したロードモジュールがタスクのメインプログラムとして実行されます。

このタスクは、タスクの優先レベルを25に設定しています。また、タスク番号には10を設定しています。

タスクの生成後、タスクの識別にはタスク番号またはタスク名を用います。

## 9. 1. 3 タスクの削除

タスクを削除するには、`svdtask`を使用します。`svdtask`には、削除すべきタスクの名前を指定します。タスクを削除する例を以下に示します。

```
$ svdtask sample_1
$ svmap sample_1 -t
** allocator map **                                     2002/04/07 15:21:09

site name = test_CP

< task-program >

** map output end **
svmap : Specified name is undefined ( sample_1 )
$ svdload sample +P
$ svdla areal
$
```

## 9. 2 常駐サブプログラムの登録と削除

## 9. 2. 1 間接リンクサブプログラム (IRSUB) とは

タスク登録後に間接リンクサブプログラムを入れ替えても、タスクを再登録する必要のないサブプログラムのことです。

間接リンクサブプログラムは、プログラム内サブルーチンと同様に作成します。プログラム内サブルーチンと異なるのは、複数のタスクから共用されるため、リエントラントでなくてはならないことです。したがって、間接リンクサブプログラム内に静的変数を宣言して使用しないでください。

タスク登録後に間接リンクサブプログラムを再登録しても、登録番号は変わらず、対応するアドレスだけが変更されるため、タスクを再登録する必要はありません。

## 9. 2. 2 間接リンクサブプログラム (IRSUB) の登録

ローダ (svload) で実行可能モジュールを作成した後、ビルダのsvbuildにより登録します。

以下にサブプログラムを間接リンクサブプログラム (IRSUB) として登録する例を示します。

間接リンクサブプログラムを作成します。notepadなどのエディタを用います。

```
$notepad sub_a.c
```

```
sub_a()
{
    return;
}
```

間接リンクサブプログラムをコンパイルしてください。

```
$shc sub_a.c
```

間接リンクサブプログラムの登録番号を10として分割領域areal1に登録します。

```
$ svdfa areal 4096 -s
$ svload +I -a areal1 -o sub_a sub_a.obj -w 0
$ svbuild sub_a -ir -e 10
$ svmap sub_a -s -ir
```

```
** allocator map **
```

```
2002/04/07 15:23:15
```

```
site name = test_CP
```

```
< IRSUB >
```

```
irno entname          st laddr  subname      offset  texttop  bsslast
tsize dsize bsize extra ssize (part )
  10 sub_a             + s ib 60000000 sub_a      000000 + s 60000000 60000004
000004 000000 000000 000000 000000(000000)
```

```
** map output end **
```

## 9. 2. 3 間接リンクサブプログラム (IRSUB) の削除

登録済みの間接リンクサブプログラム (IRSUB) を削除するには、svdbuildを使用します。svdbuildには、削除すべきサブルーチンの名前と-irオプションを指定します。

以下に間接リンクサブプログラムを削除する例を示します。

```
$ svdbuild sub_a -ir
$ svdload sub_a +I
$ svmap -s -ir
** allocator map **                               2002/04/07 15:22:20

site name = test_CP

< IRSUB >
** map output end **
$
```

## 9. 3 組み込みサブルーチンの登録と削除

## 9. 3. 1 組み込みサブルーチンとは

組み込みサブルーチンについて説明します。組み込みサブルーチンでは、ハードウェアが検出した例外やソフトウェアで検出したイベントなどが発生したときに、OSで標準的に対処する異常処理の代わりにユーザのオリジナル処理をシステムに組み込むことができます。

組み込みサブルーチンは、svloadでロードしたものをビルダsvbuildによってOSのイベント処理プログラムの一部として組み込みます。

このシステムでは、いくつかの組み込み箇所（エントリポイント）が用意してあり、それぞれがさまざまなイベントに対応しています。したがって、サブルーチンの組み込み時に、どのイベントに対応する処理かによって組み込み箇所を選びます。

各組み込み箇所はポイント名称で識別し、サブルーチンの組み込み時に、このポイント名称を指定します。

1つの組み込み箇所には4個のサブルーチンが登録できます。実際にイベントが発生すると、登録された組み込みサブルーチンをエントリ番号が小さい方から順々に呼び出され実行されます。エントリ番号とは、登録した組み込みサブルーチンの実行順を指定するものです。

## 9. 3. 2 組み込みサブルーチンの登録

プログラムを組み込みサブルーチンとして登録する例を以下に示します。例にあげたプログラムuabs\_usrは、タスクがabortされたときに実行させるサブルーチンです。そこで、このサブルーチンをポイント名称ABSの組み込み箇所に登録します。なお、組み込みサブルーチンのリンク、ロードには、ローダ（svload）を使用し、かつオプション+Uを指定します。

```
$ shc uabs_usr.c -fpscv=safe
組み込みサブルーチンを登録します。

$ svdfa areal 4096 -s
$ svload +U -a areal -o uabs_usr uabs_usr.obj -w 512
$ svbuild uabs_usr ABS 1 -ul
$ svmap -s -ul
** allocator map **                                     2002/04/07 15:26:41

site name = test_CP

< ULSUB >
pnt typ ent subname          texttop bsslast tsize dsize bsize
extra ssize (part )
abs os 1 + uabs_usr          + s 60001000 60001004 000004 000000 000000
000000 000200(000200)

** map output end **
$
```

## 9. 3. 3 組み込みサブルーチンの削除

登録済みの組み込みサブルーチンを削除するには、`svdbuild`を使用します。`svdbuild`には、削除すべきサブルーチンの名前、組み込みポイント名称と`-ul`オプションを指定します。

登録してある組み込みサブルーチンを削除する例を以下に示します。

```
$ svdbuild uabs_usr ABS -ul
$ svdload uabs_usr +U
$ svmap -s -ul
*** allocator map **                               2002/04/07 15:27:41

site name = test_CP

< ULSUB >

** map output end **
$
```

## 第10章 マップ

### 10. 1 アロケータ管理テーブル情報表示の目的

アロケータ管理テーブル情報表示の目的は、リアルタイムプログラムの開発を円滑に進めることです。

- タスク、サブプログラム、GLBなどの共有資源用の格納領域情報を表示し、個々のプログラム作成はもとより、システム設計の開発を支援します。
- コントローラにロードしたシステムのアロケータ管理テーブル情報を表示することで、デバッグ作業を支援します。

## 10. 2 svmapコマンドのオプション指定と表示情報

svmapコマンドのオプション指定ごとの表示フォーマットは、「付録F svmapの表示フォーマット」に示します。表示フォーマット中の下線 (   ) 部分は、svmapコマンドによる表示データです。

### 10. 2. 1 マップ情報の出力対象

マップ情報は、以下に示す対象の情報を出力します。

- (1) 開発系側で管理するリソースのマップ情報
- (2) PCsにダウンロードしたリソースのマップ情報

### 10. 2. 2 マップ情報の出力内容

マップ情報は、以下に示す情報を出力します。

- (1) ヘッダーとフッター
- (2) 大分割領域情報
- (3) 分割領域情報
- (4) 細分割領域情報
- (5) プログラム情報
- (6) サブプログラム情報
- (7) タスク情報
- (8) グローバル情報
- (9) VAL情報
- (10) IRSUBエントリ情報
- (11) IRGLBエントリ情報
- (12) ULSUBエントリ情報
- (13) 物理メモリの空き情報

### 10. 2. 3 マップ情報の出力形式

マップ情報は、以下に示す形式で出力することができます。

- (1) 階層マップ出力
- (2) アドレス順リスト出力
- (3) 名称順リスト出力
- (4) 番号順リスト出力
- (5) 名称指定出力

階層マップ出力は、指定大分割領域、分割領域単位に、論理空間上に配置されるリソースのマップ情報を階層的に出力します。

リスト出力は、指定情報をアドレス順、名称順、番号順に並べて出力します。また、リソースの名称を指定し、その名称単独の情報を出力することもできます。

表 1-22に出力内容と指定可能出力形式の組み合わせを示します。

表 1-22 出力内容と指定可能出力形式の組み合わせ

出力形式 出力内容	階層出力	アドレス順 出力	名称順出力	番号順出力	名称指定
大分割領域情報	○	○	×	×	○
分割領域情報	○	○	○	×	○
細分割領域情報	○	○	○	×	○
プログラム情報	×	×	○	×	○
サブプログラム 情報	×	×	○	○	○
タスク情報	×	×	○	○	○
グローバル情報	×	×	○	○	○
VAL情報	×	×	○	×	○
IRSUBエン트리 情報	×	×	○	○	○
IRGLBエン트리 情報	×	×	○	○	○
ULSUBエン트리 情報	×	×	○	○	○

○：指定可 ×：指定不可

## 10. 3 svadmコマンドの論理アドレス指定と表示情報

svadmコマンドは、指定論理アドレスに対して名称等の情報を、コマンド、会話形式で表示し、デバッグ作業を支援します。

## (1) コマンド形式

svadm論理アドレス

パラメータに論理アドレスを指定した場合、以下のフォーマットで名称等の情報を表示します。

“XXX” は、svadmコマンドによる表示データを示します。

## (a) プログラムのTEXT部とDATA部空間、GLB空間、サブプログラム空間の表示フォーマット

```
name = XXXXXXXX  type = XXXXXXXXXXXX  raddr = XXXXXXXX
```

## (b) プログラムのBSS部とSTACK部空間の表示フォーマット

```
name = XXXXXXXX  type = XXXXXXXXXXXX  raddr = XXXXXXXX  pgmno = XXXX
```

## (2) 会話形式

パラメータに論理アドレスを指定しなかった場合、会話形式で論理アドレスを取り込み、コマンド形式と同一フォーマットで名称等の情報を表示します。

— 会話形式の論理アドレス取り込み —

```
#svadm [Enter]
```

```
++ address information display start --> site(XXXXX) ++
```

```
addr : addr [Enter]
```

```
情報表示
```

```
addr : q [Enter]
```

```
++ address information display end ++
```

```
#
```

## 第11章 立ち上げ

### 11. 1 概要

PCsの立ち上げは、開発系のマシン内のPCs主メモリ（SDRAM）の初期データファイル（バックアップファイル）をPCs主メモリへダウンロードすることにより行います。

アロケータにより生成したバックアップファイルをPCs内の指定サイトの主メモリへダウンロードし、指定サイトを立ち上げます。

## 11. 2 CMUの状態遷移

## 11. 2. 1 立ち上げ操作

立ち上げは、開発系マシンから“svrpl”コマンドを実行することにより行います。

開発系マシンでの操作については、「第2編 コマンドリファレンス」の「第6章 立ち上げ」を参照してください。以下に主な機能と使用例を示します。

<機能>

svrplコマンドの主な機能を以下に示します。

- ・ 指定サイトを立ち上げます。
- ・ バックアップファイルをダウンロードします。
- ・ 立ち上げ時にCMUの時刻を設定する／しないを指定します。
- ・ バックアップファイルのダウンロード終了後にユーザタスク使用可能状態にする／しないを指定します。
- ・ “svrpl” コマンド実行時にユーザタスク使用不可状態にするかどうかの確認応答をとらないでユーザタスクを停止します。

各機能を使用した場合の立ち上げ例を以下に示します。

- 全バックアップファイルをダウンロードし、ダウンロード後にCMUを再起動する場合の例

```
>svrpl -u test
**** svrpl start (site = test) ****
CMU status(test)=RUN
Do you stop UserTask(test) ? (yes/no)= yes
Remote loading start(site = test)
address : 0X0000d000-0X0000d0ff
.
address : 0X00b00000-0X00dfffff
....
start to modify allocator management tables

finished to modify allocator management tables
Remote loading end
Reset start(test)
**** svrpl end ****
```

- ユーザタスクを停止させてもよいかの確認応答をとらないで、CMUへ全バックアップファイルをダウンロードし、ダウンロード後にCMUを再起動する場合の例 (-sオプションを指定し、確認応答をとらない場合はメッセージを表示しません。)

```
>svrpl -u test -s
```

11. 2. 2 CMU制御操作

CMU制御は、開発系マシンから“svcpuctl”コマンドを実行することにより行います。開発系マシンでの操作については、「第2編 コマンドリファレンス」の「第6章 立ち上げ」を参照してください。以下に主な機能と使用例を示します。

<機能>

svcpuctlコマンドの主な機能を以下に示します。

- CMUのユーザタスク状態（runまたはstop）を制御します。
- “svcpuctl”コマンド実行時にCMUの起動／停止の確認応答（本当に実行させてもよいか）をとらないで実行します。
- サイトの状態を表示します。

各機能を使用した場合の立ち上げ例を以下に示します。

- CMUにrun要求を行うと同時に、CMUへの時刻設定を行う場合の例

```
>svcpuctl -u test -time
status CMU(test)=STOP
Input CMU(test) status ? (stop/run)= run
Do you really request OK ? (yes/no)= yes
CMU(test) = RUN
```

- CMUにstop要求を行う場合の例

```
>svcpuctl -u test
status CMU(test)=RUN
Input CMU(test) status ? (stop/run)= stop
Do you really request OK ? (yes/no)= yes
CMU(test) = STOP
```

- ユーザタスクを停止させてもよいかの確認応答をとらないで、CMUにrun要求を行い、CMUへの時刻設定を行う場合の例

（-sオプションを指定し、確認応答を取らない場合はメッセージを表示しません。）

```
#svcpuctl -u test -s -run -time
```

- CMUの状態表示を行う場合の例

```
#svcpuctl -u test -ss
status CMU(test_CP)=STOP
```

## 第12章 svdebug (オンラインデバッガ) とデバッグ支援コマンド

## 12. 1 概要

svdebugは、PCs上で動作するプログラムを開発系マシン上からデバッグするためのコマンドで、以下のような基本的なデバッグ機能を提供します。

分類	サブコマンド	機能
タスク起動/停止	qu ab re ta su rs tm ct sht	タスクの起動要求 タスクの起動禁止 タスクの起動禁止解除 タスクの状態表示 タスクの実行抑止 タスクの実行抑止解除 タスクの周期起動 タスクの周期起動解除 タスクの周期起動表示
メモリプリント/ パッチ	md sd bs bg mcp mmv mf	アドレス指定でのメモリプリント/パッチ 名称指定でのメモリプリント/パッチ ビットデータの設定 ビットデータの表示 メモリのコピー メモリの移動 メモリのパターンセット
システムエラー表示	el ss	システムエラーの表示 (svelogコマンドの実行) システム状態表示 (svcpuctlコマンドの実行)
現在時刻設定/表示	st gt	現在時刻設定 現在時刻表示
ブレークポイント 設定/解除	br rb rd rr go	ブレークポイントの設定/表示 ブレークポイントの解除 レジスタの表示 レジスタの書き換え ブレークポイントからの実行再開
ADTの設定/解除	as ac	ADTの設定/表示 ADTの解除
アップ/ダウン ローディング、 コンペア	ld sv cm	バックアップファイル → PCsメモリ転送 PCsメモリ → バックアップファイル転送 PCsメモリ/バックアップファイル比較
DHP記録許可/禁止	dr ds	DHP記録許可 (svdhpコマンドの実行) DHP記録禁止 (svdhpコマンドの実行)
その他	si sp ps pe ver smd svadm svdhp help q !	スタック初期化 スタック使用量表示 デバッグ文の表示開始 デバッグ文の表示終了 CMUのバージョン表示 全領域に対するメモリのプリント/パッチ アドレスに対する情報表示 (svadmコマンドの実行) DHPの表示 (svdhpコマンドの実行) サブコマンド一覧表示 デバッガの終了 開発系マシン上のコマンドの実行

12. 2 PCs状態とサブコマンド実行可否

svdebugは、ユーザタスク実行不可状態では下記のサブコマンドが使用できません。  
現在の状態は、svdebug ssまたはsvcpuctlコマンドで参照できます。

分類	使用不可サブコマンド
タスク起動/停止	qu、ab、re、ta、su、rs、tm、ct、sht
現在時刻設定/表示	st、gt
ブレークポイント設定/解除	br、rb、rd、rr、go
ADTの設定/解除	as、ac
アップ/ダウンローディング、コンペア	ld (*)、sv、cm
スタック初期化/使用量表示	si、sp

(\*) ld -g、ld -Gは使用できます。

svdebugは前述の機能をサブコマンドとして実現しています。svdebugを起動するとプロンプトが表示されます。プロンプトには、処理対象サイト名もあわせて表示されます。これにより、現在のデバッグ対象となっているサイトを確認できます。このプロンプトに対して適切なサブコマンドを入力し、デバッグを進めることができます。サブコマンド“q”を入力すると、svdebugは終了します。

以下に例を示します。

```
$ svdebug
++ debugger start ++
px1> q
++ debugger end ++
```

リアルタイムリソースは、所有者種別、使用者種別によりアクセスが制限されます。svdebugを起動する前に適切な使用者種別を設定しておいてください。

svdebugのサブコマンド処理は、シグナルにより中断できます。

## 12. 3 基本的な機能

## (1) プログラム、データのダウンロード・アップロードサブコマンド

アロケータ・ローダ・ビルダ (ALB) は、バックアップファイル中にプログラムやデータを組み込みます。何らかの方法でバックアップファイルの内容をPCsのメモリに反映しなければなりません。ldサブコマンドは、バックアップファイルの内容をPCsメモリ上に反映します。また、その逆にPCsメモリ内容をバックアップファイルに反映することもできます。これにはsvコマンドを使用します。また、バックアップファイルと対応する主メモリの内容を比較するために、cmサブコマンドを用意しています。

以下の例では、TN=5のタスクの状態を確認後、ダウンロードを行い、さらにGLB (名称: indata) のダウンロードを行っています。

```
$$ svdebug
++ debugger start ++
px1> ta 5
tn=5 (0x05) tname=ta5 task state=DORMANT (0x00000000)
tcb top=0x82006280 fact=0x00000000 level=00 (27)
task top=0x02000000 stack=0x409ff000-0x40a00000

px1> ld -t ta5
address : 0x20000280-0x200002bf
address : 0x30000000-0x30000067
address : 0x20000280-0x200002bf
px1> ld -g indata
address : 0x50000400-0x50000bff
px1>
```

前述したように、ALBでの操作はそのままではPCsに反映されません。つまり、開発系マシン上のバックアップファイルとPCs上の状態が不一致となる場合があります。

- 開発系マシン上のバックアップファイルには存在するがコントローラ上には存在しない場合  
ALBで登録したが、svrplによる一括ロードやsvdebugのldサブコマンドにより反映されていない場合です。
- 開発系マシン上のバックアップファイルとPCs上の両方に存在する場合  
この場合、2つの状態が考えられます。  
1つは開発系マシン上とPCs上に全く同一のものが存在する場合と、もう1つは開発系マシン上とPCs上に異なるものが存在する場合です。前者はALBで登録したものを一括ロードやsvdebugのldサブコマンドにより反映した直後の状態です。後者はALBで登録や削除を繰り返した場合です。
- 開発系マシン上のバックアップファイルには存在しないが、PCs上には存在する場合  
ALBで登録し、さらにsvrplによる一括ロードやsvdebugのldサブコマンドにより反映した後に、ALBで該当のものを削除した場合です。
- 開発系マシン上のバックアップファイルとPCs上の両方に存在しない場合  
ALBで登録し、さらにsvrplによる一括ロードやsvdebugのldサブコマンドにより反映した後に、ALBで該当のものを削除し、PCs上からも削除した場合です。  
開発系マシンとPCsの状態は、svmapコマンドやsvdebugのldサブコマンドにより調べることができます。必要に応じて一括ロードまたは個別ロードにより状態を一致させてください。特に、リソースの削除時は、開発系マシン上だけで削除され、実機に反映されていない状態にならないようにしてください。

(2) タスク制御サブコマンド

PCs上にダウンロードしたプログラムは、通常他のタスクから起動要求を受けて起動されますが、デバッグ時にはオペレータの指示によりタスクを起動したり停止したりできると便利です。svdebugでは、タスクの起動や停止を行うためのサブコマンドを用意しています。状態を確認しながらTN=5のタスクを起動する例を以下に示します。

```

$ svdebug
++ debugger start ++
px1> ta 5
tn=5 (0x05) tname=ta5 task state=DORMANT (0x00000000)
tcb top=0x82006280 fact=0x00000000 level=00 (27)
task top=0x02000000 stack=0x409ff000-0x40a00000

px1> re 5
OK (0)
px1> qu 5
OK (0)
px1> ta 5
tn=5 (0x05) tname=ta5 task state=IDLE (0x00000000)
tcb top=0x82006280 fact=0x00000000 level=00 (27)
task top=0x02000000 stack=0x409ff000-0x40a00000

px1>
    
```

## (3) メモリプリント・パッチサブコマンド

svdebugは、メモリの内容を表示したり変更するサブコマンドを持ちます。これによってGLBなどのメモリ内容を任意の値に変更したり、参照しながらテストを行うことができます。

パッチまたは表示の対象は、主メモリ、バックアップファイル、またはその両方を指定できます。

入力データを格納するGLB (名称: indata) にデータを設定した後、該当データを処理GLB (名称: outdata) に書き込むタスクを起動し、処理結果を確認する例を以下に示します。

```
$ svdebug
++ debugger start ++
px1> sd
1 name : indata -g
2 storage (s,m,*) : m
3 baddr : 0
4 raddr : 0
0x50000000(0x000000) 00000000 : 0x1000
0x50000004(0x000004) 00000000 : e
4 raddr : *1
1 name : indata -g
2 storage (s,m,*) : m
3 baddr : 0
4 raddr : 0
0x50000000(0x000000) 00001000 :
0x50000004(0x000004) 00000000 : e
4 raddr : e
px1> qu test
OK(0)
px1> sd
1 name : outdata -g
2 storage (s,m,*) : m
3 baddr : 0
4 raddr : 0
0x50001000(0x000000) 00002000 : e
4 raddr : e
px1>
```

## (4) 時刻設定表示サブコマンド

PCsは内蔵クロックを持ちます。この内蔵クロックへの時刻設定はsvdebugからも行うことができます。PCs上の現在時刻を確認した後、時刻を再設定する例を以下に示します。

```
$ svdebug
++ debugger start ++
px1> gt
2002.04.04.20:27:32
px1> st 2002.04.04.20:30:00
OK(0)
px1>
```

(5) システムエラー表示、システム状態表示

テストやデバッグ作業を進めているときには、システムの状態やシステムエラーの内容を参照してください。S10V RPD Pではシステムエラーを表示したり、システムの状態を表示したりするためのコマンド (svelog、svcpuctl) を用意していますが、svdebugでは、それらのコマンドをサブコマンド (el、ss) として起動できます。

(6) dhp記録許可・禁止、表示サブコマンド

PCsはデバッグ支援機能としてDHP (Debugging Helper) 機能を提供します。DHPによりOSやユーザタスクの動作を解析できます。S10V RPD PではDHPの表示や、DHP記録許可、禁止を行うコマンド (svdhp) を用意していますが、svdebugでは、それらのコマンドをサブコマンド (svdhp) として起動できます。

(7) その他のサブコマンド

今まで説明したサブコマンドの他にもいくつかのサブコマンドがあります。

si、spサブコマンドによりタスク実行時のスタック使用量を求める作業がより容易になります。

svadmコマンドにより、アドレスに対応するSAREA名やIRSUB名を求めることができます。

!サブコマンドにより、svdebugを終了することなく開発系マシン上のコマンドを実行できます。

helpサブコマンドにより簡潔な説明文が表示されます。

## 12. 4 その他の機能

svdebugではユーザの利便性を考慮し、いくつかのコマンド行オプションを設けています。以下のオプションを指定することで、svdebugの動作を変更し、用途に合わせて使用できます。

- i : キー入力結果をファイルに出力します。サブコマンドの出力結果は記録されません。
- o : 指定されたファイルに操作の日付および結果を出力します。  
どのような操作を行ったのか記録したい場合に便利です。
- r : 指定されたコマンドファイル内のサブコマンド行を実行します。  
一連の操作を1回で行いたいときに便利です。  
-iオプションで作成したファイルを入力に使用できます。
- s : このオプションに指定したサブコマンドを直接実行します。  
コマンドプロシージャ等を作成する場合に便利です。
- u site : デバッガの処理対象となるサイト名称を指定します。

## 12. 5 デバッグ支援コマンド

### 12. 5. 1 svelogコマンド

PCs内エラーログバッファからエラーログ情報をネットワークを介して取り込み、最新のエラーログ情報から順に出力します。

出力されたエラーログ情報からは、いつ (発生時刻)、何が原因で (EC: エラーコード)、だれが (TN: タスク番号)、どこで (PC: プログラムカウンタ)、何をしようとしたか (iarv0~iarv8、iarvn1~iarvn8: PC前後のプログラムの内容) などを知ることができます。さらに詳細に解析するための、レジスタ情報やDHPトレース情報等も併せて出力します。代表的な操作例を以下に示します。

(例1)

サイトsiteAのエラーログ情報を、簡略化した短い形式で出力します。

```
svelog -u siteA -f s
```

-fsオプション省略時も同様です。

(例2)

サイトsiteAのエラーログ情報を、すべて出力します。

```
svelog -u siteA -f m
```

(例3)

サイトsiteAのエラーログ情報に加えてDHPトレース情報も出力します。

```
svelog -u siteA -f l
```

(例4)

サイトsiteAのエラーログ情報に加えてDHPトレース情報も出力すると共に、abcというファイルにエラーログ情報を格納します。

```
svelog -u siteA -f l -o abc
```

## 12. 5. 2 svdhcpコマンド

DHPトレースバッファに格納されている、現在のDHPトレース情報をネットワークを介して取り込んで出力します。

出力するDHPトレース情報には、DHP記録時刻、DHPトレースポイント、解析に必要なデータがあります。また、DHP記録の停止／再開を制御します。

代表的な操作の例を以下に示します。

(例1)

サイトsiteAのDHPトレース情報を指定されたカウント分 (10) 出力します。

```
svdhcp -u siteA +10
```

指定されたカウント (+10) が省略された場合は、すべてのDHPトレース情報を出力します。

(例2)

サイトsiteAのDHP記録を停止します。

```
svdhcp -u siteA -off
```

(例3)

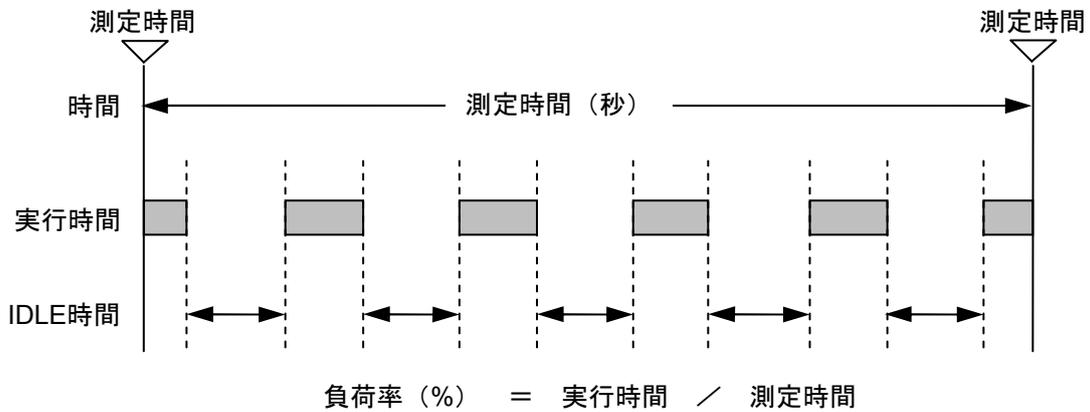
サイトsiteAのDHP記録を再開します。

```
svdhcp -u siteA -on
```

12. 5. 3 svcpunowコマンド

PCsの負荷率をサイト単位に出力します。

負荷率は、測定開始から終了までの実行時間の合計によりCMUをどの位占有したかを百分率 (%) で出力します。



代表的な操作例を以下に示します。

(例) サイトsiteAの負荷率を10秒測定します。

```

svcpunow -u siteA -t 10
1996/07/03 09:56:00 SITE=siteA ** 10 second wait **
CMU(siteA) load ratio = 12.34%
```

( \*1 )                      ( \*2 )

(注) 下線部は入力してください。

(\*1) サイト名称

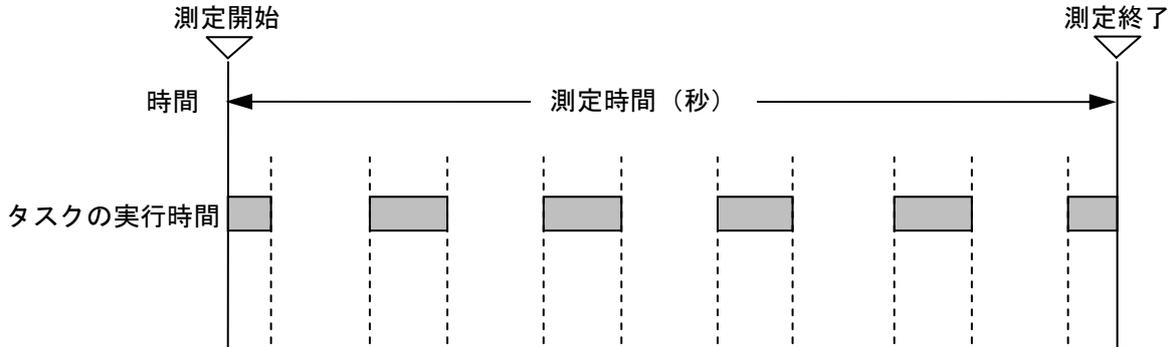
(\*2) PU負荷率

測定時間の指定 (-t 10) が省略された場合は、1秒間の負荷率を測定します。

12. 5. 4 svtimexコマンド

PCsに登録されているタスクの稼働情報を出力します。

出力するタスクの稼働情報には、タスクのPU負荷率、実行回数、実行時間、平均実行時間があります。



$$\begin{aligned} \text{タスクのPU負荷率 (\%)} &= \text{タスクの実行時間の合計} / \text{測定時間} \\ \text{タスクの平均PU実行時間} &= \text{タスクの実行時間の合計} / \text{タスクの実行回数} \end{aligned}$$

代表的な操作例を以下に示します。

- (例1) 下記タスクの稼働情報を10秒間測定します。  
 タスク名称=taska、タスク番号=123

```
#svtimex -u siteA taska -t 10
1996/07/03 13:30:24 SITE=siteA ** 10 second wait **
taska(123) load ratio=3.00% execute count=10 total time=0.030sec average time=0.003sec
```

( \*1 )                    ( \*2 )                    ( \*3 )                    ( \*4 )                    ( \*5 )

- (注) 下線部は入力してください。  
 (\*1) タスク名称 (番号)                    (\*4) タスクの実行時間の合計  
 (\*2) タスクのPU負荷率                    (\*5) タスクの平均PU実行時間  
 (\*3) タスクの実行回数

測定時間の指定 (-t 10) が省略された場合は、1秒間の負荷率を測定します。

- (例2) 会話形式により下記タスクの稼働情報を3600秒間測定します。  
 タスク名称=taska、タスク番号=123  
 タスク名称=taskb、タスク番号=124  
 タスク名称=task22c、タスク番号=111

```
#svtimex -u siteB
SITE=siteB Task measuring period[sec] = 3600
Task name or number = taska
Task name or number = taskb
Task name or number = 111
Task name or number = [Enter]
1996/07/17 13:30:24 SITE=siteB ** 3600 second wait **
taska(123) load ratio=3.00% execute count=36000 total time=108.000sec average time=0.003sec
taskb(124) load ratio=2.50% execute count=18000 total time=90.000sec average time=0.005sec
task22c(111) load ratio=0.01% execute count=360 total time=0.360sec average time=0.001sec
```

(注) 下線部は入力してください。

このページは白紙です。

## 第2編 コマンドリファレンス

## 第1章 システムジェネレーション

### <名前>

svgen

### <形式>

svgen fname

### <機能説明>

svgenは、指定したシステム構築環境定義ファイル内に定義されたシステム構築環境（サイト）を生成し、PCsシステム管理ファイル（sysadm）に登録します。

また、生成した環境下にユーザ定義情報入力ファイルの雛形を生成します。

コマンドはadministratorで実行してください。

### <引数説明>

fname：生成するシステム構築環境を定義した定義ファイルを指定します。

### <終了コード>

次の終了コードを返します。

0：正常終了

1：定義ファイルに誤りがあります。

2：システム構築環境の生成に失敗しました。

3：PCsシステム管理ファイルが更新できませんでした。

### <システム環境定義ファイルの構成>

***SUBSYS\_DIR***=△サイト格納ディレクトリ  
***SITE\_NAME***=△サイト名

太字・斜体の部分はキーワードを表します。キーワードを含めて記述してください。

△は1つ以上の空白またはタブを表します。

#### ・サイト格納ディレクトリ

サイトを格納するディレクトリをドライブレターを含むフルパスで記述します。

サイトディレクトリの文字数は255文字以内としてください。

#### ・サイト名

サイト名を指定してください。

サイト名は14文字以内の半角英数字と\_（アンダーバー）、+（プラス）とってください。

<名前>

svconf

<形式>

svconf sitename

<機能説明>

svconfは、サイト環境下のetc¥confに生成したユーザ定義情報入力ファイルの定義情報を基に、PCsシステム環境を構築します。コマンドはadministratorで実行してください。

<引数説明>

sitename : 構築するサイトの名称を指定してください。

<終了コード>

次の終了コードを返します。

0 : 正常終了

1 : 定義ファイルに誤りがあります。

2 : 定義情報出力ファイルの生成に失敗しました。

## 第1章 システムジェネレーション

### <名前>

svshconf

### <形式>

svshconf sitename [-f fname]

### <機能説明>

svshconfは、指定サイトの定義情報の一覧を出力します。

### <引数説明>

sitename : 定義情報を表示するサイト名を指定します。

### <オプション>

-f fname : 結果を出力するファイル名を指定します。

省略時は結果を標準出力に出力します。

### <終了コード>

次の終了コードを返します。

0 : 正常終了

1 : 定義情報の読み込みに失敗しました。

### <出力フォーマット>

以下に出力フォーマットを示します。

```
<<<Show Subsystem directory>>>
  c:\users\ys10v\kwsys          --- サイトディレクトリ情報

<<< Show Site name >>>
  Test

<<< Site Information (test) >>>
  TASK_SIZE          512KB          --- メモリ情報
  GLBR_SIZE          512KB          そのサイトのエリア情報となります。
  GLBW_SIZE          512KB
  IRSUB_SIZE         512KB

<<< I/O Information (test) >>> --- I/O情報

Adapter Information : LANCE
SLOT 0
STATION_NO 1
IF_COUNT 1
UNO 1 IF_NUM 1 IP_ADDR 192.192.192.1 NETMASK 255.255.255.0 BRD_ADD 192.192.192.255
```

<名前>

svsitecp

<形式>

svsitecp siteX1 siteY1

<機能説明>

svsitecpは、指定されたサイト単位にPCsシステムをコピーします。

コマンドはadministratorで実行してください。

<引数説明>

siteX1：コピー元サイト名を指定します。

siteY1：コピー先サイトのサイトディレクトリ名をドライブレターを含むフルパスで指定します。

サイトディレクトリの文字数は98文字以内、サイト名称の文字数は14文字以内にしてください。

<終了コード>

次の終了コードを返します。

0：正常終了

1：指定したサイト名に許されない文字を使用しています。

または、長さが14文字を超えています。

2：コピー先サイト名が既存のサイトと重複しています。

3：指定コピー先のサイトディレクトリの名称が誤っています。

または、アクセスする権限がありません。

6：サイトのコピーに失敗しました。

7：PCsシステム管理ファイルが更新できませんでした。

## 第1章 システムジェネレーション

### <名前>

svsitedel

### <形式>

svsitedel sitename

### <機能説明>

svsitedelは、指定されたサイトのPCsシステムを開発系マシンから削除します。  
コマンドはadministratorで実行してください。

### <引数説明>

sitename : 削除するサイト名を指定します。

### <終了コード>

次の終了コードを返します。

0 : 正常終了

1 : サイトの削除に失敗しました。

2 : PCsシステム管理ファイルが更新できませんでした。

### <注意事項>

このコマンドでサイト、ディレクトリを削除するときにエクスプローラ等でサイトディレクトリを表示中の場合、ファイルオープン中のため、サイトディレクトリの消去が不可となり、下記メッセージを出力し異常終了します。

“svsitedel : Can't Delete Directory (ディレクトリパス) (EC0613)”

この場合、エクスプローラ等での参照を止め、svsitedelコマンドを“-d”オプション付きで削除してください。

## 第2章 アロケータ

### <名前>

svdfa

### <形式>

svdfa aname size [オプション]

### <機能説明>

svdfaは、指定した大分割領域内に指定した分割領域を確保し、バックアップファイルを生成します。

### <引数説明>

aname : 確保する分割領域名称を指定します。

Size : 確保する分割領域の大きさを指定します。4096の倍数のバイト数で指定してください。

4096の倍数以外を指定した場合は、警告メッセージを表示後、4096の倍数に切り上げます。

### <オプション>

-p : タスクの格納エリアを確保します。

-s : サブプログラムの格納エリアを確保します。

-gi : 読み書き両用グローバルエリアに初期値ありGLBのエリアを確保します。

-gw : 読み書き両用グローバルエリアに初期値なしGLBのエリアを確保します。

-gr : 読み出し専用グローバルエリアに初期値ありGLBのエリアを確保します。

-S : アクセス権がシステムであることを指定します。このオプション省略時は、あらかじめ設定された環境変数RSUTYPとします。デフォルトは、ユーザ (RSUTYP=u) です。

-u site : アロケータの処理対象となるサイト名称 (site) を指定します。このオプション省略時は、あらかじめ設定された環境変数RSSITEにサイト名称を指定します。環境変数RSSITEのデフォルトはありません。

-f adr : 確保する領域のアドレス (adr) を指定します。大分割領域の先頭からの相対バイトアドレスとして4096の倍数を指定してください。4096の倍数以外を指定した場合は、警告メッセージを表示後4096の倍数に切り上げます。なお、このオプションを省略した場合は自動割り付けとなり、最初に検出した空き領域に割り付けます。

### <注意事項>

-p、-s、-gi、-gw、-grのいずれの指定もなかった場合、-pが指定されたものとします。

-gi、-grのいずれかを指定して確保した分割領域内に、svdfsコマンドで細分割領域を確保したとき、確保した細分割領域は0で初期化します。

表2-1にオプションの組み合わせを示します。

表 2-1 svdfaのオプションの組み合わせ

パラメータ	領域種別					
	タスク	サブプログラム	読み書き両用 初期値ありGLB	読み書き両用 初期値なしGLB	読み出し専用 初期値ありGLB	
aname	◎	◎	◎	◎	◎	
size	◎	◎	◎	◎	◎	
オプション	-p (デフォルト)	○	×	×	×	×
	-s	×	◎	×	×	×
	-gi	×	×	◎	×	×
	-gw	×	×	×	◎	×
	-gr	×	×	×	×	◎
	-S	○	○	○	○	○
	-u site	○	○	○	○	○
	-f adr	○	○	○	○	○

◎：必須 ○：指定できます。 ×：指定できません。

使用者種別と確保領域の所有者種別の関係を以下に示します。

使用者種別	確保領域の所有者種別
システム	システム
ユーザ	ユーザ

<終了コード>

次の終了コードを返します。

- 0：正常終了
- 1：オプションの指定に誤りがあります。
- 2：GAREAの空きが不足しています。
- 3：最大登録数を超えています。
- 4：管理テーブルが更新できませんでした。

## &lt;名前&gt;

svdla

## &lt;形式&gt;

svdla aname [オプション]

## &lt;機能説明&gt;

svdlaは、svdfaで確保した分割領域を削除し、バックアップファイルを削除します。

## &lt;引数説明&gt;

aname : 削除する分割領域名称を指定します。

## &lt;オプション&gt;

-S : アクセス権がシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト（環境変数RSUTYP）にします。

-u site : アロケータの処理対象となるサイト名称（site）を指定します。このオプション省略時は、あらかじめ設定されたデフォルト（RSSITE）にします。

## &lt;注意事項&gt;

指定した分割領域内に読み書き両用グローバル、読み出し専用グローバルの細分割領域が存在する場合は、指定した分割領域内の細分割領域も同時に削除します。

使用者種別と削除領域の所有者種別の関係を以下に示します。

使用者種別	削除領域の所有者種別	
	システム	ユーザ
システム	○	○
ユーザ	×	○

○ : 削除できます。× : 削除できません。

## &lt;終了コード&gt;

次の終了コードを返します。

0 : 正常終了

1 : オプションの指定に誤りがあります。

2 : ユーザはシステムの領域を削除できません。

3 : 管理テーブルが更新できませんでした。

## 第2章 アロケータ

### <名前>

svdfs

### <形式>

svdfs aname sname size [オプション]

### <機能説明>

svdfsは、svdfaで確保した分割領域内にグローバルの細分割領域を確保します。  
確保した領域は0で初期化します。

### <引数説明>

- aname : 細分割する分割領域名称を指定します。
- sname : 確保する外部名称（グローバル名称）を指定します。
- size : 確保する細分割領域のサイズをバイト数で指定します。

### <オプション>

- S : アクセス権がシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト（環境変数RSUTYP）にします。
- u site : アロケータの処理対象となるサイト名称（site）を指定します。このオプション省略時は、あらかじめ設定されたデフォルト（RSSITE）にします。
- l adr : 確保する細分割領域のアドレス（adr）を指定します。分割領域の先頭からの相対バイトアドレスとして4の倍数を指定してください。4の倍数以外を指定した場合は、警告メッセージを表示後、4の倍数に切り上げます。なお、このオプションを省略した場合は自動割り付けとなり、最初に検出された空き領域に割り付けます。
- a align : 細分割領域確保時のアライン数（align）を指定します。2のn乗（ $0 \leq n \leq 12$ ）を指定してください。このオプションを省略した場合のアライン数は2となります。
- t svtype : svtypeで指定するデータ型に従ったアラインとします。stypeに指定する値とアライン数の関係を表2-2に示します。
- e idxnum : 細分割領域を間接リンクグローバルに割り当てる場合に、確保する分割領域に割り当てるエントリ番号（idxnum）を指定します。このオプションを省略した場合、エントリ番号は割り当てられません。idxnumには1~IRG\_MAXNUMの値が指定できます。

### <注意事項>

- -eオプションを指定せずに確保した細分割領域を間接リンクグローバルに割り当てる場合、svirglbコマンドにて任意の細分割領域にエントリ番号を割り当てることができます。
  - アライン数は使用するデータ型に合わせて決定してください。
  - -l、-a、-tオプションを同時に指定することはできません。
- 表2-3に指定可能なオプションの組み合わせを示します。

表2-2 stypeに指定する値とアライン数の関係

svtype	データ型	アライン数
1	char	0 (1バイト)
2	short	1 (2バイト)
3	long	2 (4バイト)
4	struct	3 (8バイト)
5	float	2 (4バイト)
6	double	3 (8バイト)
7	long double	4 (16バイト)

表2-3 svdfsのオプションの組み合わせ

パラメータ	領域種別					
	タスク	サブプログラム	読み書き両用 初期値ありGLB	読み書き両用 初期値なしGLB	読み出し専用 初期値ありGLB	
aname	×	×	◎	◎	◎	
sname	×	×	◎	◎	◎	
size	×	×	◎	◎	◎	
オプション	-S	×	×	○	○	○
	-u site	×	×	○	○	○
	-l adr	×	×	○	○	○
	-a align	×	×	○	○	○
	-t svtype	×	×	○	○	○
	-e idxnum	×	×	○	○	○

◎：必須 ○：指定できます。×：指定できません。

使用者種別と確保される細分割領域の所有者種別の関係を以下に示します。

使用者種別	削除領域の所有者種別	
	システム	ユーザ
システム	○ (システム)	○ (ユーザ)
ユーザ	×	○ (ユーザ)

○：確保できます。×：確保できません。

( )内は確保される細分割領域の所有者種別を示します。

<終了コード>

次の終了コードを返します。

- 0：正常終了
- 1：オプションの指定に誤りがあります。
- 2：ユーザはシステムの領域にsareaを作成できません。
- 3：最大登録数を超過しています。
- 4：管理テーブルが更新できませんでした。

## 第2章 アロケータ

### <名前>

svdls

### <形式>

svdls aname [オプション]

### <機能説明>

svdlsは、svdfsで確保した細分割領域を削除します。

### <引数説明>

aname : 削除する細分割領域領域名称を指定します。

### <オプション>

-S : アクセス権がシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト（環境変数RSUTYP）にします。

-u site : アロケータの処理対象となるサイト名称（site）を指定します。このオプション省略時は、あらかじめ設定されたデフォルト（RSSITE）にします。

### <注意事項>

使用者種別と削除領域の所有者種別の関係を以下に示します。

使用者種別	削除領域の所有者種別	
	システム	ユーザ
システム	○	○
ユーザ	×	○

○ : 削除できます。 × : 削除できません。

### <終了コード>

次の終了コードを返します。

0 : 正常終了

1 : オプションの指定に誤りがあります。

2 : ユーザはシステムの領域を削除できません。

3 : 管理テーブルが更新できませんでした。

## &lt;名前&gt;

svdfv

## &lt;形式&gt;

svdfv *ename* *value* [オプション]

## &lt;機能説明&gt;

svdfvは、バリュ情報の外部参照情報を登録します。

## &lt;引数説明&gt;

*ename* : 登録する外部名称を指定します。*size* : 外部名の取る値 ( $-2^{31} \leq \text{value} \leq 2^{31}-1$ ) を指定します。

## &lt;オプション&gt;

-S : アクセス権がシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト（環境変数RSUTYP）にします。

-u *site* : アロケータの処理対象となるサイト名称 (*site*) を指定します。このオプション省略時は、あらかじめ設定されたデフォルト（RSSITE）にします。

## &lt;注意事項&gt;

バリュ値 (*value*) は整数型データとして扱い、 $-2^{31} \leq \text{value} \leq 2^{31}-1$ 以外を指定した場合、エラーとなります。

なお、実行時のバリュ値は、使用言語のバリュ名称 (*ename*) のデータ型に依存します。

使用者種別と確保されるバリュの所有者種別の関係を以下に示します。

使用者種別	バリュの所有者種別
システム	システム
ユーザ	ユーザ

## &lt;終了コード&gt;

次の終了コードを返します。

0 : 正常終了

1 : オプションの指定に誤りがあります。

2 : 最大登録数を超過しています。

3 : 管理テーブルが更新できませんでした。

## 第2章 アロケータ

### <名前>

svdlv

### <形式>

svdlv ename [オプション]

### <機能説明>

svdlvは、svdfvで登録された外部参照情報を削除します。

### <引数説明>

ename : 削除する外部名称を指定します。

### <オプション>

- S : アクセス権がシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト（環境変数RSUTYP）にします。
- u site : アロケータの処理対象となるサイト名称（site）を指定します。このオプション省略時は、あらかじめ設定されたデフォルト（RSSITE）にします。

### <注意事項>

使用者種別と削除バリュの所有者種別の関係を以下に示します。

使用者種別	削除バリュの所有者種別	
	システム	ユーザ
システム	○	○
ユーザ	×	○

○ : 削除できます。 × : 削除できません。

### <終了コード>

次の終了コードを返します。

- 0 : 正常終了
- 1 : オプションの指定に誤りがあります。
- 2 : ユーザはシステムのVALを削除できません。
- 3 : 管理テーブルが更新できませんでした。

## 第3章 ローダ

### <名前>

svload

### <形式>

svload [オプション] file...

### <機能説明>

svloadは、オブジェクトファイル、ライブラリを結合し、プログラム、サブプログラム、データを指定した名称で開発環境に登録し、バックアップファイルに格納します。

### <引数説明>

file : 結合するオブジェクトファイル、ライブラリを指定します。ファイルは複数指定できます。

### <オプション>

- S : 処理モードがシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト（環境変数RSUTYP）として扱います。
- u site : ローダの処理対象となるサイト名称（site）を指定します。このオプション省略時は、あらかじめ設定されたデフォルト（RSSITE）として扱います。
- C n : プログラム、サブプログラムの格納先頭アドレス（n）を指定します。アドレスは、プログラムの場合4096の倍数を、サブプログラムの場合32の倍数を指定してください。4096または32の倍数以外を指定した場合、警告メッセージを表示後、4096または32の倍数に切り上げます。
- p n : エリア内のローディング位置を相対バイトアドレス（n）で指定します。プログラム、サブプログラムのときに有効です。このオプションを省略した場合は自動登録となります。-Cオプションと同時指定はできません。アドレスは、プログラムの場合4096の倍数を、サブプログラムの場合32の倍数を指定してください。4096または32の倍数以外を指定した場合、警告メッセージを表示後、4096または32の倍数に切り上げます。
- a area : プログラム、サブプログラムのローディングエリア（area）を指定します。プログラム、サブプログラムのときは、このオプションは省略できません。
- +P : プログラム（タスク）としてローディングすることを指定します。
- +I : 間接リンクサブルーチン（IRSUB）としてローディングすることを指定します。
- +U : 組み込みサブルーチン（ULSUB）としてローディングすることを指定します。
- +D : グローバルデータとしてローディングすることを指定します。データ種別は属する分割領域の属性に従います。
- M n : マルチタスクを生成する場合に指定します。nはマルチタスクの個数を示します。nは2～128の範囲としてください。

- m n[ n... ] : IRSUBのマルチエン트리ローディングの指定であり、エン트리ポイントとしたいエン트리名 (n) を指定します。このオプションは、+I指定にだけ指定できます。
- Z : ロードモジュールのtext、data、bss、stackセクションのサイズを出力します。このオプションはプログラムを登録しません。  
この指定と-Pオプションを組み合わせることにより、実際にロードせずにリンケージマップリストを生成することができます。  
また、-w n [m]オプションと組み合わせることにより、呼び出しているIRSUBが使用するスタックサイズが増加した場合に、呼び出し側の再ロード（スタックサイズの拡張）可否をチェックすることができます。この場合、n、mには前回登録時に指定した値と同じ値を指定してください。
- d : バックアップファイルに格納後、ロードモジュールファイルを削除しません。  
ロードモジュールファイルは、サイトディレクトリの下PGM、SUB、DATA下に生成されます。
- s : スタック使用量情報ファイルを生成します。スタック使用量情報ファイルは、サイトディレクトリ下にあるPGM/SUBディレクトリに生成されます。
- llib : 結合するライブラリ (lib) を指定します。なお、libcpms.libとlibsh4nbmdn.libは自動で結合します。
- P [file] : プログラムのリンケージマップリストを出力します。fileを省略した場合は、サイトディレクトリの下PGM、SUB、DATA下に、ロードするプログラム名、サブプログラム名の後ろに、.mapを付加したファイル名で生成します。初期値データのロードの場合は、ロードした先頭のsarea名.mapの名称で生成します。
- o obj : 作成するプログラム名 (obj) を指定します。サブプログラムの場合、指定した名前がサブプログラム名となります。サブプログラム名とプログラム内の関数は一致させてください。
- E n : プログラム、サブプログラムを結合時、冗長なバイト数 (n) を加味して格納します。  
これは、将来プログラム、サブプログラムを入れ替え、容量が増えた場合に有効です。
- r : 指定したエリアにプログラム、サブプログラムが入れ替え可能かどうかをチェックします。  
格納アドレスは入れ替え前のプログラム、サブプログラムと同じアドレスを指定してください。
- w n [m] : スタックエリアの大きさをバイト長で指定します。  
このオプションはプログラム、サブプログラムの場合、省略できません。  
nには自分自身が使用するスタックエリアの大きさを指定します。  
mには実際に確保するスタックサイズを指定します。mを省略するとスタックサイズは自分自身のスタックサイズ (n) に、呼び出しているIRSUBが使用する最大スタックサイズを加算した値で確保します。  
確保するスタックサイズ (m) が、n+呼び出しているIRSUBの最大スタックサイズより小さい場合はWarningメッセージを出力します。  
n、mは、0~8388608 (0x800000) かつ8の倍数を指定してください。8の倍数以外を指定した場合は、警告メッセージ表示後8の倍数に切り上げて処理します。  
組み込みサブルーチンのスタックサイズの最大値は512バイトです。

## &lt;ライブラリサーチパス&gt;

ローダのライブラリのサーチパス (-lオプションで指定されたライブラリのサーチ順) は、shcコンパイラパッケージの最適化リンケージエディタの入力ファイルの検索順序に従います。

最適化リンケージエディタの入力ファイルの検索順は下記となります。

- (1) カレントディレクトリ
- (2) 環境変数HLNK\_DIRで指定されたディレクトリ

環境変数HLNK\_DIRには複数のパスが設定できます。

複数のパスを設定する場合は、セミコロンで区切って指定してください。

## &lt;スタック容量&gt;

プログラムでスタックエリアを使用する場合、スタックエリアの容量を指定してください。

## &lt;システム/ユーザの外部参照チェック&gt;

システムからユーザの情報は参照できません。ユーザからシステムのサブプログラムだけ参照できます。

参照できる組み合わせを下表に示します。

参照 \ 被参照		サブプログラム		グローバル		バリュ	
		S	U	S	U	S	U
プログラム	S	○	×	○	×	○	×
	U	○	○	×	○	×	○
サブプログラム	S	○	×	○	×	○	×
	U	○	○	×	○	×	○
グローバル	S	○	×	○	×	○	×
	U	○	○	×	○	×	○

S : システム U : ユーザ

○ : 参照できます。 × : 参照できません。

(注) グローバルからサブプログラムを参照する場合、名称に対応する間接リンクテーブルの番号がグローバル内に埋め込まれます。グローバルからグローバルを参照する場合、絶対アドレスがグローバル内に埋め込まれます。グローバルからバリュを参照する場合、バリュ値がグローバル内に埋め込まれます。

<注意事項>

- IRSUBやマルチタスクは、リエントラントなプログラムであるため、BSSエリアを持つことはできません。BSSエリアを持つと、警告メッセージが出力されます。
- ロードするプログラム内に複数グローバルが存在する場合、ローカルなラベルのアドレスは解決されません。この場合、グローバルを複数のファイルに分割しローディングしてください。
- タスクはプログラムの先頭から実行されます。mainから実行されるとは限りません。プログラムのロード時は、メインルーチンのオブジェクトファイルを最初に指定してください。
- オプション組み合わせは、下記となります。

		-o obj	-a area	-w n	-s	-S	-u site	-C n	-p n	-M n	-d	-Z	-P file	-E n	-r	-m n	-l	
プログラム	+P	◎	◎	◎	○	○	○	○	○	○	○	○	○	○	○	○	×	○
IRSUB	+I	◎	◎	◎	○	○	○	○	○	×	○	○	○	○	○	○	○	○
ULSUB	+U	◎	◎	◎	○	○	○	○	○	×	○	○	○	○	○	○	×	○
データ	+D	◎	×	×	×	○	○	×	×	×	×	○	○	×	×	×	○	○

◎：必須指定 ○：選択指定 ×：指定不可

<終了コード>

次の終了コードを返します。

0 : 正常終了

0以外：異常終了

### <スタック容量算出方法>

スタックエリアの使用量は、プログラムを構成する各関数のスタック使用量を算出し、関数の呼び出し関係から全体のスタック使用量を算出します。

#### (1) 各関数の使用するスタック領域の算出

各関数の使用するスタック領域の大きさは、コンパイラが出力するオブジェクトリストのframe sizeから分かります。

以下に具体例を示します。

#### ■ ソースコード

```
extern int h(char , int *, double);
int h(char a, register int *b, double c)
{
    char *d;

    d = &a;
    h(*d, b, c);
    {
        register int i;

        i = *d;
        return i;
    }
}
```

#### ■ オブジェクトリスト

SCT	OFFSET	CODE	C LABEL	INSTRUCTION	OPERAND	COMMENT
P	00000000		_h:			; function: h ; <b><u>framesize=12</u></b>
	00000000	2FE6		MOV.L	R14, @-R15	
	00000002	4F22		STS.L	PR, @-R15	

上記の例では関数hの使用するスタック領域のサイズは、オブジェクトリスト中の項目“COMMENT”の“frame size”の値12バイトとなります。

(2) 呼び出し関係からの全体スタック容量の算出

関数呼び出しの関係から使用するスタック領域のサイズを算出します。

関数呼び出しの関係からのスタック使用量算出方法の例を図2-1に示します。

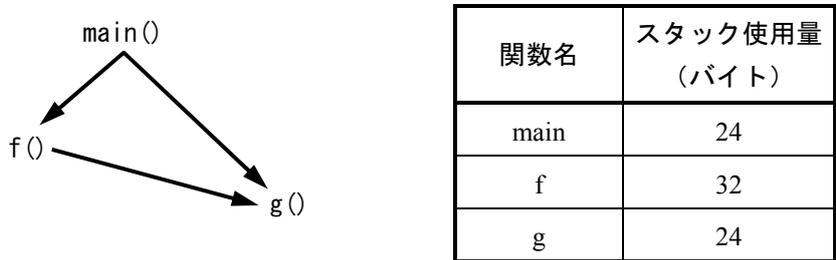


図2-1 関数呼び出しの関係とスタック使用量

上記の場合、関数fを介して関数gが呼ばれた場合のスタック領域のサイズは、表2-4に示すようになります。

表2-4 スタックサイズの計算例

呼び出し経路	スタックサイズ (バイト)
main(24)→f(32)→g(24)	80
main(24)→g(24)	48

このように、呼び出しレベルの一番深い関数についてスタック容量のサイズを計算し、その最大サイズのスタック領域を最低限割り当てなければなりません。

標準ライブラリ関数を使用する場合には、ライブラリ関数を使用するスタックサイズも考慮する必要があります。標準ライブラリ関数の使用するスタックサイズは、「付録H ライブラリの使用するスタックサイズ一覧」を参照してください。

再帰的に呼び出される関数のスタックサイズは、「関数のスタックサイズ×再帰的に呼び出される回数の最大値」で算出してください。

また、ソースプログラム上ライブラリ関数を使用していなくても、プログラムの実行に必要な実行時ルーチンがリンクされている場合があります。実行時ルーチンの使用するスタックサイズは、次ページのスタック使用量確認方法に示すスタック解析ツールで確認することができます。

### <スタック使用量確認方法>

プログラム、サブプログラムのロード時に-sオプションを指定すると、スタック使用量情報ファイルを作成することができます。

ローダで生成したスタック使用量情報ファイルをコンパイラパッケージ付属のスタック解析ツールで解析することにより、プログラム、サブプログラムの全体スタック使用量を求めることができます。

#### ● スタック使用量情報ファイルの生成

プログラム、サブプログラムのロード時に-sオプションを指定すると、スタック使用量情報ファイルを作成します。

スタック使用量情報ファイルは、サイトディレクトリ下にあるPGM/SUBディレクトリ下に、プログラム、サブプログラムの名称に、“\_sni”を付加したファイル名で生成します。

(例)

```
svload +P -o pgm01 -a tskarea -w 4096 pgm01.obj -s
```

上記の例では、サイトディレクトリ¥PGM¥pgm01\_sniが生成されます。

#### ● スタック解析ツール使用方法

以下に示す操作を行うことによりスタック解析ツールを起動し、プログラム、サブプログラムのスタック使用量を表示することができます。

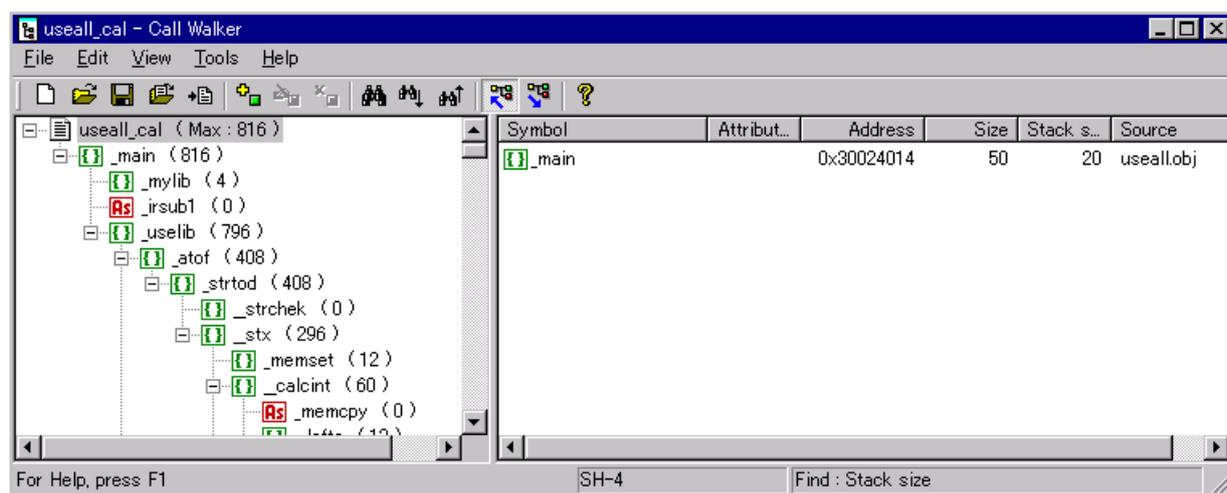
スタック解析ツールの使用方法の詳細は、コンパイラパッケージのマニュアルおよびスタック解析ツールのヘルプを参照してください。

① SHCのVerが7の場合には、Windows®の  ボタンから [Hitachi Embedded Workshop 2] - [Hitachi Call Walker] または [Renesas High-performance Embedded workshop] - [Call Walker] を選択し、スタック解析ツールを起動します。

SHCのVerが9の場合には、Windows®の  ボタンから [Renesas] - [High-performance Embedded Workshop] - [Call Walker] を選択し、スタック解析ツールを起動します。

② スタック解析ツールの [File] メニューから [Import Stack file...] を選択し、表示されたダイアログボックスの“ファイル名”にローダで生成したスタック使用量情報ファイルを指定し、 ボタンをクリックします。

#### ■ スタック解析ツールの表示例



● ロードしたプログラム、サブプログラムのスタック使用量解析時の注意点

スタック解析ツールで算出するスタック使用量では、アセンブラで記述されたプログラム、サブプログラムのスタックサイズは0バイトとして表示されます。このため、RPDPでロードしたプログラム、サブプログラムのスタック使用量をスタック解析ツールで解析する場合には、以下に示す注意事項があります。

■ memcpy()のスタックサイズ

ロードしたプログラム、サブプログラムでは、memcpy()関数はC標準ライブラリではなくCPMSライブラリのmemcpy()関数がリンクされます。スタック解析ツールではCPMSライブラリのmemcpy()が使用するスタックサイズは0として表示されますが、CPMSライブラリのmemcpy()はスタックを28バイト使用します。

スタック解析ツールの [Edit] メニューから [Modify] コマンドを使用してmemcpy()関数のスタックサイズを28バイトに変更し、スタック使用量を再計算してください。

ロードしたプログラム、サブプログラムがmemcpy()を使用しているか否かは、スタック解析ツールの検索機能でmemcpyを検索することで識別することができます。

■ IRSUBのスタックサイズ

ロードしたプログラム、サブプログラムからIRSUBを呼び出している場合、スタック解析ツールでは、IRSUBが使用するスタックは0バイトとして算出されます。呼び出しているIRSUBが使用するスタックサイズを含んだスタック使用量を算出する場合は、memcpy()と同様に、スタック解析ツールの [Edit] メニューから [Modify] コマンドを使用してスタックサイズを変更し、スタック使用量を再計算してください。

IRSUBが使用するスタックサイズは、IRSUB自身のロード時にスタック使用量情報ファイルを生成し、スタック解析ツールで解析することによって求めることができます。

ローダに指定する自分自身が使用する（呼び出しているIRSUBを含まない）スタックサイズを求めるためには、IRSUBのスタックサイズの再計算は必要ありません。

■ その他アセンブラで記述されたプログラム、サブプログラムのスタックサイズ

アセンブラで記述されたプログラム、サブプログラムのスタックサイズも同様に、スタック解析ツールの [Edit] メニューから [Modify] コマンドを使用してスタックサイズを変更し、スタック使用量を再計算してください。

## (3) svloadへのスタックサイズの指定方法

svloadで指定するスタックサイズは下記としてください。

`-w n [m]`

**n** : ロードするプログラム、サブプログラム自身が使用するスタックサイズ

**m** : ロードするプログラム、サブプログラムから呼び出しているIRSUBが使用するスタックサイズを含んだサイズ。プログラムのロード時にこの値を指定すると、指定サイズでスタックを確保します。この値を省略すると、**n**+呼び出しているIRSUBが使用するスタックサイズの値でスタックを確保します。

**m**の値に**n**+呼び出しているIRSUBが使用するスタックサイズの値よりも小さい値を指定した場合は、ローダは下記のWarningメッセージを出力します。

Warning : Stack size (name) = xxxx (zzzz) byte [Max refered (sname) size = yyyy byte] Err

**name** : ロードするプログラムまたはサブプログラム名

**xxxx** : 呼び出しているIRSUBが使用するスタックサイズを含んだサイズ

**zzzz** : **m**に指定したスタックサイズ

**sname** : 呼び出しているIRSUBの名称 (最大スタックサイズのもの)

**yyyy** : **sname**の使用するスタックサイズ

それぞれのプログラム、サブプログラムが使用するスタックサイズは、「(2) 呼び出し関係からの全体スタック容量の算出」に示す手順に従って算出してください。

以下に「図2-1 関数呼び出しの関係とスタック使用量」に示す、呼び出し関係のプログラムを例にsvloadに指定するスタックサイズの指定例を示します。

● IRSUBを使用していない場合

**main** : プログラムのメイン

**f** : IRSUB

**g** : IRSUB

の場合は下記のように指定します。

svload +P -o main ..... -w 80 4096 .....

ロードするプログラム自身が使用するスタックサイズの最大値は80バイトであるため、**n**の値には80を指定します。

**m**には実際に確保するスタックのサイズを指定してください。**m**の値を省略するとスタックサイズは80バイトとなります。

プログラムの改修時に使用するスタックサイズが増加したり、IRSUBを呼び出すようになった場合に備え、余裕を持って指定することを推奨します。

プログラムのスタック/BSSは、テキスト/データとは別ページに配置されるため、スタックサイズを大きめに指定しても、BSSサイズ+スタックサイズが4096バイトの境界を超えなければプログラムサイズは増加しません。

● IRSUBを使用している場合

main : プログラムのメイン

f : ISUB

g : IRSUB

の場合は下記のように指定します。

```
svload +I -o g ..... -w 24 .....  
svload +P -o main ..... -w 56 4096 .....
```

IRSUBとプログラムは別々にロードします。スタックサイズはIRSUB、プログラムの両方に指定します。

• IRSUBのロード時

IRSUB (g) が使用するスタックサイズは24バイトであるため、スタックサイズには24バイトを指定します。mの値は省略することを推奨します。mの値を指定した場合は、IRSUBの呼び出し側で計算するIRSUBのスタックサイズにmが使用されます。

• プログラムのロード時

プログラム自身が使用するスタックサイズは56バイトであるため、nの値には56を指定します。

mの値は56と呼び出しているIRSUB (g) の使用するスタックサイズ24を加算した80バイト以上を指定してください。IRSUBを使用しない場合と同様に、プログラムの改修に備えてmの値は余裕を持って指定することを推奨します。

mの値に80バイトよりも小さい値を指定した場合は、ローダは下記のWarningメッセージを出力します。

```
Warning : Stack size (main) = 80 (m) byte [Max refered (g) size = 24 byte] Err
```

mを省略すると、ローダがIRSUB (g) が使用するスタックサイズ24を加算し、80バイトでスタックを確保します。

## ● IRSUBからIRSUBを使用している場合

main : プログラムのメイン

f : IRSUB

g : IRSUB

の場合は下記のように指定します。

```
svload +I -o g ..... -w 24 .....
svload +I -o f ..... -w 32 .....
svload +P -o main ..... -w 24 4096 .....
```

- IRSUB (g) のロード時

IRSUB (g) が使用するスタックサイズは24バイトであるため、スタックサイズには24バイトを指定します。mの値は省略することを推奨します。mの値を指定した場合は、IRSUBの呼び出し側で計算するIRSUBのスタックサイズにmが使用されます。

- IRSUB (f) のロード時

IRSUB (f) が使用するスタックサイズは32バイトであるため、スタックサイズには32バイトを指定します。mの値を省略した場合、IRSUBの呼び出し側で計算するIRSUB (f) のスタックサイズは、IRSUB (f) から呼び出しているIRSUB (g) のスタックサイズを加算した56バイトとなります。mの値を指定した場合は、IRSUBの呼び出し側で計算するIRSUB (f) のスタックサイズはmが使用されます。mの値は省略することを推奨します。

mの値に56バイトよりも小さい値を指定した場合は、ローダは下記のWarningメッセージを出力します。

```
Warning : Stack size (f) = 56 (m) byte [Max refered (g) size = 24 byte] Err
```

- プログラムのロード時

プログラム自身が使用するスタックサイズは24バイトであるため、nの値には24を指定します。

mの値は、24と呼び出している最大スタック容量のIRSUB (f) の使用するスタックサイズ56を加算した80バイト以上を指定してください。IRSUBを使用しない場合と同様に、プログラムの改修に備えてmの値は余裕を持って指定することを推奨します。

mの値に80バイトよりも小さい値を指定した場合は、ローダは下記のWarningメッセージを出力します。

```
Warning : Stack size (main) = 80 (m) byte [Max refered (f) size = 56 byte] Err
```

mを省略すると、ローダがIRSUB (f) が使用するスタックサイズ56を加算し、80バイトでスタックを確保します。

<ライブラリの整合性チェック>

コンパイル時に`-denormalization=off-round=zero`オプション指定でコンパイルした場合は、ロード時にライブラリ`libsh4nbmzz.lib(-lsh4nbmzz)`を指定しなければなりません。`libsh4nbmzz.lib`を指定しない場合、ローダは下記のエラーメッセージを出力します。

```
svload : Error   : Undefined symbols
svload :          _use_libsh4nbmzz
```

同様に、コンパイル時に`-denormalization=off-round=zero`オプション指定でコンパイルしていないときに、`libsh4nbmzz.lib`が指定された場合は、下記のエラーメッセージを出力します。

```
svload : Error   : Undefined symbols
svload :          _use_libsh4nbmdn
```

また、両方のオブジェクトを混在させて、`-lsh4nbmzz`と`-lsh4nbmdn`の両方を指定した場合は、下記のエラーメッセージを出力します。

```
svload : rpdpload: Inconsistent object was mixed (NO:2004-25)
```

※上記の整合性チェックを行うためには、コンパイル時に“`rpdp.h`”をインクルードしてコンパイルする必要があります。

詳しくは「第1編 概説」の「4. 2 コンパイル時の注意点」を参照してください。

## &lt;名前&gt;

svdload

## &lt;形式&gt;

svdload pname [オプション]

## &lt;機能説明&gt;

svdloadは、svloadコマンドで登録したプログラム、サブプログラムを、開発環境下から削除します。ただし、バックアップファイルの0クリアはしません。

## &lt;引数説明&gt;

pname : 削除するプログラム、サブプログラム名称を指定します。

## &lt;オプション&gt;

- S : 処理モードがシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト（環境変数RSUTYP）として扱います。
- u site : ローダの処理対象となるサイト名称（site）を指定します。このオプション省略時は、あらかじめ設定されたデフォルト（RSSITE）として扱います。
- +P : プログラム（タスク）の削除を指定します。
- +I : 間接リンクサブルーチン（IRSUB）の削除を指定します。
- +U : 組み込みサブルーチン（ULSUB）の削除を指定します。

## &lt;終了コード&gt;

次の終了コードを返します。

- 0 : 正常終了
- 0以外 : 異常終了

### 第3章 ローダ

#### <名前>

svcomp

#### <形式>

svcomp [オプション] file...

#### <機能説明>

svcompはローダで登録済みのプログラム、サブプログラム、データの、バックアップファイルの内容とロードモジュールを比較して、その結果を出力します。

#### <引数説明>

file : 結合するオブジェクトファイル、ライブラリを指定します。ファイルは複数指定できます。

#### <オプション>

- S : 処理モードがシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト（環境変数RSUTYP）として扱います。
- u site : ローダの処理対象となるサイト名称（site）を指定します。このオプション省略時は、あらかじめ設定されたデフォルト（RSSITE）として扱います。
- C n : svcompでは意味を持ちません。
- p n : svcompでは意味を持ちません。
- a area : svcompでは意味を持ちません。
- +P : プログラム（タスク）としてローディングしたものを比較する指定です。
- +I : 間接リンクサブルーチン（IRSUB）としてローディングしたものを比較する指定です。
- +U : 組み込みサブルーチン（ULSUB）としてローディングしたものを比較する指定です。
- +D : グローバルデータとしてローディングしたものを比較する指定です。  
データ種別は属する分割領域の属性に従います。
- M n : svcompでは意味を持ちません。
- m n[ n... ] : svcompでは意味を持ちません。
- Z : svcompでは意味を持ちません。
- d : svcompでは意味を持ちません。
- llib : 結合するライブラリ（lib）を指定します。なお、libcpms.libとlibsh4nbmdn.libは自動で結合します。
- P [file] : svcompでは意味を持ちません。
- o obj : 作成するプログラム名（obj）を指定します。サブプログラムの場合、指定した名前がサブプログラム名となります。サブプログラム名とプログラム内の関数は一致させてください。
- E n : svcompでは意味を持ちません。
- r : svcompでは意味を持ちません。
- w n [m] : svcompでは意味を持ちません。

## &lt;ライブラリサーチパス&gt;

ローダのライブラリのサーチパス (-lオプションで指定されたライブラリのサーチ順) は、shcコンパイラパッケージの最適化リンケージエディタの入力ファイルの検索順序に従います。

最適化リンケージエディタの入力ファイルの検索順は下記となります。

- (1) カレントディレクトリ
- (2) 環境変数HLNK\_DIRで指定されたディレクトリ

環境変数HLNK\_DIRには複数のパスが設定できます。

複数のパスを設定する場合は、セミコロンで区切って指定してください。

## &lt;終了コード&gt;

次の終了コードを返します。

- 0 : 相違なし
- 1 : 相違あり
- 101 : コマンドオプション指定あり
- 上記以外 : コンペア失敗

## &lt;svcompの表示結果&gt;

svcompコマンドは、比較結果に相違がない場合とある場合で、それぞれ下記のメッセージを出力します。

## ● 比較結果に相違がない場合

svcompコマンドは、比較結果に相違がない場合、下記のメッセージを出力します。

compare OK (type = X name = xxx)

X : 比較するリソースのタイプを表します。

pgm : プログラム

irsub : IRSUB

ulsub : 組み込みサブプログラム

data : データ (GLB)

xxx : 比較するリソースの名称を表します。

データの比較の場合は、dataと表示します。

## ● 比較結果に相違がある場合

svcompコマンドは、比較結果に相違がある場合、下記のメッセージを出力します。

## &lt;表示フォーマット&gt;

svcompコマンドの比較結果の相違点表示内容は、以下のフィールドから構成されます。

- (1) ヘッダー
- (2) 詳細情報
- (3) フッター

プログラム、サブプログラムの場合のフォーマットを図2-2に示します。データの場合のフォーマットを図2-3に示します。斜体で表記している部分は、実行環境によって変わる部分です。

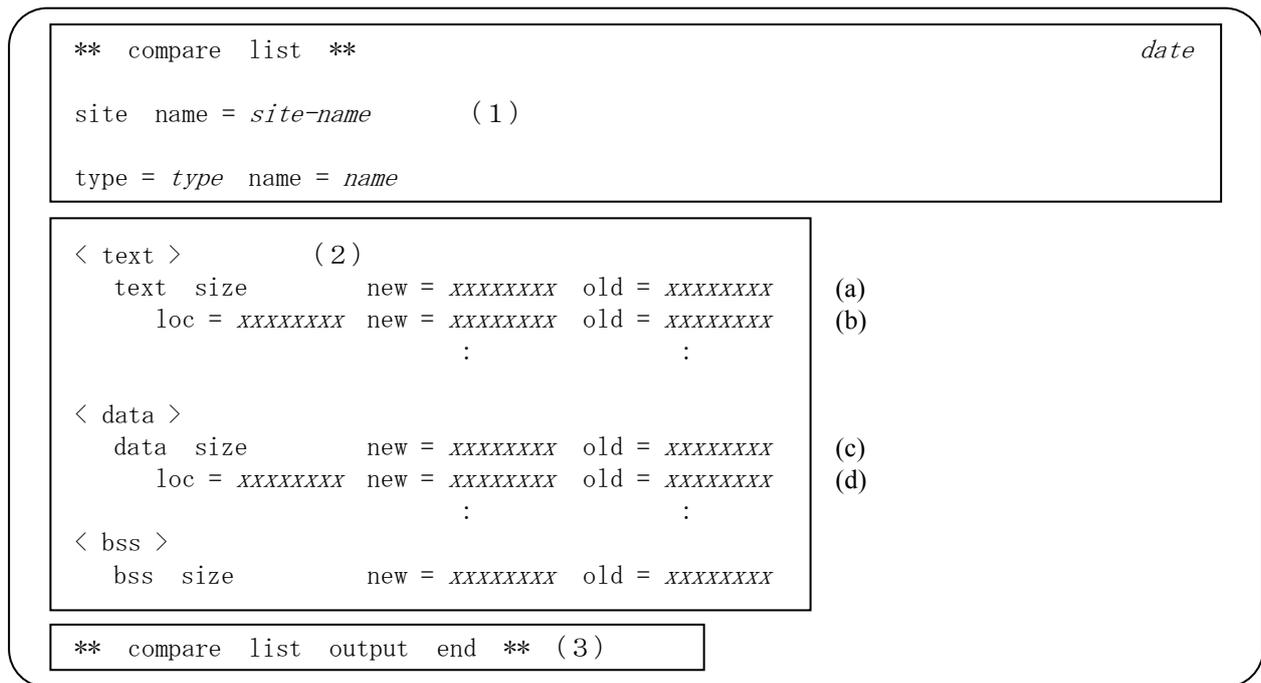


図2-2 svcomp (プログラム、サブプログラム) のフォーマット

各フィールドの詳細は以下のとおりです。

(1) ヘッダーフィールド

- date* : svcompコマンドを起動した時刻を表示します。
- site-name* : 処理対象としているサイト名称を表示します。
- type* : 比較対象の種別を示します。
  - pgm* - タスク用プログラム
  - irsub* - 間接リンクサブルーチン
  - ulsub* - 組み込みサブルーチン
- name* : 比較対象の名称です。

(2) 詳細情報フィールド

このフィールドには、テキスト、データ、BSSそれぞれのサイズの新/旧比較結果と、テキスト、データの内容の新/旧比較結果を表示します。新/旧サイズが異なる場合の比較は、小さい方のサイズにあわせて行います。

- (a)、(c) : テキスト、データ、BSSのサイズの新/旧比較結果を16進数で表示します。比較結果が同じである場合には表示しません。
- (b)、(d) : テキスト、データの内容の新/旧比較結果を16進数で表示します。比較結果が同じである場合には表示しません。loc = の直後表示される16進数は、比較内容が異なっているアドレスであり、テキスト、データ先頭からの相対値です。

(3) フッターフィールド

比較完了を示します。

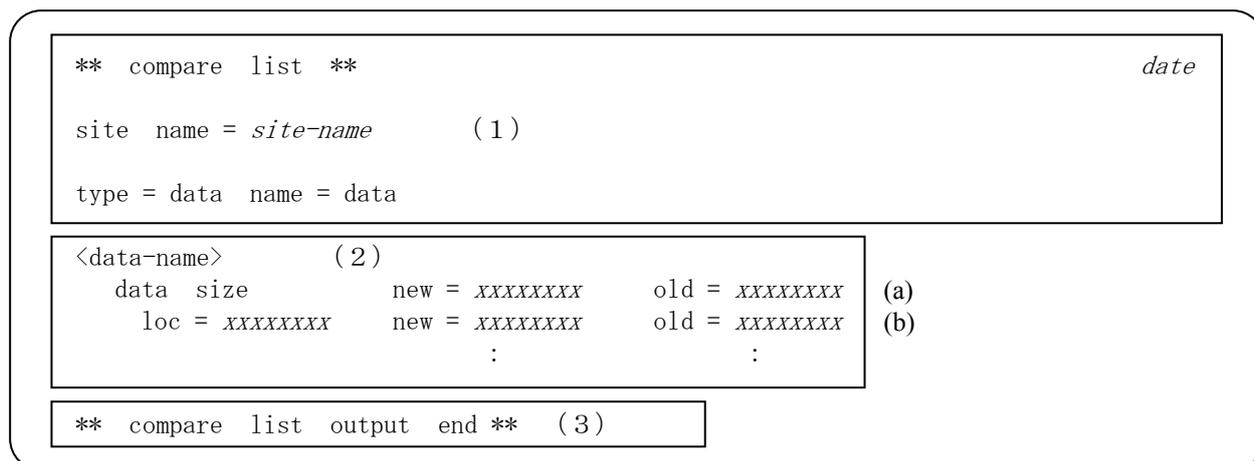


図2-3 svcomp (GLB) のフォーマット

各フィールドの詳細は以下のとおりです。

(1) ヘッダーフィールド

*date* : svcompコマンドを起動した時刻を表示します。

*site-name* : 処理対象としているサイト名称を表示します。

(2) 詳細情報フィールド

このフィールドには、GLBのサイズの新/旧比較結果と、テキスト、データの内容の新/旧比較結果を表示します。新/旧サイズが異なる場合の比較は、小さい方のサイズにあわせて行います。

(a) : GLBのサイズの新/旧比較結果を16進数で表示します。比較結果が同じである場合には表示しません。

(b) : GLBの内容の新/旧比較結果を16進数で表示します。比較結果が同じである場合には表示しません。loc =の直後に表示される16進数は、比較内容が異なっているアドレスであり、データ先頭からの相対値です。

(3) フッターフィールド

比較完了を示します。

## 第4章 ビルダ

### <名前>

svctask

### <形式>

svctask pname tname tn [オプション]

### <機能説明>

svctaskは、ローダで格納されたプログラムをリソースとして、タスクを生成します。

### <引数説明>

**pname** : 生成すべきタスクのリソースとなるプログラム名を指定します。

**tname** : 生成すべきタスク名を指定します。

**tn** : タスク番号を指定します。ユーザタスクの場合は1~224まで、システムタスクの場合は225~229までが指定できます。指定タスク番号が使用中の場合はエラーとなります。230~255までは、システム提供タスク (ROM格納) として予約されているため、登録できません。

### <オプション>

**-S** : 処理モードがシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト (環境変数RSUTYP) として扱います。

**-u site** : ビルダの処理対象となるサイト名称 (site) を指定します。このオプション省略時は、あらかじめ設定されたデフォルト (RSSITE) として扱います。

**-l lvl** : タスクの初期起動時の実行レベル (lvl) を指定します。ユーザタスクの場合は4~27まで、システムタスクの場合は0~31までが指定できます。このオプション省略時、ユーザタスクの場合はlvl=27、システムタスクの場合はlvl=0が指定されたものとみなします。

**-r n** : リソースとなるプログラムから複数のタスクを生成する場合の、使用スタックエリア番号を指定します。この数値は、ロードコマンドで指定するマルチタスクの個数 (-M n) を超えて指定することはできません。このオプション省略時は、未使用のスタックエリア番号の最小値が指定されたものとみなします。

### <注意事項>

ユーザタスクとはTN=1~224で登録したタスクを、システムタスクとはTN=225~229で登録したタスクを示します。

RSUTYP=sで登録する場合、システムタスクを生成することができます。

RSUTYP=uで登録する場合、-Sオプションを指定することでシステムタスクが生成できます。

RSUTYP=uで-Sオプションを指定しない場合、システムタスクは生成できません。

タスク種別とオプションの組み合わせを次ページに示します。

## &lt;終了コード&gt;

次の終了コードを返します。

0 : 正常終了

0以外 : 異常終了

タスク種別 パラメータ		シングルタスク	マルチタスク
		pname	◎
tname		◎	◎
tn		◎	◎
オプション	-u site	○	○
	-l lvl	○	○
	-S	○	○
	-r n	×	◎

◎ : 必須 ○ : 指定できます。× : 指定できません。

使用者種別と、プログラム所有者種別との関係を以下に示します

使用者種別	プログラム所有者種別	
	システム	ユーザ
システム	○	○
ユーザ	×	○

○ : タスク生成できます。× : タスク生成できません。

## 第4章 ビルダ

### <名前>

svdtask

### <形式>

svdtask tname [オプション]

### <機能説明>

svdtaskは、svctaskで生成したタスクを削除します。

### <引数説明>

tname : 削除すべきタスク名を指定します。

### <オプション>

- S : 処理モードがシステムであることを指定します。このオプション省略時のアクセス権は、あらかじめ設定されたデフォルト（環境変数RSUTYP）として扱います。
- u site : ビルダの処理対象となるサイト名称（site）を指定します。このオプション省略時は、あらかじめ設定されたデフォルト（RSSITE）として扱います。

### <注意事項>

使用者種別と、削除タスクの所有者種別との関係を以下に示します。

使用者種別	削除タスク所有者種別	
	システム	ユーザ
システム	○	○
ユーザ	×	○

○ : タスク削除できます。 × : タスク削除できません。

### <終了コード>

次の終了コードを返します。

0 : 正常終了

0以外 : 異常終了

## &lt;名前&gt;

svbuild (間接リンクサブプログラムの登録)

## &lt;形式&gt;

svbuild name -ir -e irno [オプション]

## &lt;機能説明&gt;

svbuildは、間接リンクサブプログラムを登録します。

## &lt;引数説明&gt;

name : 間接リンクサブプログラム名称 (マルチエン트리ローディング時はエン트리名称) を指定します。

-ir -e irno : 間接リンクサブプログラムの登録番号 (irno) を指定します。

## &lt;オプション&gt;

-u site : ビルダの処理対象となるサイト名称 (site) を指定します。このオプション省略時は、あらかじめ設定されたデフォルト (RSSITE) として扱います。

パラメータ		種別	間接リンクサブプログラム
name -ir -e irno			◎
オプション	-u site		○

◎ : 必須 ○ : 指定できます。

## &lt;終了コード&gt;

次の終了コードを返します。

0 : 正常終了

0以外 : 異常終了

## 第4章 ビルダ

### <名前>

svbuild (組み込みサブルーチンの登録)

### <形式>

svbuild name point en -ul [オプション]

### <機能説明>

svbuildは、組み込みサブルーチンを登録します。

### <引数説明>

name : 組み込みサブルーチン名称を指定します。

point en -ul : 組み込みサブルーチンの組み込み箇所 (point) およびエントリ番号 (en) を指定します。エントリ番号として1~4を、組み込み箇所には以下の文字列を指定してください。

CPES : CPES組み込みサブルーチンを示します。

IES : IES組み込みサブルーチンを示します。

EAS : EAS組み込みサブルーチンを示します。

INS : INS組み込みサブルーチンを示します。

EXS : EXS組み込みサブルーチンを示します。

ABS : ABS組み込みサブルーチンを示します。

PCKS : PCKS組み込みサブルーチンを示します。

MODES : MODES組み込みサブルーチンを示します。

WDTES : WDTES組み込みサブルーチンを示します。

ADTS : ADT例外発生時にリンクするサブルーチンを示します。

### <オプション>

-u site : ビルダの処理対象となるサイト名称 (site) を指定します。このオプション省略時は、あらかじめ設定されたデフォルト (RSSITE) として扱います。

### <注意事項>

同一のプログラムを複数の組み込み箇所 (point) に組み込むことはできますが、複数のエントリ番号 (en) に組み込むことはできません。

オプションの組み合わせを以下に示します。

パラメータ		種別	組み込みサブ ルーチン
name point en -ul			◎
オプション	-u site		○

◎ : 必須 ○ : 指定できます。

### <終了コード>

次の終了コードを返します。

0 : 正常終了

0以外 : 異常終了

## &lt;名前&gt;

svdbuild (間接リンクサブプログラムの削除)

## &lt;形式&gt;

svdbuild name -ir [オプション]

## &lt;機能説明&gt;

svdbuildは、間接リンクサブプログラムを削除します。

## &lt;引数説明&gt;

name : 間接リンクサブプログラム名称 (マルチエントリローディング時はエントリ名称) を指定します。

-ir : 間接リンクサブプログラムの削除指定です。

## &lt;オプション&gt;

-u site : ビルダの処理対象となるサイト名称 (site) を指定します。このオプション省略時は、あらかじめ設定されたデフォルト (RSSITE) として扱います。

パラメータ		種別	間接リンクサブプログラム
name -ir			◎
オプション	-u site		○

◎ : 必須 ○ : 指定できます。

## &lt;終了コード&gt;

次の終了コードを返します。

0 : 正常終了

0以外 : 異常終了

## 第4章 ビルダ

### <名前>

svdbuild (組み込みサブルーチンの削除)

### <形式>

svdbuild name point -ul [オプション]

### <機能説明>

svdbuildは、組み込みサブルーチンを削除します。

### <引数説明>

name : 組み込みサブルーチン名称を指定します。

point -ul : 組み込みサブルーチンの削除指定です。

組み込み箇所 (point) には、以下の文字列を指定してください。

CPES : CPES組み込みサブルーチンを示します。

IES : IES組み込みサブルーチンを示します。

EAS : EAS組み込みサブルーチンを示します。

INS : INS組み込みサブルーチンを示します。

EXS : EXS組み込みサブルーチンを示します。

ABS : ABS組み込みサブルーチンを示します。

PCKS : PCKS組み込みサブルーチンを示します。

MODES : MODES組み込みサブルーチンを示します。

WDTES : WDTES組み込みサブルーチンを示します。

ADTS : ADT例外発生時にリンクするサブルーチンを示します。

### <オプション>

-u site : ビルダの処理対象となるサイト名称 (site) を指定します。このオプション省略時は、あらかじめ設定されたデフォルト (RSSITE) として扱います。

### <注意事項>

オプションの組み合わせを以下に示します。

パラメータ \ 種別		組み込みサブルーチン
name point -ul		◎
オプション	-u site	○

◎ : 必須 ○ : 指定できます。

### <終了コード>

次の終了コードを返します。

0 : 正常終了

0以外 : 異常終了

## &lt;名前&gt;

svirglb

## &lt;形式&gt;

svirglb idxnum name [オプション]

## &lt;機能説明&gt;

svirglbは、svdfsで確保した細分割領域を間接リンクグローバルとして登録または削除します。

## &lt;引数説明&gt;

idxnum : 間接リンクグローバルテーブルの登録番号を指定します (1~4096の範囲で指定できます)。

name : 間接リンクグローバル名称を指定します (-s、-aを指定しない場合には間接リンクグローバル名称は、svdfsで確保済みの細分割領域名称を指定します)。

## &lt;オプション&gt;

-u site : ビルダの処理対象となるサイト名称 (site) を指定します。このオプション省略時は、あらかじめ設定されたデフォルト (RSSITE) として扱います。

-s name : 間接リンクグローバルテーブルに格納するアドレスとして、間接リンクグローバル+オフセットで格納する場合に、細分割領域名称を指定します。

-o offset : 間接リンクグローバルテーブルに格納するアドレスとして、間接リンクグローバル+オフセットで格納する場合に、オフセット (offset) を16進数または10進数で指定します。0xで始めると16進数として扱います。

-a : 間接リンクグローバルテーブルに格納するアドレスを16進数または10進数の絶対アドレスで指定します。0xで始めると16進数として扱います。

-d : 間接リンクグローバルテーブルから指定した登録番号を削除します。

## 第4章 ビルダ

### <注意事項>

- -s、-oオプションは同時に指定してください。
- -s、-oオプションと-aオプションを同時に指定するとエラーになります。  
オプションの組み合わせを以下に示します。

パラメータ		種別	間接リンクグローバル	
			登録	削除
idxnum			◎	◎
name			◎	◎
オプション	-u site		○	○
	-s name		○	○
	-o offset		○	○
	-a		○	○
	-d		×	◎

◎：必須 ○：指定できます。×：指定できません。

- -aオプションにおいて指定できるアドレスは、GLB空間内のアドレスになります。
- -sオプションにおいて指定した細分割領域を含む分割領域の範囲外にオフセットを指定するとエラーとなります。

### <終了コード>

次の終了コードを返します。

- 0 : 正常終了
- 0以外 : 異常終了

## 第5章 マップ

## &lt;名前&gt;

svmap

## &lt;形式&gt;

svmap [オプション]

## &lt;機能説明&gt;

svmapは、RPDPで管理しているリソースの情報を出力します。

## &lt;オプション&gt;

- u sitename : マップ情報を出力するサイト名称を指定します。
- G : 大分割領域の情報を出力します。
- a : 分割領域の情報を出力します。
- e : 細分割領域の情報を出力します。
- p : プログラム情報を出力します。  
プログラム・タスク情報を名前順にソートして出力します。
- s : サブプログラム情報を出力します。
- t : タスク情報を出力します。  
プログラム・タスク情報をタスク番号順にソートして出力します。
- g : グローバル (GLB) 情報を出力します。
- v : VAL情報を出力します。
- irs : IRSUBのエントリ情報を出力します。
- irg : IRGLBのエントリ情報を出力します。
- uls : ULSUBのエントリ情報を出力します。
- fm : 物理メモリの空き情報を出力します。
- CON : PCsメモリ上のマップを出力します。  
このオプション省略時は、開発系側で管理するリソースのマップ情報を出力します。
- help : コマンドの起動形式を説明するリストを表示します。
- +a : 結果をアドレス順にソートして出力します。
- +n : 結果を名称順にソートして出力します。
- +e : 結果をエントリ番号順にソートして出力します。

+gn {name|num|pnt,typ,ent} :

指定されたリソースに関する情報を出力します。

-irs、-irgオプションと組み合わせて指定する場合は、エントリ番号を指定してください。

-ulsオプションと組み合わせて指定する場合は、pnt、typ、ent（組み込みポイントとエントリ番号）を指定してください。

-irs、-irg、-uls以外のオプションと組み合わせて指定する場合は、名称を指定してください。

-f : 詳細情報を表示します。

-en : IRSUB、GLB、VALのエントリ情報だけを出力します。

+gn nameと-G、-a、-eオプションを組み合わせることで、garea/areaの階層マップを出力することができます。オプションの組み合わせと出力結果の対応は表2-5を参照してください。

-uオプション以外のすべてのオプションを省略すると、分割領域・細分割領域のアドレス順リスト、プログラム・サブプログラム・タスク・グローバルの名称順リスト、IRSUB・IRGLBの番号順リスト、VALの名称順リストを出力します。

<終了コード>

次の終了コードを返します。

0 : 正常終了

0以外 : 異常終了

表2-5 オプションの組み合わせと出力内容

出力内容	オプション														
	+gn	+a	+n	+e	-G	-a	-e	-p	-s	-t	-g	-v	-irs	-irg	-uls
指定garea/areaの階層マップ (アドレス順)	○	(○)			○	○									
	○	(○)			○	○	○								
	○	(○)				○	○								
指定garea/areaの階層マップ (名称順)	○		○		○	○									
	○		○		○	○	○								
	○		○			○	○								
指定名称の情報出力	○				●	●	●	●	●	●	●	●	●	●	●
指定情報のアドレス順リスト		○			◎	◎	◎	×	×	×	×	×	×	×	×
指定情報の名称順リスト			○		◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
指定情報の番号順リスト				○	×	×	×	×	◎	◎	◎	×	◎	◎	◎
指定情報のリスト (デフォルト順) (*)					+a	+a	+a	+n	+n	+e	+n	+n	+e	+e	+e

○ : 指定要 (○) : 指定要 (省略可) ● : 指定可 (重複不可) ◎ : 指定可 (重複可) × : 指定不可

(\*) 表中の+a、+n、+eは、ソート順指定オプション省略時のデフォルトを表します。

## 第5章 マップ

### <名前>

svadm

### <形式>

svadm [addr] [オプション]

### <機能説明>

svadmは、指定した論理アドレスに登録されているリソース（グローバル、IRSUB）の名称等（詳細は<表示フォーマット>参照）の情報を出力します。

論理アドレスを省略した場合は、会話形式で論理アドレスを取り込み、名称等の情報を出力します。

### <引数説明>

addr：論理アドレス（addr）を指定します。省略時は、会話形式で論理アドレスを取り込みます。

アドレスの範囲は、0x30000000～0x7bffffff（タスク空間、GLB空間、サブプログラム空間）です。

### <オプション>

-u site：論理アドレス情報の表示対象となるサイト名称（site）を指定します。このオプション省略時は、あらかじめ設定されたデフォルトのサイト名称です。

-o file：オペレーション結果出力先のファイル名称（file）を指定します。

### <終了コード>

次の終了コードを返します。

0：正常終了

0以外：異常終了

## &lt;オペレーション&gt;

引数でアドレス指定した場合

#svadm addr[Enter]

情報表示

#

引数でアドレス指定しなかった場合

#svadm[Enter]

++ address information display start --&gt; site(XXXXX) ++

\*addr : addr[Enter]

情報表示

\*addr : q[Enter]

++ address information display end ++

#

## &lt;説明&gt;

下線 (   ) の箇所は、ユーザの入力部分です。

XXXXX : サイト名称

addr : 情報を得たいアドレスを指定します。

q : コマンドを終了します。

[Enter] : [Enter] キーを押します。

## &lt;表示フォーマット&gt;

表示内容は以下の4種類です。

[1] name = XXXXXXXXX type = XXXXXXXXXXXXX raddr = XXXXXXXXX

[2] lspace = XXXXXXXXX external name is not defined

[3] address error

## &lt;説明&gt;

XXXがデータです。

[1]は、指定アドレスの論理空間にリソースが登録されている場合の表示フォーマットです。

[2]は、指定アドレスにリソースが登録されていない場合の表示フォーマットです。

[3]は、下記の場合に表示します。

- ・引数説明で明記したアドレスの範囲以外を指定した場合
- ・引数説明で明記したアドレスの範囲内であるがGAREAの範囲を超えている場合

## 第5章 マップ

**name** : 外部名称 (プログラム名称、サブプログラム名称、細分割領域名称)

**type** : 外部名称の属性

**task** : プログラム名称

**sub** : サブプログラム名称

**data** : グローバル

**raddr** : 外部名称の領域の先頭アドレスからの相対アドレス。外部名称により以下の相対アドレスを表示します。

**task** : プログラムのtext部の先頭論理アドレスからの相対アドレス

**sub** : サブプログラムのtext部の先頭論理アドレスからの相対アドレス

**data** : グローバルの細分割領域名称の先頭論理アドレスからの相対アドレス

**lspace** : 指定アドレスで示した領域の論理空間名称 (ただし、外部名称が定義されていない場合)

<名前>

svsitectl

<形式>

svsitectl [オプション]

<機能説明>

svsitectlは、指定サイトの状態を制御し、表示します。

<オプション>

-query : 開発系に登録されているサイトの一覧を表示します。

-loadall sitename :

指定サイトに登録されているリソースをすべてダウンロード済みの状態にします。

<終了コード>

次の終了コードを返します。

0 : 正常終了

0以外 : 異常終了

## 第6章 立ち上げ

### <名前>

svrpl — リモートローディング

### <形式>

svrpl [{-u site}{-s}] [-r] [{-time|-notime}]

### <機能説明>

svrplは、指定サイトをストップさせ、バックアップファイルの内容をPCs側の主メモリに転送し、指定サイトを立ち上げます。オプションには以下のものがあります。

### <オプション>

- u site : ローディング対象となるサイト名称を指定します。このオプション省略時は、環境変数“RSSITE”に設定されたサイトに対して処理します。
- r : ローディング終了後、指定サイトをスタートしません。
- s : 指定サイトをストップするかの確認応答を取らずにストップします。  
-uオプションと同時に指定してください。  
また、-sオプションを指定しなかった場合はオペレータ操作とみなし、ダウンロード時のストップ確認を会話形式により行います。
- time : 指定サイトの時刻を設定します。
- notime : 指定サイトの時刻を設定しません。  
時刻を設定したくない場合には、このオプションを設定してください。

### <使用上の留意点>

- ・ローディング中にエラーが起きた場合、指定サイトをストップさせたまま終了します。
- ・ダウンロード後、スタートさせない (-rオプション指定) ときは、-timeオプション指定があっても無視します。時刻設定を行いません。
- ・ダウンロード時、-timeオプションを指定しなかった場合は時刻を設定しません。時刻は、状態制御 (svcpuctl) コマンドでRUN要求をするとき、-timeオプションを指定することにより設定できます。

### <終了コード>

次の終了コードを返します。

- 0 : 正常終了
- 1 : 異常終了
- 2 : 通信異常
- 3 : シグナル受信

## &lt;名前&gt;

svcpuctl — リモート状態制御

## &lt;形式&gt;

svcpuctl [{-u site} {-s{-stop | -run}}] [-time] (状態制御)

svcpuctl [-u site] -ss (状態表示)

## &lt;機能説明&gt;

svcpuctlは、指定サイトの状態を制御します。また、指定サイトの状態を表示します。オプションには以下のものがあります。

## &lt;状態制御オプション&gt;

**-u site** : 処理対象となるサイト名称を指定します。このオプション省略時は、環境変数“RSSITE”に設定されたサイトに対して処理します。

**-s** : 本当にコマンドを実行してよいかの確認応答を取りません。

**-u**オプションと同時に指定してください。

**-s**オプションを指定しなかった場合は、オペレータ操作とみなし、状態 (stop/run) 指定を会話形式により指定します。

**-stop** : 指定サイトをSTOP状態にします。

**-run** : 指定サイトをRUN状態にします。

**-stop**、**-run**オプションは、**-s**オプションと同時に指定してください。

また、**-stop**、**-run**は同時に指定できません。

**-time** : RUN要求時、指定サイトの時刻設定をします。

svcpuctlは、最初に指定サイトの状態を表示します。次に制御したい状態 (runまたはstop) を入力することにより、指定サイトの状態を変更させてもよいかを確認します。ここで、yesを入力すると指定サイトの状態を変更します。noを入力すると何もせずに終了します。

## &lt;状態表示オプション&gt;

**-ss** : 指定サイトの状態 (RUNまたはSTOP) を表示します。

## &lt;使用上の留意点&gt;

状態制御オプションと状態表示オプションは同時に指定できません。

## &lt;終了コード&gt;

次の終了コードを返します。

0 : 正常終了

1 : 異常終了

2 : 通信異常

3 : シグナル受信

## 第7章 svdebug (オンラインデバッガ) とデバッグ支援コマンド

### <名前>

svdebug - オンラインデバッガ

### <形式>

svdebug [オプション]

### <機能説明>

svdebugは、オンラインデバッグ機能を提供します。

svdebugコマンドは、起動・終了時に下記メッセージを出力し、デバッガ起動後、プロンプト“サイト名称”が表示されると、各サブコマンドが受け付けられます。

svdebug起動時にブレークポイントおよびADTが設定されていた場合、それぞれの設定内容を表示します。

### <デバッガ起動時のメッセージ>

```
++ debugger start ++
```

```
address detect trap
```

```
addr = 0x*****-0x***** mode = *****
```

```
break point
```

```
name = プログラム名称 raddr = プログラム内相対アドレス object = 機械語命令パターン
```

```
·  
·
```

```
name = プログラム名称 raddr = プログラム内相対アドレス object = 機械語命令パターン
```

```
サイト名称>
```

### <デバッガ終了時のメッセージ>

```
++ debugger end ++
```

(注) -sオプションを指定して起動した場合、上記のメッセージは出力されません。

### <オプション>

-i fname : キー入力結果をファイルに出力するときのファイル名を指示します。

-o fname : 指定されたファイルにオペレーションの日付および結果を出力します。

-r fname : 指定されたコマンドファイル内のサブコマンド行を実行します。すべてのサブコマンド行を実行した後、svdebugコマンドは自動的に終了します。

コマンドファイルは、-iオプションで作成したファイルを利用できます。

-s sub command :

このオプションに指定したサブコマンドをダイレクト実行します。サブコマンド実行後、svdebugコマンドは自動的に終了します。

-u site : デバッガの処理対象となるサイト名を指定します。

-debug : デバッグモードを指定します。拡張サブコマンドが使用できます。

## &lt;終了コード&gt;

svdebugコマンドは、正常終了した場合リターン値として0を返し、異常終了した場合リターン値として1を返します。ただし、-sオプションで指定したサブコマンドがエラーになった場合は、リターン値として255を返します。

## &lt;注意事項&gt;

- 複数オプション指定時には、-sオプション以降のオプション指定はサブコマンドとみなされ、無視されます。

(例)

svdebug -i fname -s サブコマンド : -iはオプションとして判断されます。

svdebug -s サブコマンド -i fname : -iは-sのサブコマンドの一部としてみなされます。

- svdebugコマンドに指定したオプションが、<機能説明>の項で説明されているオプションのいずれにも該当しない場合は、以下の使用例が出力されます。

```
Usage: svdebug [options]
```

```
Options:
```

```

-i fname          specify a "key-input result file"
-o fname          specify a "operation result file"
-r fname          specify a "command file"
-s sub command   "sub command" direct run
-u site           specify a "site name"
```

- -rオプションで使用するコマンドファイルを、-iオプション指定時のキー入力結果ファイルを使用せず作成する場合は、以下の点に注意してください。
  - ・サブコマンド仕様に準じないデータが設定されている場合でも、チェックが行われずにその行を実行します。
  - ・対話型インタフェースを持つサブコマンドについては、オペレーション手順ごとにプロンプト番号に対応するデータを行単位に指定します。
  - ・コマンドファイル内に現れた空行は、サブコマンド行が現れなかったものとしてその行を無視します。
- -rオプション、-sオプションを使用して、ブレイクポイントのサブコマンド (br、rb、rr、rd、go) は使用できません。

● サブコマンド一覧

svdebugの機能を表2-6に示します。

表2-6 svdebug機能一覧

分類	サブコマンド	機能
タスク起動/停止	qu	タスクの起動要求
	ab	タスクの起動禁止
	re	タスクの起動禁止解除
	ta	タスクの状態表示
	su	タスクの実行抑止
	rs	タスクの実行抑止解除
	tm	タスクの周期起動
	ct	タスクの周期起動解除
	sht	タスクの周期起動表示
	si	スタック初期化
	sp	スタック使用量の表示
メモリプリント/ パッチ	md	アドレス指定によるメモリ内容の表示/変更
	sd	名称指定によるメモリ内容の表示/変更
	bs	指定ビットへのデータ設定
	bg	指定ビットのデータ表示
	mcp	メモリ内容のコピー
	mmv	メモリ内容の移動
	mf	メモリへのパターン値設定
ブレークポイント	br	ブレークポイントの設定/表示
	rb	ブレークポイントの解除
	rd	レジスタの表示
	rr	レジスタの内容変更
	go	ブレークポイントからの実行再開
システムエラー表示	el	エラーログの表示
	ss	システムの状態表示
現在時刻設定/表示	st	現在時刻の設定
	gt	現在時刻の表示
アップ/ダウンロー ディング、コンペア	ld	リソースの個別ダウンロード
	sv	リソースの個別バックアップ
	cm	バックアップファイルとPCsメモリの内容比較
DHP記録許可/禁止	dr	DHP記録許可
	ds	DHP記録禁止
ADT	as	ADTの設定/表示
	ac	ADTの解除
その他	svdhp	DHPの表示
	svadm	アドレスに対するリソース名称の表示
	ps	デバッグ文の表示開始
	pe	デバッグ文の表示終了
	ver	CMUのバージョン表示
	help	サブコマンド一覧表示
	q	デバッガの終了
	!	svdebug実行時の開発系マシン上のコマンドの実行

## ● シグナル受信時の動作

シグナル受信時の動作を表2-7に示します。

表2-7 シグナル受信時の動作

シグナル	動作
SIGQUIT SIGINT	実行中のサブコマンドを終了しサブコマンド入力待ちに戻る
SIGHUP SIGTERM	svdebugコマンド処理を終了する
その他	SIG_DFL

<名前>

qu - タスクの起動要求

<形式>

qu tn[,fact]

qu tname[,fact]

<機能説明>

quは、指定されたタスクを起動します。指定するパラメータは以下のとおりです。

tn : タスク番号 (1~最大タスク番号)

fact : 起動要因 (1~32)

tname : タスク名

<結果>

OK(0) : 正常終了

NG(≠0) : マクロエラー

マクロエラーのときは、マクロのリターンコードが表示されます。

<注意事項>

- factを省略した場合、fact=0が設定されたものとして扱われます。
- このサブコマンド起動時に何もパラメータを指定しなかった場合、または正しくない引数を指定した場合、下記メッセージを出力し“:”の後でパラメータの入力待ちとなります。  
また、この状態で [e] または [Enter] キーを押した場合にはサブコマンド処理を終了します。

```
input tn[,fact] or tname[,fact]
```

```
:
```

## &lt;名前&gt;

ab - タスクの起動禁止

## &lt;形式&gt;

ab tn1[-tn2]

ab tname

## &lt;機能説明&gt;

abは、指定されたタスクを起動禁止状態にします。指定するパラメータは以下のとおりです。

tn1 : 先頭タスク番号 (1~最大タスク番号)

tn2 : 最終タスク番号 (1~最大タスク番号)

tname : タスク名

## &lt;結果&gt;

OK(0) : 正常終了

NG(≠0) : マクロエラー

ただし、tn1-tn2指定時はすべて正常終了となります。

マクロエラーのときは、マクロのリターンコードが表示されます。

## &lt;注意事項&gt;

このサブコマンド起動時に何もパラメータを指定しなかった場合、または正しくない引数を指定した場合、下記メッセージを出力し “:” の後でパラメータの入力待ちとなります。

また、この状態で [e] または [Enter] キーを押した場合にはサブコマンド処理を終了します。

```
input tn1[-tn2] or tname
```

```
:
```

<名前>

re - タスクの起動禁止解除

<形式>

re tn1[-tn2]

re tname

<機能説明>

reは、指定されたタスクの起動禁止状態を解除します。指定するパラメータは以下のとおりです。

tn1 : 先頭タスク番号 (1~最大タスク番号)

tn2 : 最終タスク番号 (1~最大タスク番号)

tname : タスク名

<結果>

OK(0) : 正常終了

NG(≠0) : マクロエラー

ただし、tn1-tn2指定時はすべて正常終了となります。

マクロエラーのときは、マクロのリターンコードが表示されます。

<注意事項>

このサブコマンド起動時に何もパラメータを指定しなかった場合、または正しくない引数を指定した場合、下記メッセージを出力し “:” の後でパラメータの入力待ちとなります。

また、この状態で [e] または [Enter] キーを押した場合にはサブコマンド処理を終了します。

```
input tn1[-tn2] or tname
```

```
:
```

<名前>

ta - タスクの状態表示

<形式>

ta tn1[-tn2 [-s|-r]]

ta tname

<機能説明>

taは、指定されたタスクの状態を表示します。指定されたタスクが実行抑止状態の場合、レジスタの内容も表示します。指定するパラメータおよびオプションは以下のとおりです。

tn1 : 先頭タスク番号 (1~最大タスク番号)

tn2 : 最終タスク番号 (1~最大タスク番号)

tname : タスク名

-s : タスク番号、タスク名、タスクステータスだけ表示されます。

-r : NON EXISTENT、DORMANT、IDLE以外のタスクの状態が表示されます。

taサブコマンドの表示フォーマットは以下のとおりです。

```
tn=*** (0x**)  tname=*****          task state=*****          (0x*****)
tcb top=0x*****  fact=0x*****  level=**  (**)
task top=0x*****  stack=0x*****-0x*****
(レジスタ情報)
```

● 表示情報の説明

tn : タスク番号 (10進および16進で表示)

tname : タスク名

task state : タスク状態 (ステータスビットを16進表示)

fact : タスクの起動要因

level : カレントタスクレベル (カッコ内はタスクレベル初期値)

tcb top : TCBの先頭アドレス

task top : タスク先頭アドレス

stack : タスクスタック範囲

レジスタ情報 : タスクが実行抑止状態のとき、レジスタ情報も表示されます。

レジスタ情報の出力フォーマットは以下のとおりです。

```

IAR =0x***** MSR =0x***** CR =0x***** LR =0x*****
CTR =0x***** XER =0x*****
GR0 =0x***** GR1 =0x***** GR2 =0x***** GR3 =0x*****
GR4 =0x***** GR5 =0x***** GR6 =0x***** GR7 =0x*****
GR8 =0x***** GR9 =0x***** GR10=0x***** GR11=0x*****
GR12=0x***** GR13=0x***** GR14=0x***** GR15=0x*****
GR16=0x***** GR17=0x***** GR18=0x***** GR19=0x*****
GR20=0x***** GR21=0x***** GR22=0x***** GR23=0x*****
GR24=0x***** GR25=0x***** GR26=0x***** GR27=0x*****
GR28=0x***** GR29=0x***** GR30=0x***** GR31=0x*****
    
```

タスク状態として表示する情報は、表 2-8 に示すいずれかとなります。

表 2-8 タスクの状態

状態	意味
NON EXISTENT	未登録
DORMANT	起動抑止状態
IDLE	起動待ち状態
READY	実行中または実行待ち状態
WAIT	イベント待ち状態
SUSPENDED	実行抑止状態

ステータスビットの構成とその意味は、表 2-9 のとおりです。

表 2-9 ステータスビットの構成

ステータスビット値	意味
0x1	多重起動あり
0x10	DELAYによる実行抑止中
0x20	SUSPによる実行抑止中
0x40	RSERV、PRSRVの資源解放待ち
0x1000	EXIT処理実行中
0x2000	RLEAS処理ペンディング中
0x4000	ABORT処理実行中
0x8000	QUEUE処理ペンディング中

(注) ステータスビットは、複数のビットが同時にONになる場合があります。

## &lt;注意事項&gt;

- このサブコマンド起動時に何もパラメータを指定しなかった場合、または正しくない引数を指定した場合、下記メッセージを出力し “:” の後でパラメータの入力待ちとなります。  
また、この状態で [e] または [Enter] キーを押した場合にはサブコマンド処理を終了します。

```
input tn1[-tn2 [-s|-r]] or tname
:
```

- -suspオプションを指定した場合、タスクの範囲指定は行えません。
- -suspオプションを指定時にタスク状態が、DORMANT状態、またはNON EXIST状態である場合、エラーメッセージを出力し、サブコマンドを終了します。
- -suspオプションを指定した場合、レジスタ情報を取得するために、タスクの状態がSUSPENDED状態でなくても、一時的にSUSPENDED状態にします。
- -suspオプションは、オンラインでは使用しないでください。

<名前>

su - タスクの実行抑止

<形式>

su tn1[-tn2]

su tname

<機能説明>

suは、指定されたタスクの実行を抑止します。指定するパラメータは以下のとおりです。

tn1 : 先頭タスク番号 (1~最大タスク番号)

tn2 : 最終タスク番号 (1~最大タスク番号)

tname : タスク名

<結果>

OK(0) : 正常終了

NG(≠0) : マクロエラー

ただし、tn1-tn2指定時はすべて正常終了となります。

マクロエラーのときは、マクロのリターンコードが表示されます。

<注意事項>

このサブコマンド起動時に何もパラメータを指定しなかった場合、または正しくない引数を指定した場合、下記メッセージを出力し“:”の後でパラメータの入力待ちとなります。

また、この状態で [e] または [Enter] キーを押した場合にはサブコマンド処理を終了します。

```
input tn1[-tn2] or tname
```

```
:
```

-suspオプションは、オンラインでは使用しないでください。

## &lt;名前&gt;

rs - タスクの実行抑止解除

## &lt;形式&gt;

rsは、指定されたタスクの実行抑止状態を解除します。指定するパラメータは以下のとおりです。

rs tn1[-tn2]

rs tname

## &lt;機能説明&gt;

tn1 : 先頭タスク番号 (1~最大タスク番号)

tn2 : 最終タスク番号 (1~最大タスク番号)

tname : タスク名

## &lt;結果&gt;

OK(0) : 正常終了

NG(≠0) : マクロエラー

ただし、tn1-tn2指定時はすべて正常終了となります。

マクロエラーのとき、マクロのリターンコードが表示されます。

## &lt;注意事項&gt;

このサブコマンド起動時に何もパラメータを指定しなかった場合、または正しくない引数を指定した場合、下記メッセージを出力し“:”の後でパラメータの入力待ちとなります。

また、この状態で [e] または [Enter] キーを押した場合にはサブコマンド処理を終了します。

```
input tn1[-tn2] or tname
```

```
:
```

<名前>

tm — タスクの周期起動

<形式>

tm

id:

tn[,fact]: (または tname[,fact])

t,cyct:

または

tm

id:

tn[,fact]: (または tname[,fact])

t:

<機能説明>

tmは、指定されたタスクに対して周期起動をかけます。tmが起動するとサブプロンプトが表示されます。指定するパラメータは以下のとおりです。

id : 起動するタイマタスクの種類 (1~4)

tn : タスク番号 (1~最大タスク番号)

tname : タスク名

fact : 起動要因 (1~32)

t : 初回のタイマイベントの時刻または現在からの相対時間  
相対時間はミリ秒で指定します。

idとして1または3を指定したとき、相対時間として1~86400000が指定できます。

idとして2または4を指定したとき、相対時間として0~86399999が指定できます。

cyct : 周期的にイベントを発生させる場合の周期時間

周期時間はミリ秒で指定します。周期時間として1~86400000が指定できます。

id、t、cyctの詳細は表 2-10を参照してください。

<結果>

OK(0) : 正常終了

NG(≠0) : マクロエラー

マクロエラーのときは、マクロのリターンコードが表示されます。

<注意事項>

- factを省略した場合、fact=0が設定されたものとして扱います。
- パラメータの入力待ちの状態、[e] または [Enter] キーを押した場合、サブコマンドを終了します。

表 2-10 id、t、cyctの説明

タイマイイベント	id	t	cyct	説明
時間指定	1	現時刻から起動までの 相対時間		パラメータtで指定された時間経過 後、パラメータtn、tnameで指定さ れたタスクを起動します。
時刻指定	2	午前0時を起点にした 起動時刻		パラメータtで指定された時刻に、 パラメータtn、tnameで指定された タスクを起動します。
時間周期指定	3	現時刻から起動までの 相対時間 (初回の起動までの 相対時間)	初回の起動の後、周期 的に起動する周期時間	パラメータtで指定された時間経過 後、パラメータtn、tnameで指定さ れたタスクを起動します。 その後、パラメータcyctで指定され た周期で、パラメータtn、tnameで 指定されたタスクを起動します。
時刻周期指定	4	午前0時を起点にした 起動時刻 (初回の起動時刻)	初回の起動の後、周期 的に起動する周期時間	パラメータtで指定された時刻に、 パラメータtn、tnameで指定された タスクを起動します。 その後、パラメータcyctで指定され た周期で、パラメータtn、tnameで 指定されたタスクを起動します。

<名前>

ct - タスクの周期起動解除

<形式>

ct tn[,fact]

ct tname[,fact]

<機能説明>

ctは、指定されたタスクの周期起動を解除します。指定するパラメータは以下のとおりです。

tn : タスク番号 (1~最大タスク番号)

fact : 起動要因 (1~32)

tname : タスク名

<結果>

OK(0) : 正常終了

NG(≠0) : マクロエラー

マクロエラーのときは、マクロのリターンコードが表示されます。

<注意事項>

- factを省略した場合、fact=0が設定されたものとして扱います。
- このサブコマンド起動時に何もパラメータを指定しなかった場合、あるいは正しくない引数を指定した場合、下記メッセージを出力し“:”の後でパラメータの入力待ちとなります。  
また、この状態で [e] または [Enter] キーを押した場合にはサブコマンド処理を終了します。

```
input tn[,fact] or tname[,fact]
```

```
:
```

## &lt;名前&gt;

sht — タスクの周期起動表示

## &lt;形式&gt;

sht

## &lt;機能説明&gt;

shtは、現在設定されているタスクの周期起動をすべて表示します。

## &lt;結果&gt;

以下の形式でタスク周期起動を表示します。

ID	TN	FACT	TIME	CYT
*	***	**	****/**/** **:*:*:**.***	*****
			.	
			.	
			.	

ID : タイマ種別

(1 : 時間起動 (timer) 2 : 時刻起動 (timer) 3 : 時間指定周期起動 (timer)

4 : 時刻指定周期起動 (timer) 10 : 時刻起動 (wake) 11 : 時刻指定周期起動 (wake) )

TN : タスク番号

FACT : 起動要因

TIME : 起動時刻 (年/月/日 時:分:秒.ミリ秒)

CYT : 周期時間 (ミリ秒)

## &lt;注意事項&gt;

shtサブコマンドは、同一サイトに対して同時に複数の起動はできません。

<名前>

md - アドレス指定によるメモリ内容の表示/変更

<形式>

md

```
1 storage(s,m,*) : {s}
                   {m}
                   {*}
                   {e}
                   {return}
2 addr : {addr1 [-addr2] [-h |-d |-f] [-l |-w |-b] [-all |-omit ]}
         {addr1 [-addr2] [-fd] [-all |-omit ]}
         {addr1 [.len] [-h |-d |-f] [-l |-w |-b] [-all |-omit ]}
         {addr1 [.len] [-fd] [-all |-omit ]}
         {*n}
         {e}
0xaaaaaaaa 0xdddddddd : {data}
                       {return}
                       {e}
```

(注) 下線 (   ) の箇所は、ユーザの入力部分を示します。

<機能説明>

mdは、論理アドレス指定によりメモリ内容を表示/変更します。起動するとサブプロンプトが表示されます。各々のサブプロンプトに対する入力は以下のようになります。

1 storage(s,m,\*)

- s : バックアップファイルを変更・表示の対象とします。
- m : 実機メモリを変更・表示の対象とします。
- return : 実機メモリを変更・表示の対象とします。
- \* : バックアップファイル、実機メモリを変更の対象とします。
- e : サブコマンドの終了を指示します。

(注) storage=\* 指定時のデータ表示は、バックアップファイルを対象とします。

## 2 addr

- addr1-addr2 : 表示先頭アドレスaddr1から表示最終アドレスaddr2までを表示することを指示します。
- addr1,len : 表示先頭アドレスaddr1からlenで指定したバイト数を表示することを指示します。
- h : データ出力形式を16進表示とします。
- d : データ出力形式を10進表示とします。
- f : データ出力形式を単精度浮動小数点表示とします。
- fd : データ出力形式を倍精度浮動小数点表示とします。
- l : データ長を4バイトとします。
- w : データ長を2バイトとします。
- b : データ長を1バイトとします。
- all : 同一データが行単位で連続しても省略表示をしないように指示します。
- omit : 同一データが行単位で連続した場合に省略表示をするように指示します。
- \*n : nに指定したプロンプト番号の前処理に戻ります (n=1だけ指定できます)。
- e : サブコマンドの終了を指示します。

## 0xaaaaaaaa 0xdddddddd

- data : 変更するデータの値を指定します。
- return : データの変更がないことを指示します。
- e : 2 addrの入力に戻ります。

## &lt;注意事項&gt;

- 2 addrの指定で、addr2およびlenのパラメータ指定がないときデータ変更 (パッチ) モードとなり、このパラメータ指定があるときはデータ表示 (プリント) モードとなります。  
また、データ表示終了後は、再度2 addrの入力待ちとなります。
- データ出力形式とデータ長の両オプションを省略した場合、このサブコマンド内で最後に指定したものが有効となります。デフォルトは16進4バイト (-h、-l) です。
- データ変更 (パッチ) 時に指定できる値は、データ出力形式とデータ長の両オプションの指定に依存します。表 2-11に指定できる値とオプションの対応を示します。

表 2-11 指定できる値とオプションの組み合わせ

fmt \ size	-l	-w	-b	指定なし
-h	○	○	○	—
-d	○	○	○	—
-f	◎	○	○	—
-fd	×	×	×	◎
指定なし	—	—	—	—

○ : 8進、10進、16進が指定できます。  
 ◎ : 実数が指定できます。  
 — : 前回指定したfmt、sizeに依存します。  
 × : 指定できないオプションの組み合わせです。

(fmt : データ出力形式 size : データ長)

- データ出力形式に-fd (倍精度浮動小数点形式) を指定し、データ表示・変更した後で、データ出力形式またはデータ長のオプションを省略してデータ表示・変更を行う場合、データ出力形式のオプション省略時は-h (16進形式)、データ長のオプション省略時は-l (4バイト) が指定されたものとして扱います。
- 1 storageの指定で's' を指定し、2 addrの指定でバックアップファイルの存在しないアドレスを指定した場合は、エラーメッセージを表示し、2 addrの入力待ちとなります。
- 1 storageの指定で's' を指定し、2 addrの指定でパッケージの共有リソースを含むアドレスを指定した場合は、エラーメッセージを表示し、2 addrの入力待ちとなります。
- データ表示 (プリント) の表示フォーマットは、データ出力形式とデータ長の両オプションの組み合わせにより、表 2-12 のようになります。

表 2-12 オプションの組み合わせによる表示フォーマット

fmt \ size	-l	-w	-b	指定なし
-h	h/4	h/2	h/1	—
-d	d/4	d/2	d/1	—
-f	f/4	h/2	h/1	—
-fd	×	×	×	f/8
指定なし	—	—	—	—

fmt/size : fmt : h (16進形式)  
 d (10進形式)  
 f (浮動小数点形式)  
 size : バイトサイズ  
 — : 前回指定したfmt、sizeに依存します。  
 × : 指定できないオプションの組み合わせです。

(fmt : データ出力形式 size : データ長)

- mdの表示 (READ) /変更 (WRITE) 時のメモリアクセス範囲は、下図の網掛けで示される範囲内になりますが、物理メモリにマッピングされていない空間はアクセスできません。

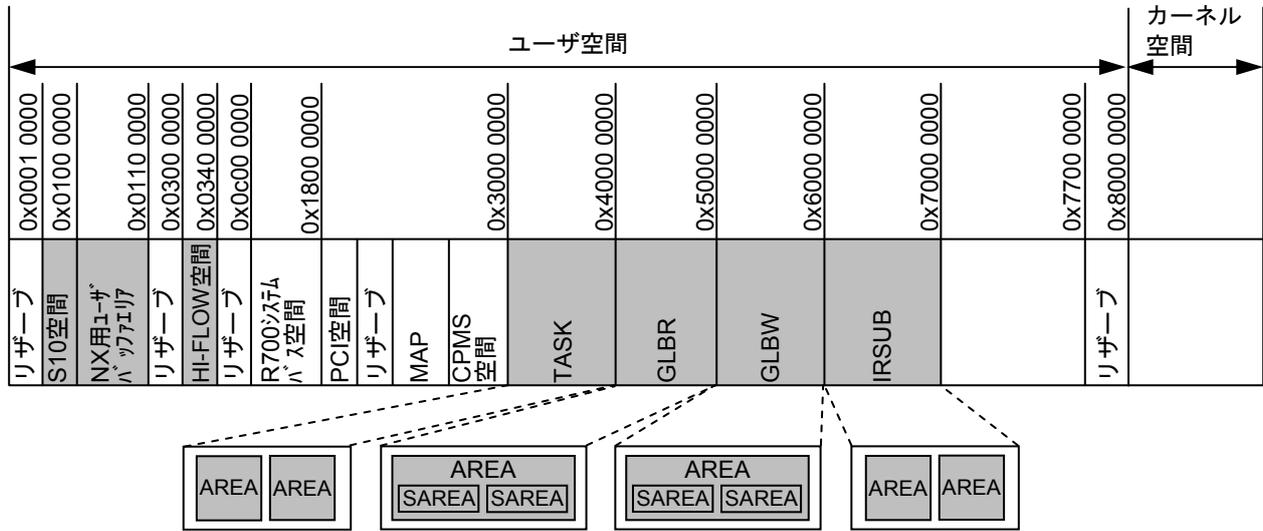


図2-4 メモリアクセス範囲

- mdでは、データ表示 (プリント) モードでのダイナミック表示機能をサポートしています。ダイナミック表示のオペレーションは、図2-5を参照してください。

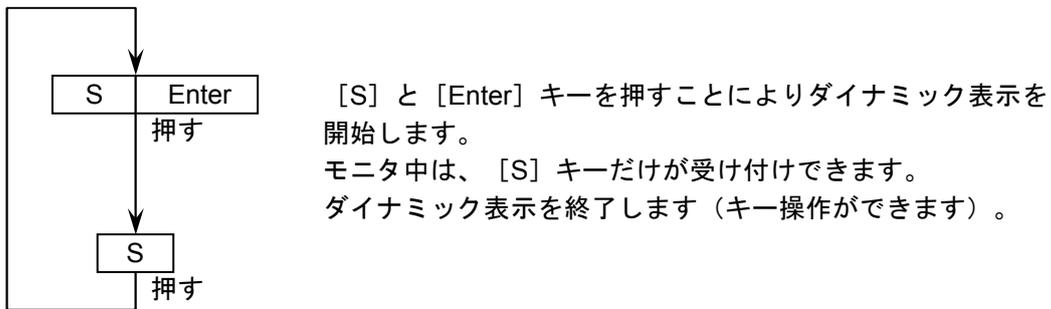


図2-5 ダイナミック表示のオペレーション

<名前>

sd - 名称指定 (タスク名、サブプログラム名、プログラム名、グローバル名) によるメモリ内容の表示/変更

<形式>

sd

1 name : {name [-t | -s | -g | -b | -w]}  
{e}

2 storage(s,m,\*): {s}  
{m}  
{\*}  
{\*n}  
{e}  
{return}

3 baddr : {addr [-h | -d | -f] [-l | -w | -b] [-all | -omit]}  
{addr [-fd] [-allv|-omit]}  
{\*n}  
{e}

4 raddr : {addr1 [-addr2 | -\*]}  
{addr1 [,len | -\*]}  
{\*}  
{\*n}  
{e}

0xaaaaaaaa(0xl|lll) 0xdddddd : {data}  
{return}  
{e}

(注) 下線 ( ) の箇所は、ユーザの入力部分を示します。

## &lt;機能説明&gt;

sdは、名称指定 (タスク名、サブプログラム名、プログラム名、グローバル名) によってメモリ内容を表示/変更します。起動するとサブプロンプトが表示されます。各々のサブプロンプトに対する入力は以下のとおりです。

## 1 name

- name : 表示・変更の対象となる名称を指定します。
- t : nameで指定した名称がプログラム名であることを指示します。  
相対アドレスをテキスト先頭からの相対として扱います。
- s : nameで指定した名称がサブプログラム名であることを指示します。
- g : nameで指定した名称がグローバルの細分割領域 (SAREA) 名であることを指示します。
- b : nameで指定したプログラム名称に対応するプログラムのBSS領域であることを指示します。  
相対アドレスを指定プログラムのBSS領域先頭からの相対として扱います。
- w : nameで指定したタスク名称に対応するタスクのスタック領域であることを指示します。  
相対アドレスを指定タスクのスタック領域先頭からの相対として扱います。
- e : サブコマンドの終了を指示します。

(注) -t、-s、-g、-b、-wオプションの指定を省略した場合、-gが指定されたものとして扱います。

## 2 storage(s,m,\*)

- s : バックアップファイルを変更・表示の対象とします。
- m : 実機メモリを変更・表示の対象とします。
- return : 実機メモリを変更・表示の対象とします。
- \* : バックアップファイル、実機メモリを変更の対象とします。
- \*n : nに指定したプロンプト番号の前処理に戻ります (n=1だけ指定できます)。
- e : サブコマンドの終了を指示します。

(注) storage=\* 指定時のデータ表示は、バックアップファイルが対象となります。

## 3 baddr

- addr : 変更・表示の対象領域先頭からの相対アドレスを指定します。
- \*n : nに指定したプロンプト番号の前処理に戻ります ( $1 \leq n \leq 2$ の範囲で指定できます)。
- h : データ出力形式を16進表示とします。
- d : データ出力形式を10進表示とします。
- f : データ出力形式を単精度浮動小数点表示とします。
- fd : データ出力形式を倍精度浮動小数点表示とします。
- l : データ長を4バイトとします。
- w : データ長を2バイトとします。
- b : データ長を1バイトとします。
- all : 同一データが行単位で連続しても省略表示をしないように指示します。
- omit : 同一データが行単位で連続した場合に省略表示をするように指示します。
- e : サブコマンドの終了を指示します。

#### 4 raddr

- addr1-addr2 : 表示先頭アドレスaddr1から表示最終アドレスaddr2までを表示することを指示します (アドレスはbaddrのaddrを起点として計算します)。
- addr1,len : 表示先頭アドレスaddr1からlenで指定したバイト数を表示することを指示します (アドレスはbaddrのaddrを起点として計算します)。
- addr1,\* : 表示先頭アドレスaddr1から指定シンボルの残りの領域をすべて表示することを指示します (アドレスはbaddrのaddrを起点として計算します)。
- addr1-\* : 表示先頭アドレスaddr1から指定シンボルの残りの領域をすべて表示することを指示します (アドレスはbaddrのaddrを起点として計算します)。
- \* : 指定シンボルの該当領域すべてを表示することを指示します。
- \*n : nに指定したプロンプト番号の前処理に戻ります ( $1 \leq n \leq 3$ の範囲で指定できます)。
- e : サブコマンドの終了を指示します。

#### 0xaaaaaaaa(0xlIlllll) 0xdddddddd

- data : 変更するデータの値を指定します。
- return : データの変更がないことを指示します。
- e : 4 raddrの入力に戻ります。

#### <注意事項>

- 4 raddrの入力でaddr1単独でパラメータ指定されたときデータ変更 (パッチ) モードとなり、これ以外のパラメータ指定のときはデータ表示 (プリント) モードとなります。また、データ表示終了後は、再度4 raddrの入力待ちとなります。
- データ出力形式・データ長の両オプションを省略した場合、このサブコマンド内で最後に指定したものを有効とします。デフォルトは16進4バイト (-h、-l) とします。
- sdで表示・変更できるエリアの領域範囲は、グローバルの場合、グローバルに割り付けた細分割領域 (SAREA) の範囲内で、プログラム・サブプログラムの場合、TEXT+DATA部、BSS部、STACK部の範囲内とします。
- データ変更 (パッチ) 時に指定できる値は、データ出力形式とデータ長の両オプションの指定に依存します。指定できる値とオプションは表2-11を参照してください。
- データ表示 (プリント) 時の表示フォーマットは、データ出力形式とデータ長の両オプションの組み合わせはmdサブコマンドと同様です。
- sdサブコマンドは、mdサブコマンド同様にデータ表示 (プリント) モードでのダイナミック表示機能をサポートします。ダイナミック表示のオペレーションは、mdサブコマンドを参照してください。
- 1 nameの指定でバックアップファイルの存在しない領域を指定し、2 storageの指定で 's' を指定した場合は、エラーメッセージを表示し、2 storageの入力待ちとなります。
- 1 nameの指定でパッケージの共有リソース領域を指定し、2 storageの指定で 's' を指定した場合は、エラーメッセージを表示し、2 storageの入力待ちとなります。

## &lt;名前&gt;

bs — 指定ビットへのデータ設定

## &lt;形式&gt;

bs

```
1 storage(s,m,*) : {s}
                    {m}
                    {*}
                    {e}
                    {return}

2 addr : {addr}
        {*n}
        {e}

3 bit : {bit1, len}
        {bit1-bit2}
        {*n}
        {e}

4 data : {data}
        {*n}
        {e}
```

(注) 下線 (   ) の箇所は、ユーザの入力部分を示します。

## &lt;機能説明&gt;

bsは、指定されたアドレスのビット位置へデータを設定します。起動するとサブプロンプトが表示されます。各々のサブプロンプトに対する入力は以下のようになります。

## 1 storage(s,m,\*)

- s : ビットセットの対象がバックアップファイルであることを指示します。
- m : ビットセットの対象が実機メモリであることを指示します。
- return : ビットセットの対象が実機メモリであることを指示します。
- \* : ビットセットの対象がバックアップファイル、実機メモリであることを指示します。
- e : サブコマンドの終了を指示します。

## 2 addr

- addr : セット対象メモリのアドレスを指示します。
- \*n : nに指定したプロンプト番号の前処理に戻ります (n=1だけ指定できます)。
- e : コマンドの終了を指示します。

### 3 bit

bit1, len : 先頭ビット番号bit1からビット長lenの範囲に設定することを指示します。

bit1-bit2 : 先頭ビット番号bit1から最終ビット番号bit2の範囲に設定することを指示します。

\*n : nに指定したプロンプト番号の前処理に戻ります ( $1 \leq n \leq 2$ の範囲で指定できます)。

e : コマンドの終了を指示します。

### 4 data

data : 設定するデータを指示します。

‘0x’または‘0X’で始めると16進数として処理し、それ以外は2進数として処理します。  
指示したビット数分のパターンを指定してください。

\*n : nに指定したプロンプト番号の前処理に戻ります ( $1 \leq n \leq 3$ の範囲で指定できます)。

e : コマンドの終了を指示します。

### <注意事項>

- メモリ内容を変更できる範囲はmdサブコマンドと同じです。
- 1 storageの指定で‘s’を指定し、2 addrの指定でバックアップファイルの存在しないアドレスを指定した場合は、エラーメッセージを表示し、2 addrの入力待ちとなります。
- 1 storageの指定で‘s’を指定し、2 addrの指定でパッケージの共有リソースを含むアドレスを指定した場合は、エラーメッセージを表示し、2 addrの入力待ちとなります。

## &lt;名前&gt;

bg — 指定ビットのデータ表示

## &lt;形式&gt;

bg

1 storage(s,m,\*) : {s}  
                   {m}  
                   {\*}  
                   {e}  
                   {return}

2 addr : {addr}  
           {\*n}  
           {e}

3 bit : {bit1, len}  
           {bit1-bit2}  
           {\*n}  
           {e}

(注) 下線 (    ) の箇所は、ユーザの入力部分を示します。

## &lt;機能説明&gt;

bgは、指定されたビット位置のデータを表示します。起動するとサブプロンプトが表示されます。各々のサブプロンプトに対する入力は以下ようになります。

## 1 storage(s,m,\*)

- s : ビット取り出し対象がバックアップファイルであることを指示します。
- m : ビット取り出し対象が実機メモリであることを指示します。
- return : ビット取り出し対象が実機メモリであることを指示します。
- \* : ビット取り出し対象がバックアップファイル、実機メモリであることを指示します。
- e : サブコマンドの終了を指示します。

## 2 addr

- addr : 取り出し対象メモリのアドレスを指示します。
- \*n : nに指定したプロンプト番号の前処理に戻ります (n=1だけで指定できます)。
- e : コマンドの終了を指示します。

## 3 bit

- bit1, len : 先頭ビット番号bit1からビット長lenの範囲を表示することを指示します。
- bit1-bit2 : 先頭ビット番号bit1から最終ビット番号bit2の範囲を表示することを指示します。
- \*n : nに指定したプロンプト番号の前処理に戻ります (1 ≤ n ≤ 2の範囲で指定できます)。
- e : コマンドの終了を指示します。

<結果>

メモリ内容は以下のフォーマットで出力されます。

```
  a d d r   0 1 2 3 4 5 6 7 8 9 a b c d e f  
0xNNNNNNNN c c c c c c c c c c c c c c c c
```

0xNNNNNNNN : アドレス

c : 1、0または\*。ビット位置が指定範囲外の場合は\*を表示します。

メモリ内容出力後、2 addrの入力に戻ります。

<注意事項>

- メモリ内容を参照できる範囲はmdサブコマンドと同じです。
- 1 storageの指定で 's' を指定し、2 addrの指定でバックアップファイルの存在しないアドレスを指定した場合は、エラーメッセージを表示し、2 addrの入力待ちとなります。

## &lt;名前&gt;

mcp — メモリ内容のコピー

## &lt;形式&gt;

mcp

```
1 storage(s,m,*) : {s}
                   {m}
                   {*}
                   {e}
                   {return}
2 s_addr : {addr1,len}
           {addr1-addr2}
           {*n}
           {e}
3 d_addr : {addr}
           {*n}
           {e}
```

(注) 下線 (   ) の箇所は、ユーザの入力部分を示します。

## &lt;機能説明&gt;

mcpは、メモリの内容やバックアップファイルの内容を指定アドレスにコピーします。

## 1 storage(s,m,\*)

- s : コピーの対象がバックアップファイルであることを指示します。
- m : コピーの対象が実機メモリであることを指示します。
- return : コピーの対象が実機メモリであることを指示します。
- \* : バックアップファイル、実機メモリをコピーの対象とします。
- e : サブコマンドの終了を指示します。

## 2 s\_addr

- addr1,len : コピー元先頭アドレスaddr1からlenで指定したバイト数をコピーすることを指示します。
- addr1-addr2 : コピー元先頭アドレスaddr1からコピー元最終アドレスaddr2までをコピーすることを指示します。
- \*n : nに指定したプロンプト番号の前処理に戻ります (n=1だけで指定できます)。
- e : サブコマンドの終了を指示します。

## 3 d\_addr

- addr : コピー元先頭アドレスを指示します。
- \*n : nに指定したプロンプト番号の前処理に戻ります (1 ≤ n ≤ 2の範囲で指定できます)。
- e : サブコマンドの終了を指示します。

<注意事項>

- mcpサブコマンドは、3 d\_addrのコピー先アドレス入力後、コピーを開始する前に、下記フォーマットで確認メッセージを表示します。

```

                コピー元アドレス範囲      コピー先アドレス範囲
copy : 0x*****-0x***** -> 0x*****-0x*****
memory data copy ok ? (y/n) :
    
```

↑  
storage種別に対応した  
メッセージとなります。

storage種別=s : backup file  
m : memory  
\* : memory and backup file

ここで、‘y’ または ‘Y’ が入力された場合、コピーを開始し、コピー開始・終了時に下記メッセージを表示します。これ以外の文字が入力された場合は、コピーを行わずに2 s\_addrの入力に戻ります。

```

コピー開始時   ***** memory copy start *****
コピー終了時   ***** memory copy end   *****
    
```

↑  
storage種別に対応したメッセージとなります。

- 奇数アドレスが指定された場合は、マイナス方向の偶数アドレスに補正します。
- メモリコピー終了後は、2 s\_addrの入力に戻ります。
- 1 storageの指定で ‘s’ または ‘\*’ を指定し、バックアップファイルの存在しないアドレスを指定した場合は、エラーメッセージを表示し、2 s\_addr入力待ちとなります。
- 1 storageの指定で ‘s’ または ‘\*’ を指定し、パッケージの共有リソースを含むアドレスを指定した場合は、エラーメッセージを表示し、2 s\_addrの入力待ちとなります。
- mcpのメモリアクセス範囲は、下図に網掛けで示す範囲内となります。  
mcpはmd、bs、bgと異なり、R700システムバス空間、S10空間はアクセス対象外となります。また、物理メモリにマッピングされていない空間はアクセスできません。

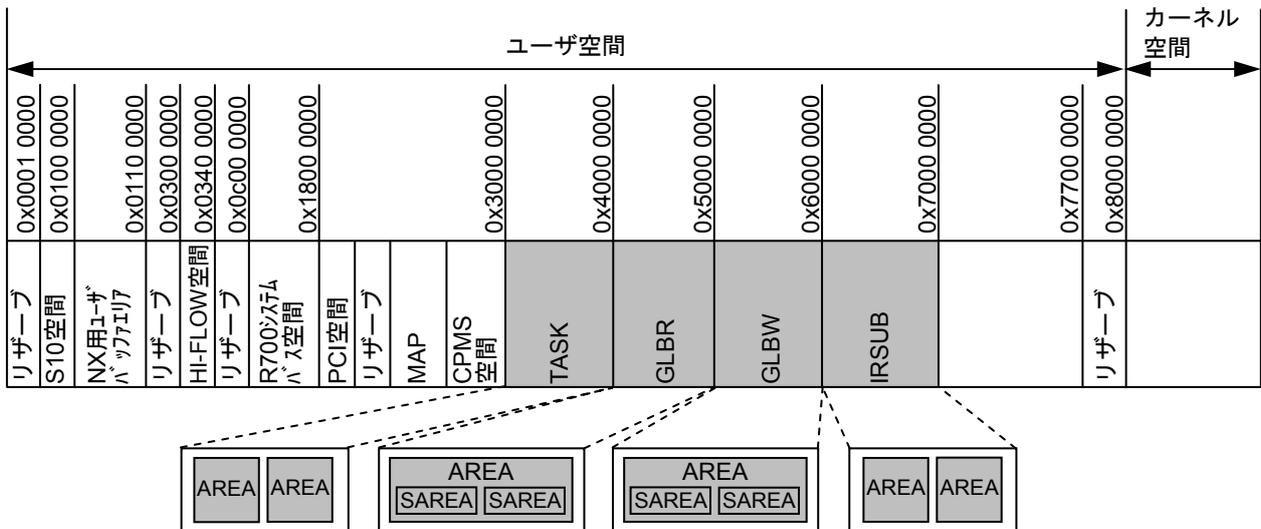


図 2-6 メモリアクセス範囲

## &lt;名前&gt;

mmv ー メモリ内容の移動

## &lt;形式&gt;

mmv

```
1 storage(s,m,*) : {s}
                   {m}
                   {*}
                   {e}
                   {return}
2 s_addr : {addr1,len}
           {addr1-addr2}
           {*n}
           {e}
3 d_addr : {addr}
           {*n}
           {e}
```

(注) 下線 (   ) の箇所は、ユーザの入力部分を示します。

## &lt;機能説明&gt;

mmvは、メモリ内容を移動します。移動元は0クリアします。

## 1 storage(s,m,\*)

- s : 移動の対象がバックアップファイルであることを指示します。
- m : 移動の対象が実機メモリであることを指示します。
- return : 移動の対象が実機メモリであることを指示します。
- \* : バックアップファイルと実機メモリの両方を移動の対象とします。
- e : サブコマンドの終了を指示します。

## 2 s\_addr

- addr1,len : 移動元先頭アドレスaddr1からlenで指定したバイト数を移動することを指示します。
- addr1-addr2 : 移動元先頭アドレスaddr1から移動元最終アドレスaddr2までを移動することを指示します。
- \*n : nに指定したプロンプト番号の前処理に戻ります (n=1だけ指定できます)。
- e : サブコマンドの終了を指示します。

## 3 d\_addr

- addr : 移動先先頭アドレスを指示します。
- \*n : nに指定したプロンプト番号の前処理に戻ります (1 ≤ n ≤ 2の範囲で指定できます)。
- e : サブコマンドの終了を指示します。

<注意事項>

- mmvサブコマンドは、3 d\_addrの移動先アドレス入力後、移動を開始する前に、下記フォーマットで確認メッセージを表示します。

```

移動元アドレス範囲      移動先アドレス範囲
move : 0x*****-0x***** -> 0x*****-0x*****
memory data move ok ? (y/n) :
    
```

↑  
storage種別に対応した  
メッセージとなります。

storage種別=s : backup file  
m : memory  
\* : memory and backup file

ここで、‘y’ または ‘Y’ が入力された場合、移動を開始し、移動開始・終了時に下記メッセージを表示します。これ以外の文字が入力された場合は、移動を行わずに2 s\_addrの入力に戻ります。

```

移動開始時   ***** memory move start *****
移動終了時   ***** memory move end   *****
    
```

↑  
storage種別に対応したメッセージとなります。

- 奇数アドレスが指定された場合は、マイナス方向の偶数アドレスに補正します。
- 指定できるアドレスの指定範囲は、mcpサブコマンドのメモリアクセス範囲と同じです。
- メモリ移動終了後は、2 s\_addrの入力に戻ります。
- 1 storageの指定で‘s’ または ‘\*’ を指定し、バックアップファイルの存在しないアドレスを指定した場合は、エラーメッセージを表示し、2 s\_addrの入力待ちとなります。
- 1 storageの指定で‘s’ または ‘\*’ を指定し、パッケージの共有リソースを含むアドレスを指定した場合は、エラーメッセージを表示し、2 s\_addrの入力待ちとなります。

## &lt;名前&gt;

mf — メモリへのパターン値設定

## &lt;形式&gt;

mf

1 storage(s,m,\*) : {s}

{m}

{\*}

{e}

{return}

2 addr : {addr1,len [-l/-w/-b]}

{addr1-addr2 [-l/-w/-b]}

{\*n}

{e}

3 data : {data}

{\*n}

{e}

(注) 下線 (   ) の箇所は、ユーザの入力部分を示します。

## &lt;機能説明&gt;

mfは、与えられたアドレス範囲に、一定のパターン値をセットします。

## 1 storage(s,m,\*)

s : セット対象メモリがバックアップファイルであることを指示します。

m : セット対象メモリが実機メモリであることを指示します。

return : セット対象メモリが実機メモリであることを指示します。

\* : セット対象メモリがバックアップファイルと実機メモリの両方であることを指示します。

e : サブコマンドの終了を指示します。

## 2 addr

addr1,len : セット先頭アドレスaddr1からlenで指定したケース数の範囲にセットすることを指示します。

addr1-addr2 : セット先頭アドレスaddr1からセット最終アドレスaddr2までの範囲にセットすることを指示します。

\*n : nに指定したプロンプト番号の前処理に戻ります (n=1だけ指定できます)。

e : サブコマンドの終了を指示します。

-l : データ長を4バイトとします。

-w : データ長を2バイトとします。

-b : データ長を1バイトとします。

## 3 data

data : セットするデータを指示します。

\*n : nに指定したプロンプト番号の前処理に戻ります (1 ≤ n ≤ 2の範囲で指定できます)。

e : サブコマンドの終了を指示します。

<注意事項>

- データ長を省略した場合は、このコマンド内で最後に指定したものを有効とします。  
デフォルトは-1 (4バイト) です。
- mfサブコマンドは、3 dataのパターンデータ設定後、セットを開始する前に、下記フォーマットで確認メッセージを表示します。

セット先アドレス範囲	パターンデータ (16進)
<pre>write : 0x*****-0x*****  pattern data = 0x***** memory data write ok ? (y/n) :</pre>	

↑  
storage種別に対応した  
メッセージとなります。

storage種別=s : backup file  
m : memory  
\* : memory and backup file

ここで、‘y’ または ‘Y’ が入力された場合、パターンデータのセットを開始します。これ以外の文字が入力された場合は、パターンデータのセットを行わずに2 addrの入力に戻ります。

- 指定アドレスがデータ長の境界ではない場合、マイナス方向の境界へ補正します。
- 指定できるアドレスの指定範囲は、mcpのメモリアクセス範囲と同じです。
- メモリセット終了後は、2 addrの入力に戻ります。
- 1 storageの指定で ‘s’ または ‘\*’ を指定し、2 addrの指定でバックアップファイルの存在しないアドレスを指定した場合は、エラーメッセージを表示し、2 addrの入力待ちとなります。
- 1 storageの指定で ‘s’ または ‘\*’ を指定し、2 addrの指定でパッケージの共有リソースを含むアドレスを指定した場合は、エラーメッセージを表示し、2 addrの入力待ちとなります。

<名前>

el - エラーログの表示

<形式>

el [-u site] [-f {s|m|l}] [-logno] [+count] [-o fname]

<機能説明>

elは、svelogコマンドを起動し、エラーログを表示します。

elサブコマンドの詳細は、svelogコマンドのコマンド仕様を参照してください。

<名前>

ss - システムの状態表示

<形式>

ss

<機能説明>

ssは、svcpuctlコマンドを起動し、システムの状態を表示します。

ssサブコマンドの詳細は、svcpuctlコマンドのコマンド仕様を参照してください。

<名前>

st - 現在時刻の設定

<形式>

st [yyyy.mm.dd.hh:mt:ss]

<機能説明>

stは、PCsの管理している現在時刻を設定します。指定するパラメータは以下のとおりです。

yyyy	年 (西暦4桁)
mm	月
dd	日
hh	時
mt	分
ss	秒

(注) 各時刻データは10進で入力します。

<結果>

OK(0) : 正常終了

<注意事項>

- サブコマンドのコマンド行の設定時刻 (yyyy.mm.dd.hh:mt:ss) を省略した場合は、下記メッセージを出力し入力待ちとなります。この状態で [e] または [Enter] キーを押した場合には、サブコマンドを終了します。

YYYY.MM.DD.HH:MT:SS :

- このサブコマンドが-sオプションによりダイレクト実行されたときは、エラーがあってもエラーメッセージは表示されません。

<名前>

gt - 現在時刻の表示

<形式>

gt

<機能説明>

gtは、PCsの管理している現在時刻を表示します。

<結果>

以下の形式で現在時刻を表示します。

yyyy.mm.dd.hh:mt:ss

yyyy 年（西暦4桁）

mm 月

dd 日

hh 時

mt 分

ss 秒

<名前>

br - ブレークポイントの設定/表示

<形式>

br [pname break1 ... break5 [-t|-s]]

<機能説明>

brは、ブレークポイントの設定および設定されているブレークポイントを表示します。

ブレークポイントを設定できるのは、タスクのプログラム (TEXT空間) および間接リンクサブプログラム (TEXT空間) だけです。

ブレークが発生したタスクは、WAIT状態となります。このとき、タスクの周期起動タイマは継続され、タイマ起動は実行されますが、ブレークポイントからの実行を再開しない限り、タスクはWAIT状態のままとなります。

ブレーク発生により停止しているタスクが存在する場合、その他のブレークポイントではブレークは発生しません。ブレーク発生により停止しているタスクが再開後、その他のブレークポイントが有効となります。

指定するパラメータは以下のとおりです。

pname : ブレークポイントを設定するプログラム名称を指定します。

break1~break5 : ブレークポイント (プログラム内相対アドレス) を指定します。

-t : 指定したプログラム名称がタスクのプログラム名称であることを示します。

-s : 指定したプログラム名称がサブプログラム名称であることを示します。

(注) -t、-sオプションを省略した場合は、タスク、サブプログラムの順に検索されます。

<結果>

ブレークポイントが正常に設定された場合は、下記メッセージを表示します。

break point set

name = プログラム名称 raddr = プログラム内相対アドレス object = 機械語命令パターン

pname、breakを指定しない場合は、現在設定されているブレークポイントを下記のように表示します。

break point

name = プログラム名称 raddr = プログラム内相対アドレス object = 機械語命令パターン

name = プログラム名称 raddr = プログラム内相対アドレス object = 機械語命令パターン \*BREAK\*

・  
・

↑  
現在ブレーク中の設定  
に表示されます。

他端末からブレークポイントを設定中の場合は、エラーメッセージを表示します。

## &lt;注意事項&gt;

- ブレークポイントは、サイトごとに最大5箇所まで設定できます。
- ブレークポイントに達すると、以下のようなメッセージが表示されます。

break!!

tn = タスク番号 name = プログラム名称 raddr = プログラム内相対アドレス

- rb、rd、rr、goなどのサブコマンドが失敗した場合は、brだけを発行してブレークポイントの状態を確認してください。これにより、開発系マシンのブレークポイント情報とコントローラの情報と異なっている場合に、開発系マシンの情報をコントローラに合わせることができます。
- コントローラの再立ち上げ後には、コントローラの停止前に設定されていたブレークポイント情報は保存されません。この場合、brだけを発行して開発系マシンのブレークポイント情報をコントローラの情報に合わせてください。
- ブレークポイントは、システム専用のタスクおよび組み込みサブルーチンには設定できません。
- このサブコマンドは、svdebug起動時に-debugオプションを指定した場合に使用できます。それ以外はエラーとなります。
- コントローラの再立ち上げやデバッガの異常終了により実行系と開発系のブレークポイントの設定が異なってしまった場合は、下記を実行してください。

brサブコマンドをパラメータなしで実行し、その後、ブレークポイントの設定が残っていた場合は、rbサブコマンドで解除してください。

<使用例>

以下にブレーク設定ポイントの設定手順を示します。

- 下記のprog1.c、prog2.cからprogというタスクプログラムを生成した場合のポイント①にブレークポイントを設定する場合を示します。

prog1.c

```
main()
{
    int    a, b, c;
    int    ret;

    a = 10;
    b = 20;

    c = add( a , b );
    ans_print( c );

    exit();
}

int add( int a , int b )
{
    int ans;

    ans = a + b; ← ①

    return(ans);
}
```

prog2.c

```
extern char  glb01_g[1024];      /* 1024byte */

void ans_print( int ans )
{
    int    ret;

    ret = rs_printf( &glb01_g[0] , "anser = %d¥n" , ans );

    return;
}
```

- ・ソースコンパイル時に、“-listfile -show=source,object” オプションを指定してコンパイルすることで、アセンブラソースにC言語のソースファイルを挿入することができます。prog1.cのソース内にブレークポイントを設定するため、prog1.lstを参照します。

ブレークポイントを設定するC言語のソースは (1) なので、その直後のアセンブラ命令 (2) がC言語のソースに対応しています。

該当する命令のソース内 (prog1.c) でのオフセットは、0x00000014となります。

prog1.lst

```

***** OBJECT LISTING *****

FILE NAME: prog1.c

SCT OFFSET   CODE       C LABEL      INSTRUCTION OPERAND    COMMENT
-----
P 00000000   prog1.c     1  main()
   00000000   _main:                                ; function: main
   00000000   4F22                                ; frame size=4
   00000000   STS.L      PR,@-R15
   prog1.c     2  {
   prog1.c     3  int    a, b, c;
   prog1.c     4  int    ret;
   prog1.c     5
   prog1.c     6  a = 10;
   prog1.c     7  b = 20;
   prog1.c     8
   prog1.c     9  c = add( a , b );
00000002   E40A      MOV      #10,R4    ; H' 0000000A
00000004   B006      BSR     _add
00000006   E514      MOV      #20,R5    ; H' 00000014
   prog1.c    10  ans_print( c );
00000008   D204      MOV.L   L12+2, R2   ; _ans_print
0000000A   420B      JSR     @R2
0000000C   6403      MOV     R0, R4
   prog1.c    11
   prog1.c    12  exit();
0000000E   D204      MOV.L   L12+6, R2   ; _exit
00000010   422B      JMP     @R2
00000012   4F26      LDS.L  @R15+, PR
   prog1.c    13  }
   prog1.c    14
   prog1.c    15  int add( int a , int b )
00000014   _add:                                ; function: add
   prog1.c    16  {
   prog1.c    17  int ans;
   prog1.c    18
   prog1.c    19  ans = a + b; (1)
   prog1.c    20
   prog1.c    21  return(ans);
00000014   345C      ADD     R5, R4     (2)
00000016   000B      RTS
00000018   6043      MOV     R4, R0
0000001A   L12:
0000001A   00000002 .RES.W  1
0000001C   <00000000> .DATA.L _ans_print
00000020   <00000000> .DATA.L _exit
   prog1.c    22  }

```

- svloadコマンドで実行プログラム生成時に、“-P” オプションを設定することで、マップファイルを生  
成します。

マップのSECTION=Pを確認すると、prog1.objは、0x3003c014から始まることが解ります。

ブレークポイントを設定するソース内相対アドレスは0x00000014なので、ブレークポイントを設定する  
アドレスは0x3003c028 (=0x3003c014 + 0x00000014) となります。

プログラム内相対アドレスは、ブレークポイント設定アドレスとプログラムの先頭アドレスから求めれ  
ばよいので、0x00000028 (=0x3003c028 - 0x3003c000) となります。

prog.map

SECTION=P					
FILE=C:\WINNT40\renix\tmp\str2545.obj					
	3003c000	3003c013	14		
_start					
	3003c000	0	none ,g		*
L237					
	3003c00c	0	none ,l		*
FILE=prog1.obj					
	3003c014	3003c037	24		
_main					
	3003c014	14	func ,g		*
_add					
	3003c028	10	func ,g		*
FILE=prog2.obj					
	3003c038	3003c04f	18		
_ans_print					
	3003c038	18	func ,g		*
FILE=cpms_exit					
	3003c050	3003c09f	50		
_exit					
	3003c098	0	none ,g		*
FILE=rpdp_rsprintf					
	3003c0a0	3003c113	74		
_rs_printf					
	3003c0a0	74	func ,g		*
FILE=rpdp_rssetmsg					
	3003c114	3003c16b	58		
_rssetmsg					
	3003c160	0	none ,g		*
FILE=sprintf					
	3003c16c	3003c19b	30		
_sprintf					
	3003c16c	30	func ,g		*

## &lt;名前&gt;

rb - ブレークポイントの解除

## &lt;形式&gt;

rb [pname break1 ... break5 [-t|-s]]

## &lt;機能説明&gt;

rbは、現在設定されているブレークポイントを解除します。パラメータを指定しない場合、現在設定されているブレークポイントをすべて解除します。指定するパラメータは以下のとおりです。

- pname : ブレークポイントを設定するプログラム名称を指定します。
- break1~break5 : ブレークポイント (プログラム内相対アドレス) を指定します。
- t : 指定したプログラム名称がタスクのプログラム名称であることを示します。
- s : 指定したプログラム名称がサブプログラム名称であることを示します。

(注) -t, sオプションを省略した場合は、タスク、サブプログラムの順に検索されます。

## &lt;結果&gt;

正常にブレークポイントを解除した場合には、以下のようなメッセージを表示します。

break point reset

name = プログラム名称 raddr = プログラム内相対アドレス object = 機械語命令パターン

他端末からブレークポイントを設定中の場合は、エラーメッセージを表示します。

## &lt;注意事項&gt;

- このサブコマンドは、svdebug起動時に-debugオプションを指定した場合に使用できます。それ以外はエラーとなります。
- ブレーク中のタスクが存在した場合、該当ブレークポイントの設定は解除できません。

<名前>

rd - レジスタの表示

<形式>

rd [-f|-h]

<機能説明>

rdは、ブレーク時の各レジスタの内容を表示します。パラメータを指定しない場合、浮動小数点レジスタは表示されません。指定するパラメータは以下のとおりです。

- f : 浮動小数点レジスタを実数で表示します。
- h : 浮動小数点レジスタを16進数で表示します。

<結果>

OK(0) : 正常終了

各レジスタの内容が下記のように表示されます。  
異常終了時は、エラーメッセージが表示されます。

```
SR =0x***** PC =0x***** GBR =0x***** PR =0x*****
MACH=0x***** MACL=0x***** FPUL=0x***** FPSCR=0x*****
R0 =0x***** R1 =0x***** R2 =0x***** R3 =0x*****
R4 =0x***** R5 =0x***** R6 =0x***** R7 =0x*****
R8 =0x***** R9 =0x***** R10 =0x***** R11 =0x*****
R12 =0x***** R13 =0x***** R14 =0x***** R15 =0x*****
```

-fオプションが指定された場合は、以下のように浮動小数点レジスタを実数で表示します。

```
FR0 =**, *****E** FR1 =**, *****E** FR2 =**, *****E** FR3 =**, *****E**
FR4 =**, *****E** FR5 =**, *****E** FR6 =**, *****E** FR7 =**, *****E**
FR8 =**, *****E** FR9 =**, *****E** FR10=**, *****E** FR11=**, *****E**
FR12=**, *****E** FR13=**, *****E** FR14=**, *****E** FR15=**, *****E**
XF0 =**, *****E** XF1 =**, *****E** XF2 =**, *****E** XF5 =**, *****E**
XF4 =**, *****E** XF5 =**, *****E** XF6 =**, *****E** XF7 =**, *****E**
XF8 =**, *****E** XF9 =**, *****E** XF10=**, *****E** XF11=**, *****E**
XF12=**, *****E** XF13=**, *****E** XF14=**, *****E** XF15=**, *****E**
DR0 =**, *****E** DR2 =**, *****E**
DR4 =**, *****E** DR6 =**, *****E**
DR8 =**, *****E** DR10=**, *****E**
DR12=**, *****E** DR14=**, *****E**
XD0 =**, *****E** XD2 =**, *****E**
XD4 =**, *****E** XD6 =**, *****E**
XD8 =**, *****E** XD10=**, *****E**
XD12=**, *****E** XD14=**, *****E**
```

-hオプションが指定された場合は、以下のように浮動小数点レジスタを16進数で表示します。

```

FR0 =0x***** FR1 =0x***** FR2 =0x***** FR3 =0x*****
FR4 =0x***** FR5 =0x***** FR6 =0x***** FR7 =0x*****
FR8 =0x***** FR9 =0x***** FR10=0x***** FR11=0x*****
FR12=0x***** FR13=0x***** FR14=0x***** FR15=0x*****
XF0 =0x***** XF1 =0x***** XF2 =0x***** XF3 =0x*****
XF4 =0x***** XF5 =0x***** XF6 =0x***** XF7 =0x*****
XF8 =0x***** XF9 =0x***** XF10=0x***** XF11=0x*****
XF12=0x***** XF13=0x***** XF14=0x***** XF15=0x*****
DR0 =0x***** 0x***** DR2 =0x***** 0x*****
DR4 =0x***** 0x***** DR6 =0x***** 0x*****
DR8 =0x***** 0x***** DR10=0x***** 0x*****
DR12=0x***** 0x***** DR14=0x***** 0x*****
XD0 =0x***** 0x***** XD2 =0x***** 0x*****
XD4 =0x***** 0x***** XD6 =0x***** 0x*****
XD8 =0x***** 0x***** XD10=0x***** 0x*****
XD12=0x***** 0x***** XD14=0x***** 0x*****

```

<レジスタ説明>

- SR : ステータスレジスタです。
- PC : プログラムカウンタです。
- GBR : グローバルベースレジスタです。ディスプレイメント付きGBR間接およびインデックス付きGBR間接のアドレッシング使用するベースアドレスを格納するレジスタです。
- PR : プロシジャレジスタです。プロシジャレジスタはサブルーチンの呼び出しに使われます。実行されていたプログラムがサブルーチン呼び出し関係の末尾である場合、このレジスタに戻りアドレスが格納されています。
- MACH : システムレジスタ (積和上位レジスタ) です。MAC命令 (積和演算) の加算値、およびMAC命令、MUL命令の結果を格納するために使用するレジスタです。演算結果が64ビット値だった場合、上位32ビットが格納されます。演算結果が32ビット値だった場合、32ビットが格納されます。
- MACL : システムレジスタ (積和下位レジスタ) です。演算結果が64ビット値だった場合、下位32ビットが格納されます。
- FPUL : 浮動小数点通信レジスタです。
- FPSCR : 浮動小数点ステータス/コントロールレジスタです。
- R0~R15  
汎用レジスタです (R15がスタックポインタとして使われます)。
- FR0~FR15  
単精度浮動小数点レジスタです。FPSCR.PR (31-0ビット値の19ビット目) =0の場合、FPRxx\_BANK0の値。FPSCR.PR=1の場合、FPRxx\_BANK1の値。
- XF0~XF15  
単精度浮動小数点拡張レジスタです。FPSCR.PR (31-0ビット値の19ビット目) =0の場合、FPRxx\_BANK1の値。FPSCR.PR=1の場合、FPRxx\_BANK0の値。
- DR0~DR15  
倍精度浮動小数点レジスタです。FPSCR.PR (31-0ビット値の19ビット目) =0の場合、FPRxx\_BANK0の値。FPSCR.PR=1の場合、FPRxx\_BANK1の値。
- XD0~XD15  
倍精度浮動小数点レジスタです。FPSCR.PR (31-0ビット値の19ビット目) =0の場合、FPRxx\_BANK1の値。FPSCR.PR=1の場合、FPRxx\_BANK0の値。

<注意事項>

- このサブコマンドは、svdebug起動時に-debugオプションを指定した場合に使用できます。それ以外はエラーとなります。
- 浮動小数点データが以下のような場合は、実数表示を指定された場合でも、16進数に変換して表示します。また、16進数表示後に対応する文字列を表示します。

浮動小数点データ	文字列	表示例	
		単精度	倍精度
非数	Na	FR0 =0x7fffffff:Na	DR0 =0xfff 00000 0x00000001:Na
無限大	In	FR0 =0x7f800000:In	DR0 =0xfff 00000 0x00000000:In
表現できる最大値	Ma	FR0 =0x7f7fffff:Ma	DR0 =0x7fefffff 0xffffffff:Ma
表現できる最小値	Mi	FR0 =0xff7fffff:Mi	DR0 =0xffefffff 0xffffffff:Mi

## &lt;名前&gt;

rr - レジスタの内容変更

## &lt;形式&gt;

rr

register : rx

data : datax

(注) 下線 (   ) の箇所は、ユーザの入力部分を示します。

## &lt;機能説明&gt;

rrは、ブレーク中のレジスタの内容を変更します。

rx : 内容を変更するレジスタの略称を指定します。

レジスタ略称は、rdサブコマンドで表示される名称を指定してください。

datax : 変更データを指定します (8進、10進、16進、実数の指定が可能)。

倍精度浮動小数点レジスタのデータ入力の実数でだけ可能です。

## &lt;結果&gt;

OK(0) : 正常終了

異常終了時は、エラーメッセージが表示されます。

## &lt;注意事項&gt;

- rrサブコマンドは、タスクがブレークポイントで中断しているときにだけ有効となります。
- このサブコマンドは、svdebug起動時に-debugオプションを指定した場合に使用できます。それ以外はエラーとなります。

<名前>

go - ブレークポイントからの実行再開

<形式>

go

<機能説明>

goは、ブレークポイントで中断していたアドレスから、プログラムを再開させます。プログラムの再開後、該当ブレークポイントの設定は解除されます。

<結果>

OK(0) : 正常終了

異常終了時は、エラーメッセージが表示されます。

<注意事項>

- goサブコマンドは、タスクがブレークポイントで中断しているときにだけ有効となります。
- このサブコマンドは、svdebug起動時に-debugオプションを指定した場合に使用できます。それ以外はエラーとなります。

## &lt;名前&gt;

as - ADTの設定/表示

## &lt;形式&gt;

as

1 addr : {addr[\_mask]}

{return}

{e}

2 mode(r,w,a) : {r}

{w}

{a}

{\*n}

{return}

{e}

(注) 下線 (   ) の箇所は、ユーザの入力部分を示します。

## &lt;機能説明&gt;

asは、ADT (アドレスディテクトトラップ (\*)) を設定します。起動するとサブプロンプトが表示されます。各々のサブプロンプトに対する入力は以下のとおりです。

## 1 addr

addr : トラップを設定する論理アドレスをバイト単位で指定します。

mask : 論理アドレスをマスクするマスク値を指定します。

10 : アドレスの下位10ビットをマスクします (範囲 : 1KB)。

12 : アドレスの下位12ビットをマスクします (範囲 : 4KB)。

16 : アドレスの下位16ビットをマスクします (範囲 : 64KB)。

20 : アドレスの下位20ビットをマスクします (範囲 : 1MB)。

e : サブコマンドの終了を指示します。

return : 現在設定されているADTの設定を以下のように表示します。

```
address detect trap
```

```
addr = 0x*****-0x***** mode = *****
```

## 2 mode(r,w,a)

r : リードトラップを指定します。

w : ライトトラップを指定します。

a : リード/ライトトラップを指定します。

return : リード/ライトトラップを指定します。

\*n : nに指定したプロンプト番号の前処理に戻ります (n=1だけ指定できます)。

e : サブコマンドの終了を指示します。

(\*) ADTとは、アドレスディテクトトラップの略語で、指定アドレスに対するアクセスが発生した場合に組み込みサブルーチンADTSにリンクし、エラーログを出力する機能です。

設定した論理アドレスに対するバイトアクセス、ワードアクセス、ロングワードアクセスがADTの対象となります。

ADTで設定した論理アドレスに指定したモードのアクセスが発生した場合、組み込みサブルーチンADTSにリンク後、エラーログが行われます。

<結果>

処理が正常に終了した場合、以下のようなメッセージを表示します。

```
address detect trap set
addr = 0x*****-0x*****   mode = *****
```

↑

ADTを設定したアドレスの範囲が表示されます。

↑

ADTに設定したモードが表示されます。

モード=read : リードトラップ  
 write : ライトトラップ  
 access : リード/ライトトラップ

<注意事項>

- ADTは、サイトごとに1箇所だけ設定できます。
- ADTは、CPUのリスタートでは解除されません。
- ADTで検出できるのは、プロセッサのMMUを経由したアクセスだけです。このため、RPL/RRB経由のアクセスや物理アドレスでアクセスするデバッガのメモリアクセスサブコマンドは、ADTでは検出できません。
- asサブコマンドで設定したADTは、svcpuctlコマンドに“-time”オプションを付けて実行すると解除されます。
- ADTが設定できる範囲は、下図の網掛けで示される範囲内になりますが、未マッピングエリアには設定できません。
- ADTSの組み込みサブルーチン内で、ADTでトラップを設定したアドレスにアクセスすると、システムダウンになりますので絶対に行わないでください。

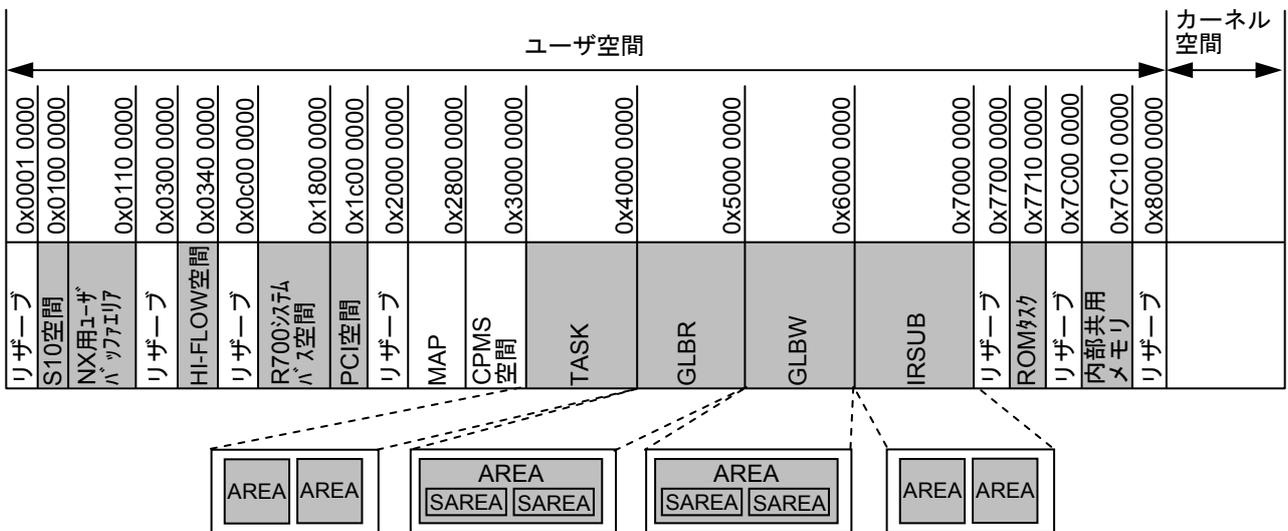


図 2 - 7 ADT設定範囲

<名前>

ac - ADTの解除

<形式>

ac

<機能説明>

acは、ADT (アドレスディテクトトラップ) の設定を解除します。

<結果>

正常にADTを解除した場合には、以下のようなメッセージを表示します。

```
address detect trap reset
addr = 0x*****-0x*****   mode = *****
```

<名前>

ld - リソースの個別ダウンロード

<形式>

ld {-t tname}

ld {-s sname}

ld {-g gname}

ld {-a aname}

ld {-m addr,len}

ld {-T [tno]}

ld {-U [point,ent]}

ld {-S [sno]}

ld {-G [gno]}

ld {-f fname}

<機能説明>

ldは、バックアップファイルの内容をPCsの主メモリに転送します。転送する対象は以下に示すオプションで指示します。

-t tname : tnameで指定されたプログラム本体をダウンロードし、PCs上でタスクを生成します。tnameがマルチタスクとして生成されている場合は、PCs上でもマルチタスクとして生成されません。

-s sname : snameで指定されたサブプログラムをローディングします。

サブプログラム本体のダウンロード、および間接リンクサブプログラムアドレステーブルのエントリまたは組み込みサブルーチン管理テーブルのエントリをダウンロードします。

snameで指定されたサブプログラムがマルチエントリIRSUBとして登録されている場合は、該当IRSUBのエントリポイントに対応するすべての間接リンクサブプログラムアドレステーブルのエントリをダウンロードします。

-g gname : gnameで指定されたグローバルをローディングします。

グローバルデータをダウンロードします。間接リンクグローバルである場合は、間接リンクグローバルアドレステーブルのエントリもダウンロードします。

-a aname : anameで指定された分割領域の内容をローディングします。

anameで指定する分割領域は、GLB領域に定義されている分割領域でなければなりません。

- m addr,len : 先頭アドレス (addr) 、バイト数 (len) 指定でローディングすることを指示します。  
指定するアドレスの範囲がアロケートされていない空間を含む場合は、ローディングしません。
- T [tno] : tnoで指定されたタスク番号のタスクをコントローラ上で削除または生成します。  
tnoが開発系マシン上で生成されているタスクである場合、タスクを生成します。  
tnoが開発系マシン上で生成されていないタスクである場合、タスクを削除します。  
tnoが省略された場合は、開発系マシン上の生成/削除操作がPCsに反映されていないタスクの一覧を表示します。
- U [point,ent] : point, entで指定される組み込みサブルーチン管理テーブルのエントリをローディングします。  
pointおよびentは以下のように指定します。  
point : 組み込みサブルーチンの組み込み箇所を表す文字列を指定します。  
ent : エントリ番号を表す1~4の数値です。  
point, entが省略された場合は、開発系マシン上での登録/削除操作がPCsに反映されていない組み込みサブルーチン管理テーブルの一覧を表示します。
- S [sno] : snoで指定された間接リンクサブプログラムアドレステーブルエントリをローディングします。  
snoが省略された場合は、開発系マシン上の登録/削除がPCsに反映されていない間接リンクサブプログラムアドレステーブルの一覧を表示します。
- G [gno] : gnoで指定された間接リンクグローバルアドレステーブルをローディングします。  
gnoが省略された場合は、開発系マシン上での登録/削除操作がPCsに反映されていない間接リンクグローバルアドレステーブルの一覧を表示します。
- f fname : 指定ファイルfnameの内容をローディングします。  
fnameにはsvサブコマンドで出力したファイルだけ指定できます。

<結果>

ldによってリソースのローディングを行った場合、ローディング終了時に状況を表示します。

ldオプションごとの結果について以下に説明します。

(1) -t tname (プログラムの個別ロード)

1行目は本体のアドレス、2行目以降はタスクの生成/削除によりダウンロードを行ったTCBのアドレスが表示されます。

タスクがマルチタスクの場合は、ダウンロードするすべてのTCBのアドレスが表示されます。

```
address : 0x*****-0x*****
address : 0x*****-0x*****
```

また、分割領域の確保/解放、プログラムのBSS領域の確保/解放、タスク生成/削除を行った場合は下記メッセージが出力されます。

```
allocate : ANAME (0x*****-0x*****)
free : ANAME (0x*****-0x*****)
allocate : bss (0x*****-0x*****)
free : bss (0x*****-0x*****)
ctask : TN (STACK=0x*****-0x*****)
dtask : TN (STACK=0x*****-0x*****)
```

ANAMEは確保/解放した分割領域名を、TNはタスク生成/削除を行ったタスクのタスク番号を表します。

(2) -s sname (サブプログラムの個別ロード)

ローディング対象のサブプログラムの種別により、結果表示が異なります。

● IRSUBの場合

ダウンロードしたアドレスの範囲が下記フォーマットで表示されます。

1行目は本体のアドレス、2行目以降は間接リンクサブプログラムアドレステーブルのアドレスが表示されます。

IRSUBがマルチエントリ化されている場合は、ダウンロードするすべての間接リンクサブプログラムアドレステーブルのアドレスが表示されます。

```
address : 0x*****-0x*****
address : 0x*****-0x*****
```

## ● 組み込みサブルーチンの場合

ダウンロードしたアドレスの範囲および組み込みサブルーチン管理テーブルのエントリが下記フォーマットで表示されます。

addressはダウンロードした組み込みサブルーチン本体のアドレスを、point, entはダウンロードした組み込みサブルーチン管理テーブルのエントリを表す組み込み箇所およびエントリ番号を表示します。

```
address : 0x*****-0x*****
point, ent : POINT, N
```

POINTには、組み込み箇所を表すCPES、IES、EAS、INS、EXS、ABS、PCKS、MODES、WDTES、NXS、XEAS、BRKS、ADTSのいずれかの文字列を表示します。

Nには、エントリ番号を表す1~4の数値を表示します。

## (3) -g gname (グローバルの個別ロード)

ダウンロードしたアドレスの範囲が下記フォーマットで表示されます。

間接リンクグローバルの場合は、間接リンクグローバルアドレステーブルのアドレスも表示されます。

gnameが複数の間接リンクグローバルとして登録されている場合は、すべての間接リンクグローバルアドレステーブルのアドレスが表示されます。

```
address : 0x*****-0x*****
```

## (4) -a aname (分割名称の個別ロード)

ダウンロードしたアドレスの範囲が下記フォーマットで表示されます。

```
address : 0x*****-0x*****
```

## (5) -m addr, len (アドレス指定のロード)

ダウンロードしたアドレスの範囲が下記フォーマットで表示されます。

```
address : 0x*****-0x*****
```

(6) -T tno (タスク生成/削除のPCsメモリへの反映)

- タスクの生成/削除のPCsメモリへの反映を行った場合  
タスクの生成/削除によりダウンロードを行ったTCBのアドレスが表示されます。

```
address : 0x*****-0x*****
```

- PCsメモリに反映が必要なタスクの一覧表示を行った場合  
-Tオプションだけ指定した場合は、下記フォーマットでPCsメモリに反映が必要なタスクの一覧が表示されます。

```
TN   TNAME           PNAME           STATUS
tn   tname           pname           stat
```

上記一覧で、tnはタスク番号、tnameはタスク名称、pnameはプログラム名称、statはタスクの管理状態を表わします。管理状態の意味は、表2-13を参照してください。

タスクの管理状態がunmatchであり、開発系マシン上の管理とPCs上の管理で、異なるTNが与えられている場合は、開発系マシン上のタスクのTNが表示されます。

表2-13 リソースの管理状態

状態	意味
non-exist	開発系マシンにもPCsにも未登録の状態です。
not-build	サブプログラムでロードされているが、buildされていない状態です。
not-build2	PCsにダウンロード後にdbuildした場合は、not-build2になります。
defined-POC	開発系マシンにだけ登録されている状態です。
defined	開発系マシンにもPCsにも登録されている状態です。
defined-CON	PCsにだけ登録されている状態です。 PCsにダウンロード後に開発系マシンから削除した状態です。
unmatch	開発系マシンにもPCsにも登録されているが、整合の取れていない状態です。 開発系マシンで削除・再登録を行い、ダウンロードしていない状態です。

(7) -U point, ent (組み込みサブルーチン管理テーブルのダウンロード)

- 組み込みサブルーチン管理テーブルをダウンロードした場合  
更新した組み込みサブルーチン管理テーブルのエントリは下記フォーマットで表示されます。

```
address : 0x*****-0x*****
```

- PCsメモリに反映が必要な組み込みサブルーチン管理テーブル一覧表示を行った場合  
-Uオプションだけ指定した場合は、下記フォーマットでPCsメモリに反映の必要な組み込みサブルーチン管理テーブルの一覧が表示されます。

POINT	ENT	SUBNAME	STATUS
pnt	eno	subname	stat

上記表示で、pntは組み込み箇所、enoはエントリ番号、subnameはサブプログラム名、statは組み込みサブルーチン管理テーブルのエントリの管理状態を表します。管理状態の意味については、表2-13を参照してください。

組み込みサブルーチン管理テーブルのエントリの管理状態には、defined-POC、defined-CON、not-build、not-build2、unmatchのいずれかが表示されます。

組み込みサブルーチン管理テーブルのエントリの管理状態がunmatch状態であり、開発系マシン上の管理とPCs上の管理で、異なる組み込み箇所が与えられている場合は、開発系マシン上の組み込み箇所が表示されます。

(8) -S sno (間接リンクサブプログラムアドレステーブルのダウンロード)

- 間接リンクサブプログラムアドレステーブルをダウンロードした場合  
ダウンロードした間接リンクサブプログラムアドレステーブルのアドレスは、下記フォーマットで表示されます。

```
address : 0x*****-0x*****
```

- PCsメモリに反映が必要な間接リンクサブプログラムアドレステーブルの一覧表示を行った場合  
-Sオプションだけ指定した場合は、下記フォーマットでPCsメモリに反映の必要な間接リンクサブプログラムアドレステーブル一覧が表示されます。

IRSBNO	ENTNAME	SUBNAME	STATUS
sno	entname	subname	stat

上記一覧で、snoはIRSUB番号、entnameはエントリ名、subnameはサブプログラム名、statはIRSUBTのエントリの管理状態を表示しています。管理状態の意味については、表2-13を参照してください。

IRSUBTのエントリの管理状態がunmatch状態であり、開発系マシン上の管理とPCs上の管理で、異なるIRSUB番号が与えられている場合は、開発系マシン上のIRSUB番号が表示されません。

また、サブプログラム名が異なる場合は、開発系マシン上のサブプログラム名が表示されません。

(9) -G gno (間接リンクグローバルアドレステーブルのダウンロード)

- 間接リンクグローバルアドレステーブルのダウンロード

ダウンロードした間接リンクグローバルアドレステーブル (間接リンクグローバルアドレステーブル) のアドレスが下記フォーマットで表示されます。

```
address : 0x*****-0x*****
```

- PCsメモリに反映が必要な間接リンクグローバルアドレステーブルの一覧表示を行った場合 -Gオプションだけ指定した場合は、下記フォーマットでPCsメモリに反映の必要な間接リンクグローバルの一覧が表示されます。

IRGLBNO	ENTNAME	SNAME	STATUS
gno	entname	sname	stat

上記一覧で、**gno**は間接リンクグローバル番号、**entname**はエントリ名、**sname**は細分割領域名、**stat**は間接リンクグローバルアドレステーブルのエントリの管理状態を表示します。管理状態の意味については表 2-13を参照してください。

間接リンクグローバルアドレステーブルのエントリの管理状態が**unmatch**状態であり、開発系マシン上の管理とPCs上の管理で、異なる間接リンクグローバル番号が与えられている場合は、開発系マシン上の間接リンクグローバル番号が表示されます。

また、細分割領域名が異なる場合は、開発系マシン上の細分割領域名が表示されます。

(10) -f fname (指定ファイルからのロード)

ダウンロードしたアドレスの範囲が下記フォーマットで表示されます。

```
address : 0x*****-0x*****
```

<注意事項>

- リソースの入れ換え時には、リソースを入れ換えても問題が発生しないように、リソースを参照するタスクをすべてabortするなどの操作が必要となります。
- -aオプション、-mオプションでリソースをダウンロードした場合、-T、-U、-S、-Gオプションで管理テーブルをダウンロードおよびタスクを生成／削除してください。
- -fオプションでファイル名を指定せず、バックアップファイルの存在しないリソースまたは領域を指定した場合はエラーメッセージを表示し、サブコマンドを終了します。
- 障害回復処理をスキップした場合、svrplコマンドで一括ロードするまで、ldサブコマンドによる個別ロード機能は使用できなくなります。
- ブレークポイントを設定しているプログラムまたはサブプログラムはダウンロードできません。

<名前>

sv - リソースの個別バックアップ

<形式>

```
sv {-t tname [-f fname]}
sv {-s sname [-f fname]}
sv {-g gname [-f fname]}
sv {-a aname [-f fname]}
sv {-m addr,len [-f fname]}
```

<機能説明>

svは、PCsのメモリの内容をバックアップファイルに転送します。転送する対象は以下に示すオプションで指示します。

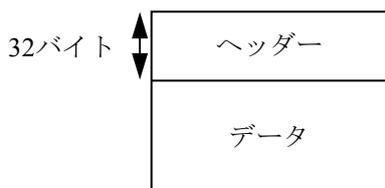
- t tname : tnameで指定されたプログラムのtext部およびdata部を転送します。
- s sname : snameで指定されたサブプログラムのtext部およびdata部を転送します。
- g gname : gnameで指定された細分割領域の内容を転送します。
- a aname : anameで指定された分割領域の内容を転送します。  
anameで指定する領域はGLB領域に定義された分割領域にしてください。
- m addr,len : 先頭アドレス (addr) 、バイト数 (len) 指定で転送します。  
指定するアドレスの範囲がマッピングされていない空間を含む場合は、転送しません。
- f fname : 指定ファイルfnameに転送します。  
この指定を省略した場合は、バックアップファイルに転送します。  
セーブ途中に異常が発生した場合は、指定ファイルを削除して終了します。

<結果>

転送したアドレスの範囲が下記フォーマットで表示されます。

address : 0x\*\*\*\*\*-0x\*\*\*\*\*

バックアップファイル以外に転送した場合のファイルのフォーマットを以下に示します。



32バイトのヘッダー+バイナリデータです。  
ヘッダーの内容は下記文字列となります。  
S10V△\*\*\*\*\*△\*\*\*\*\*¥0¥0...¥0¥0  
先頭アドレス 容量  
\*\*\*\*\*は8桁の16進数、¥0はNULL(0)を、△は空白文字(0x20)を表します。

<注意事項>

- -fオプションでファイル名称を指定せずに、バックアップファイルが存在しない領域を指定した場合は、エラーメッセージを出力し、サブコマンドを終了します。
- -fオプションでファイル名称を指定せずに、パッケージの共有ソースを指定した場合は、エラーメッセージを出力し、サブコマンドを終了します。

<名前>

cm - バックアップファイルとPCsメモリの内容比較

<形式>

cm {-t tname}  
cm {-s sname}  
cm {-g gname}  
cm {-a aname}  
cm {-m addr,len}  
cm {-f fname}

<機能説明>

cmは、バックアップファイルとPCsの主メモリの内容を比較します。比較する対象は以下に示すオプションで指示します。

- t tname : tnameで指定されたプログラムのtext部およびdata部を比較します。
- s sname : snameで指定されたサブプログラムのtext部およびdata部を比較します。
- g gname : gnameで指定された細分割領域の内容を比較します。
- a aname : anameで指定された分割領域の内容を比較します。  
anameとして指定する領域は、GLB領域に定義された分割領域にしてください。
- m addr,len : 先頭アドレス (addr)、バイト数 (len) 指定で比較します。  
指定するアドレスの範囲がアロケートされていない空間を含む場合は、比較しません。
- f fname : 指定ファイルfnameとメモリの内容を比較します (svで出力したファイルだけ使用できます)。

<結果>

比較結果が正常の場合、アドレス範囲が下記フォーマットで表示されます。

```
address : 0x*****-0x*****  
++ compare OK ++
```

比較で違いが見つかった場合には、ワード (2バイト) 単位で以下のように表示されます。

```
address : 0x*****-0x*****  
address = 0x***** memory data = 0x**** backup data = 0x****
```

<注意事項>

-fオプションでファイル名称を指定せずに、バックアップファイルが存在しない領域を指定した場合は、エラーメッセージを出力し、サブコマンドを終了します。

<名前>

dr — DHP記録許可

<形式>

dr

<機能説明>

drは、svdhpコマンドを起動し、DHPの記録を許可モードにします。

drサブコマンドの詳細は、svdhpコマンドの“-on” オプションの仕様を参照してください。

<名前>

ds — DHP記録禁止

<形式>

ds

<機能説明>

dsは、svdhpコマンドを起動し、DHPの記録を禁止モードにします。

dsサブコマンドの詳細は、svdhpコマンドの“-off” オプションの仕様を参照してください。

<名前>

svdhp — DHPの表示

<形式>

svdhp [-u site] [+count] [-on|-off] [-o fname] [-f fname] [-d fname]

<機能説明>

svdhpサブコマンドは、svdhpコマンドを起動し、DHPを表示します。

svdhpサブコマンドの詳細は、svdhpコマンドを参照してください。

<名前>

svadm — アドレスに対するリソース名称の表示

<形式>

svadm [addr] [-u site]

<機能説明>

svadmサブコマンドは、svadmコマンドを起動し、指定した論理アドレスに対して名称等の情報を表示します。

svadmサブコマンドの詳細は、svadmコマンドを参照してください。

<名前>

si - スタック初期化

<形式>

si tn1[-tn2][,data]

si tname[,data]

<機能説明>

siは、指定されたタスクのスタックを固定パターンで初期化します。指定するパラメータは以下のとおりです。

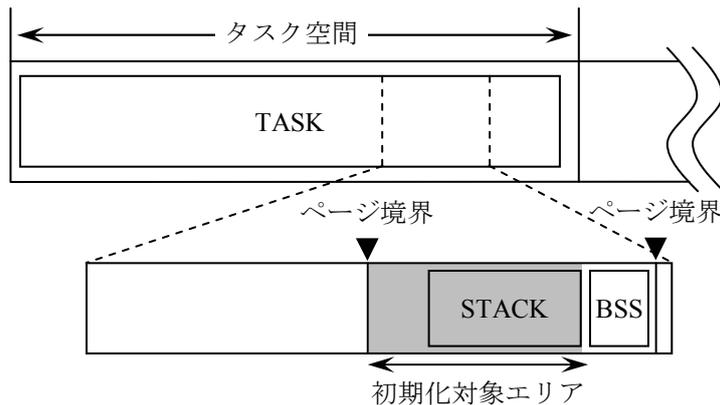
- tn1 : 先頭タスク番号 (1~最大タスク番号)
- tn2 : 最終タスク番号 (1~最大タスク番号)
- tname : タスク名称
- data : 初期化データ (0~0xf)

<結果>

OK(0) : 正常終了

<注意事項>

- 初期化データ指定 (data) を省略した場合、すべてfで初期化します。
- 初期化の対象となるタスクは、DORMANT状態にしておいてください。
- スタックの初期化は、指定したタスクのスタックが存在するページ領域のうち、ページ先頭アドレスからスタック最終アドレスまでの範囲を対象とします。(下図参照)



- このサブコマンド起動時にパラメータを指定しなかった場合、あるいは正しくない引数を指定した場合、下記メッセージを出力し “:” の後でパラメータの入力待ちとなります。また、この状態で [e] または [Enter] キーを押した場合にはサブコマンド処理を終了します。

```
input tn1[-tn2] [,data] or tname [,data]
:
```

<名前>

sp - スタック使用量の表示

<形式>

sp tn1[-tn2][,data]

sp tname[,data]

<機能説明>

spは、指定されたタスクのスタック使用量を表示します。指定するパラメータは以下のとおりです。

tn1 : 先頭タスク番号 (1~最大タスク番号)

tn2 : 最終タスク番号 (1~最大タスク番号)

tname : タスク名称

data : チェックパターン指定 (0~0xf)

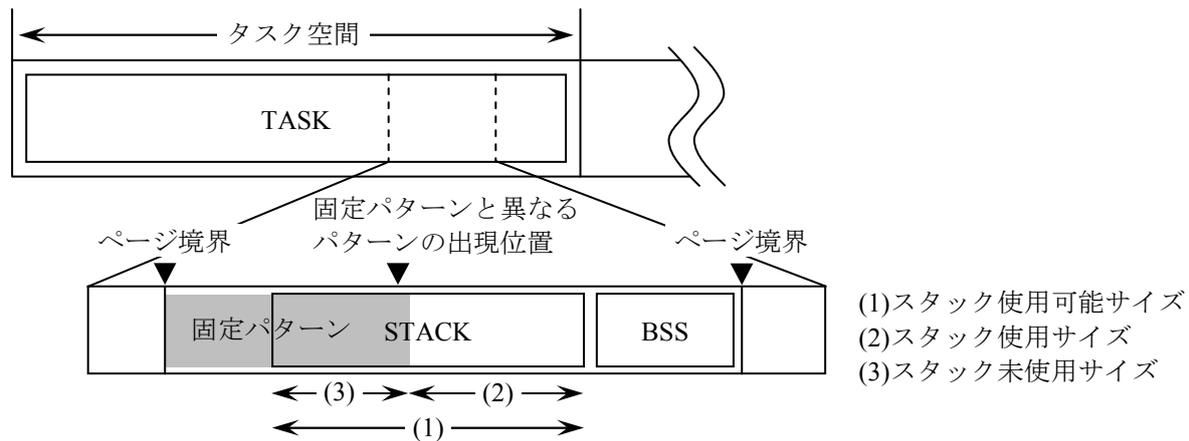
<結果>

タスクのスタック使用量は、下記フォーマットで表示します。

tn=***	total:*****bytes	use:*****bytes	rest:*****bytes
タスク番号	スタック使用可能サイズ	スタック使用サイズ	スタック未使用サイズ

<注意事項>

- チェックパターン指定 (data) に指定する値は、siの初期化データに対応した値を指定してください。  
 チェックパターン指定 (data) を省略した場合、0xfが指定されたものとして扱います。
- スタック使用量は、指定したタスクが使用するスタックのページ領域に対して、チェックパターン指定 (data) で指定した値と、異なるパターンが現れたアドレスを基に算出します。このため、スタックの先頭が、チェックパターンと同一のパターンであった場合、正しいスタック使用量の表示はできません。
- spが表示する内容と、タスクの動作空間との対応を下図に示します。



- このサブコマンド起動時にパラメータを指定しなかった場合、あるいは正しくない引数を指定した場合、下記メッセージを出力し “:” の後でパラメータの入力待ちとなります。  
 また、この状態で [e] または [Enter] キーを押した場合にはサブコマンド処理を終了します。

```
input tn[-tn2] [-data] or tname [-data]
:
```

<名前>

ps — デバッグ文の表示開始

<形式>

ps

<機能説明>

psは、プログラム内のrs\_printf()にて出力するメッセージの端末表示を開始することを指示します。

psサブコマンドの実行前に出力されたデバッグ文は表示されません。

(rs\_printf()は、「付録B ライブラリ」を参照してください。)

<結果>

処理が正常に終了後、デバッグ文の表示を開始します。

<注意事項>

デバッグ文を格納するバッファが足りなくなった場合、デバッグ文が出力されない場合があります。

<名前>

pe — デバッグ文の表示終了

<形式>

pe

<機能説明>

peは、プログラム内のrs\_printf()にて出力するメッセージの端末表示を終了することを指示します。

(rs\_printf()は、「付録B ライブラリ」を参照してください。)

<結果>

処理が正常に終了後、デバッグ文の表示を終了します。

## 第7章 svdebug（オンラインデバッガ）とデバッグ支援コマンド

### <名前>

ver - CMUのバージョン表示

### <形式>

ver

### <機能説明>

verは、CMUのバージョンを表示します。

### <結果>

処理が正常に終了したとき、CMUのバージョンが下記フォーマットで表示されます。

CMU Ver. \*\* Rev. \*\*

<名前>

help - サブコマンド一覧表示

<形式>

help

<機能説明>

helpは、svdebugのサブコマンド一覧を表示します。

サブコマンド名と機能概要が、以下のフォーマットで表示されます。

<Sub Command>	<Function>
qu	...task queue
ab	...task abort
re	...task release
ta	...task status
su	...task suspend
rs	...task resume
tm	...task timer request
ct	...task cancel timer
sht	...task show timer
md	...memory print/patch appointed address
sd	...memory print/patch appointed name
mcp	...memory copy
mmv	...memory move
mf	...memory fill
bs	...bit data set
bg	...bit data get
el	...system error display
ss	...system status display
st	...set timer
gt	...get timer
br	...break point set
rb	...break point reset
rd	...register print
rr	...register set
go	...break point restart
as	...adress detect trap set
ac	...adress detect trap reset
ld	...memory load
sv	...memory save
cm	...memory compare
dr	...DHP regist start
ds	...DHP regist stop
svdhp	...DHP data display
svadm	...address -> sarea name
si	...stack initial
sp	...stack print
ps	...debug message print start
pe	...debug message print end
ver	...CMU version print
q	...debughr end
!	...execute external command
help	...command menu display

<名前>

q — デバッガの終了

<形式>

q

<機能説明>

qは、デバッガを終了させます。ただし、ブレークポイントが設定されている場合には、設定されているブレークポイントを表示してキー入力待ちになります。

<注意事項>

“ブレークポイントが設定されています”と表示された場合、ブレーク中の場合はgoサブコマンド、ブレーク中でない場合はrbサブコマンドにてブレークポイントを解除してから、このサブコマンドを再発行してください。

<名前>

! — svdebug実行時の開発系マシン上のコマンドの実行

<形式>

! 開発系マシン上のコマンド

<機能説明>

!以降を開発系マシン上のコマンドとして実行します。

## &lt;名前&gt;

svelog - エラーログ情報出力

## &lt;形式&gt;

```
svelog [-u site] [-f format] [-logno] [+case] [-d fname] [-o fname]
      {
      エラーログ1画面分表示
      }
      { p }
      { - }
      { ±nl }
      { n }
      { 何も入れず }
      { q }
```

## &lt;機能説明&gt;

svelogは、コントローラ内のエラーログバッファからエラーログ情報を読み出し、エラーログ情報を出力します。オプションには以下のものがあります。

- u site : 処理対象となるサイト名称を指定します。このオプション省略時は、環境変数RSSITEに設定されたサイトに対して処理します。
- f format : エラーログ情報の出力形式を指定します。  
以下の形式があります。省略時は“m”になります。  
s : エラー情報を簡略化した短い形式で出力します。  
m : エラー情報をすべて出力します。  
l : エラーの情報に加えDHPトレース情報も出力します。
- logno : lognoで指定されたログ番号のエラーログ情報を出力します。
- +case : 表示ログケース数を指定します。このオプション省略時は、最新のケースから順にすべてのエラーログ情報を表示します。
- d fname : 画面操作履歴 (オペレーション結果) を格納するファイルを指定します。  
同名のファイルがすでに存在した場合、そのファイルに画面操作履歴を追加して格納します。
- o fname : エラーログ情報を格納するファイルを指定します。  
同名のファイルがすでに存在した場合、そのファイルを消却し、新しいファイルを作ります。

また、エラーログ情報の表示は、以下に示すように閲覧用のコマンドにより制御します。

- p, 何も入れず : 次ページ表示
- : 前ページ表示
- ±n : n行前 (-) または後 (+) の行から表示します。
- n : n番目の行から表示します。
- q : エラーログ表示を終了します。

### <使用上の留意点>

- svdebugは、ユーザタスクがRUN/STOP状態において動作できます。
- -lognoで指定されたログ番号が、最新のエラーログ情報より大きい場合は、最新のエラーログ情報が表示されます。
- -logno、+caseが同時に指定された場合、lognoで指定されたログ番号のケースより+caseで指定されたケース数のログ情報を表示します。
- -f formatの省略時の扱いは、mとなります。

### <終了コード>

次の終了コードを返します。

- 0 : 正常終了
- 1 : パラメータエラー
- 2 : 通信エラー
- 3 : シグナル受信

## &lt;名前&gt;

svdhp — DHPトレース情報の表示

## &lt;形式&gt;

```
svdhp [-u site] [+count] [-on|-off] [-d fname] [-o fname] [-f fname]
```

DHPの1画面分表示

```
{ p }
{ - }
{ ±nl }
{ n }
{ 何も入れず }
{ q }
```

## &lt;機能説明&gt;

svdhpは、PCs内DHPトレースバッファに記録されているDHPトレース情報を時刻の新しい順に表示します。オプションには以下のものがあります。

- u site : 処理対象となるサイト名称を指定します。このオプション省略時は、環境変数“RSSITE”に設定されたサイトに対して処理します。
- +count : countで指定されたトレース情報を出力します。  
このオプション省略時は、すべてのトレース情報を出力します。
- on : DHPの記録を許可モードにします。
- off : DHPの記録を禁止モードにします。
- d fname : 画面操作履歴 (オペレーション結果) を格納するファイルを指定します。  
同名のファイルがすでに存在した場合、そのファイルに画面表示を追加して格納します。
- o fname : DHP表示結果を格納するファイルを指定します。  
同名のファイルがすでに存在した場合、そのファイルを消却し、新しいファイルを作ります。
- f fname : DHPログ入力ファイル名称を指定します。  
サイト名称には、S10Vのサイト名称を指定してください。  
入力ファイルは、dhpreadマクロでDHPログをGLBに格納し、デバッガのsvサブコマンドでGLBのDHPログを格納したファイルだけが対象となります。  
dhpreadマクロの詳細は、「S10Vソフト 機能仕様書 (CPMS編)」 (図番: 331SR26517) を参照してください。

dhpreadを使用したプログラム例を以下に示します。

<プログラム例>

```
#include <cpms_dhp.h>
extern char dhp_g[4096];
main( ) {
    long size;
    size=4096;
    dhpread(dhp_g, &size);
}
```

また、DHPトレース情報の表示は、以下に示すように閲覧用のコマンドにより制御します。

- p, 何も入れず : 次ページ表示
- : 前ページ表示
- ±nl : nl行前 (-) または後 (+) の行から表示します。
- n : n番目の行から表示します。
- q : DHP表示を終了します。

svdhpで出力する情報を以下に示します。

```
Debugging helper trace list      [XXXXXX]   Tue Oct 31 15:37:05 2001
                                ⑦

DHP TIME      EVENT          TN LV DATA1  DATA2  DATA3  DATA4  DATA5
nnnn tt.tttttt ssssssssssss xxx xx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
①      ②          ③          ⑤ ⑥          ④
```

- ① DHPトレース情報の表示番号
- ② トレース時刻  
     tt.tttttt  
     |  |  
     秒 1マイクロ秒まで出力
- ③ トレースポイント種別
- ④ トレースデータ (16進で出力)
- ⑤ タスク番号
- ⑥ 優先レベル
- ⑦ サイト名または-fオプション指定時のファイル名

<使用上の留意点>

- -on、-offオプション指定時に、DHPトレース情報は出力しません。

<注意事項>

次の終了コードを返します。

- 0 : 正常終了
- 1 : パラメータエラー
- 2 : 通信エラー
- 3 : シグナル受信

## &lt;名前&gt;

svcpunow - PU負荷率の表示

## &lt;形式&gt;

svcpunow [-u site] [-t second]

## &lt;機能説明&gt;

svcpunowは、指定サイト (PU) のIDLE時間の累積と時刻を取り込み、PU負荷率を表示します。

(計算式)

PU負荷率 = (測定時間 - IDLE時間) / 測定時間

オプションには以下のものがあります。

-u site : 対象となるサイトを指定します。

省略時は環境変数RSSITEに設定されているサイトについて処理します。

-t second : PU負荷率測定時間を秒 (1~3600) で指定します。

省略時はデフォルト=1秒となります。

## &lt;使用上の留意点&gt;

svcpunowコマンドを実行するとき、このコマンドがすでに実行されているときは要求を受け付けません。

## &lt;終了コード&gt;

次の終了コードを返します。

0 : 正常終了

1 : 異常終了

2 : 通信異常

3 : シグナル受信

## &lt;出力フォーマット&gt;

出力結果は、下記ようになります。

```
2002/04/24 17:57:33 SITE=pcs01b_cp ** 1 second wait **
CPU(pcs01b_cp) load ratio = 0.06%
```

<名前>

svtimex - タスク稼働率表示

<形式>

svtimex [-u site] [tn] [-t second]  
[tname]

<機能説明>

svtimexは、測定時間内におけるタスクの実行回数、実行時間の累積と時刻を取り込んだタスク稼働率を表示します。

オプションには以下のものがあります。

-u site : 対象となるサイトを指定します。

省略時は環境変数RSSITEに設定されているサイトについて処理します。

tn : タスク番号 (1~255) を10進または16進 (0xを前置) で指定します。

tname : タスク名称を指定します。

tnまたはtnameを省略すると会話形式となり最初に測定時間の入力を促します。ここで、1~86400の間で測定時間を入力すると、次にタスク名称または番号の入力を促します。ここで  
の入力は最大10タスク分の設定ができます。svtimexコマンドを実行させたいときは何も入力  
せずに [Enter] キーを押してください。

-t second : タスク稼働率測定時間を秒 (1~86400) で指定します。

省略時はデフォルト=1秒となります。

<使用上の留意点>

- ・-tオプションにより測定時間を指定した場合は、タスク番号 (tn) またはタスク名称 (tname) を合わせて指定してください。
- ・svtimexコマンドを実行するとき、このコマンドがすでに実行されているときは要求を受け付けません。
- ・tnオプションとtnameオプションは、同時には指定できません。会話形式によるタスク名称またはタスク番号の指定は最大10個まで指定できます。

<終了コード>

次の終了コードを返します。

0 : 正常終了

1 : 異常終了

2 : 通信異常

3 : シグナル受信

<出力フォーマット>

出力結果は、下記のようになります。

```
2002/04/24 18:02:18 SITE=pcs01b_cp ** 1 second wait **
sist(255) load ratio=0.00% execute count=0 total time=0.000sec average time=0.000sec
```

## 付録

## 付録A プログラムで使用できる名称

あらかじめシステムで用意されたサブルーチンの名称と同一名称のプログラムを使用する場合、注意が必要です。システムで用意されたサブルーチンは、すべてライブラリファイルに納められています。svloadにおいて-lオプションを指定すると簡単に結合できます。しかし、システムのサブルーチンと同一名称である場合、そのサブルーチンを定義するオブジェクトファイルをsvloadで引数に指定しないとライブラリファイルより同一名称のサブルーチンが結合されます。

以下に各システム用のライブラリファイルとそこで定義されている名称を示します。名称が重複しないようにプログラミングしてください。

重複した名称を使用する場合、ライブラリファイルの指定順序を結合したいオブジェクトファイルの後にすればライブラリファイルから結合されません。

以降にシステムで用意されたサブルーチンをライブラリ別に示します。

なお、\_（アンダーライン）で始まる名称はシステムで予約した名称ですので使用しないでください。ライブラリの構成は下記のようになっています。

ライブラリ名	ライブラリの内容	備考
libsh4nbmzz.lib	C言語用のサブルーチン群 非正規化数：非正規化数 値の丸め方：切り捨て	詳細はshcコンパイラマニュアルを参照してください。
libsh4nbmdn.lib	C言語用のサブルーチン群 非正規化数：0 値の丸め方：切り捨て	
libfirad.lib	間接リンクアドレス参照サブルーチン群	
libcpms.lib	CPMSマクロリンケージサブルーチン群	詳細は各マニュアルを参照してください。
libsysctl.lib	システム制御用サブルーチン群	
libcpr_hr.lib	リモートCPU制御サブルーチン群	
libcycm.lib	サイクリック通信サブルーチン群	
libnet.lib	ソケット通信サブルーチン群	

&lt;libsh4nbmzz.lib&gt;

&lt;libsh4nbmdn.lib&gt;

atof	memcpy	strlen	fpcheck	fmod	tan
fpgetmask	memset	strncat	fpchecko	log	tanh
fpgetround	modf	strncmp	acos	log10	
fpgetsticky	sscanf	strncpy	asin	matherr	
fpsetmask	sprintf	strpbrk	atan	pow	
fpsetround	strcat	strrchr	atan2	cos	
fpsetsticky	strchr	strspn	ceil	sin	
frexp	strcmp	strtod	exp	cosh	
ldexp	strcpy	strtol	fabs	sinh	
memchr	strcspn	vsprintf	floor	sqrt	

&lt;libfirad.lib&gt;

irglbad	irsubad
---------	---------

&lt;libcpms.lib&gt;

abort	dhpread	printf	stime	elset
arsum	exit	prsrv	susp	geterrno
asusp	free	queue	timer	gettimebase
chap	gfact	read	usrdhpset	gtkmem
chml	cpms_ginfo	resume_env	usrelset	getsysinfo
close	gtime	rleas	wait	gettaskinfo
cpms_copy	ioctl	rserv	write	usrdhp
ctime	open	rsum	chkbmem	usrel
delay	pfree	save_env	chktaer	wrtmem
dhpctl	post	sfact	dhpset	

## 付録A プログラムで使用できる名称

### <libsysctl.lib>

cardoff	cardstat	dsuctl	ptnwrite	sysRegWrite
slotewrite	dcctl	dsustat	regLRread	sysdo
cardon	dcmrb	ledctl	hdctl	wdtset
sloteclear	dcmstat	pioctl	sysRegRead	

### <libcpr\_hr.lib>

cpr_grstart	cpr_stop	cpr_rpl	cpr_rrb	cpr_chk
cpr_setipa	setipa_mkdata	setipa_parchk	setipa_sndrcv	setipa_dupchk
setipa_inet_addr	setipa_send	setipa_headchk	setipa_rcv	

### <libcym.lib>

getcym_hr	restcym_hr	wcym_hr	cycinf_hr	stcym_hr
rcym_hr				

### <libnet.lib>

accept	bind	connect	getsockopt	listen
recv	recvfrom	send	sendto	setsockopt
shutdown	socket			

## 付録B ライブラリ

## (1) ライブラリファイルの指定条件

ライブラリファイルをsvloadで指定する場合、表A-1に示すようにライブラリを指定してください。

表A-1 ライブラリの指定条件

条件	ライブラリ名	loadhrでの指定方法	備考
作成したプログラムがC言語で記述されている場合（「付録A プログラムで使用できる名称」に示す該当のライブラリ内の関数を使用している場合）	libcvt.lib libmrs.lib libcrs.lib	-lcvt -lmrs -lcrs	「ソフトウェアマニュアル CPMS概説&マクロ仕様（マニュアル番号 SVJ-3-201）」を参照してください。
作成したプログラムでCPMSマクロを使用している場合	libcpms.lib	-lcpms	
間接リンクアドレスを参照している場合または間接リンクサブルーチンを参照している場合	libfirad.lib	-lfirad	「(3) 間接リンクアドレス参照サブルーチン」を参照してください。
ユーザ専用のライブラリを使用する場合	ユーザライブラリ	-l文字列またはライブラリ名	

## (2) ライブラリの指定順序

svloadでライブラリを指定する場合、下記の点に注意してください。

- 共通サブルーチンを含むライブラリは、できるだけ後の方で指定してください。
- 指定した複数のライブラリの中に同一名称がある場合、結合したいオブジェクトファイルのあるライブラリを前に指定してください。
- システム用ライブラリは下記の順に指定してください。
  - ・ libcvt.libライブラリを使用する場合  
-lcvt -lmrs -lcrs
  - ・ libmrs.libライブラリを使用する場合  
-lmrs -lcrs
  - ・ libcrs.lib、libcpms.lib、またはlibfirad.libライブラリを使用する場合  
-lcrs、-lcpms、-lfiradだけを指定します。

(3) 間接リンクアドレス参照サブルーチン

下記モジュールを利用する場合は、libfirad.libをリンクします。

<名前>

irglbad

<形式>

int \*irglbad (no)

int no;

<機能説明>

irglbadは、noが間接リンクグローバル番号（1～最大グローバル番号）のとき、対応するグローバルアドレスを返します。

<リターンコード>

noが登録済みの間接リンクグローバル番号のとき、対応するグローバルアドレスを返します。

noが未登録の間接リンクグローバル番号のとき、0を返します。（間接リンクグローバルアドレス管理テーブルの先頭アドレスは、noが1の場合、0xafd00004番地です。noが1増すごとに、このアドレスに4を加算したアドレスが、noに対応する間接リンクグローバル管理テーブルのアドレスとなります。）

<名前>

irsubad

<形式>

int \*irsubad (no)

int no;

<機能説明>

irsubadは、noがIRSUB番号（1～最大IRSUB番号）のとき、対応するIRSUBアドレスを返します。

<リターンコード>

noが登録済みのIRSUB番号のとき、対応するIRSUBアドレスを返します。

noが未登録のIRSUB番号のとき、0を返します。

## (4) メッセージ出力ルーチン

下記モジュールを利用する場合は、libcpms.libをリンクします。

## &lt;名前&gt;

rs\_printf

## &lt;形式&gt;

```
int rs_printf (buf, fmt, p1, p2, ...,p10)
```

```
char *buff;
```

```
char *fmt;
```

```
long p1, p2, ...,p10;
```

## &lt;機能説明&gt;

rs\_printfは、データを書式に従ってメッセージに変換し、OS内のメッセージ用バッファエリアにメッセージを書き込みます。

メッセージは、debughrコマンドのpsサブコマンドで読み出すことができます。

変換後のメッセージの文字列サイズ（バイト数）が1～1024の範囲となるように指定してください。変換後のメッセージの文字列サイズ（バイト数）が1～1024の範囲以外の場合は、パラメータ異常となり書き込みは行われません。

## &lt;パラメータ&gt;

buff : 書き込むメッセージを一時的に格納するメモリ領域の先頭アドレスを指定します。

fmt : 書式を示す文字列を格納したメモリ領域の先頭アドレスを指定します。

p1～p10 : データを指定します。

## &lt;リターンコード&gt;

正常終了した場合は、変換後のメッセージの文字列サイズ（バイト数）が返されます。

それ以外の場合は、以下のリターンコードが返されます。

0 : OS内のメッセージ用バッファエリアが満杯で書き込みできません。

-1 : パラメータ異常で書き込みできませんでした。

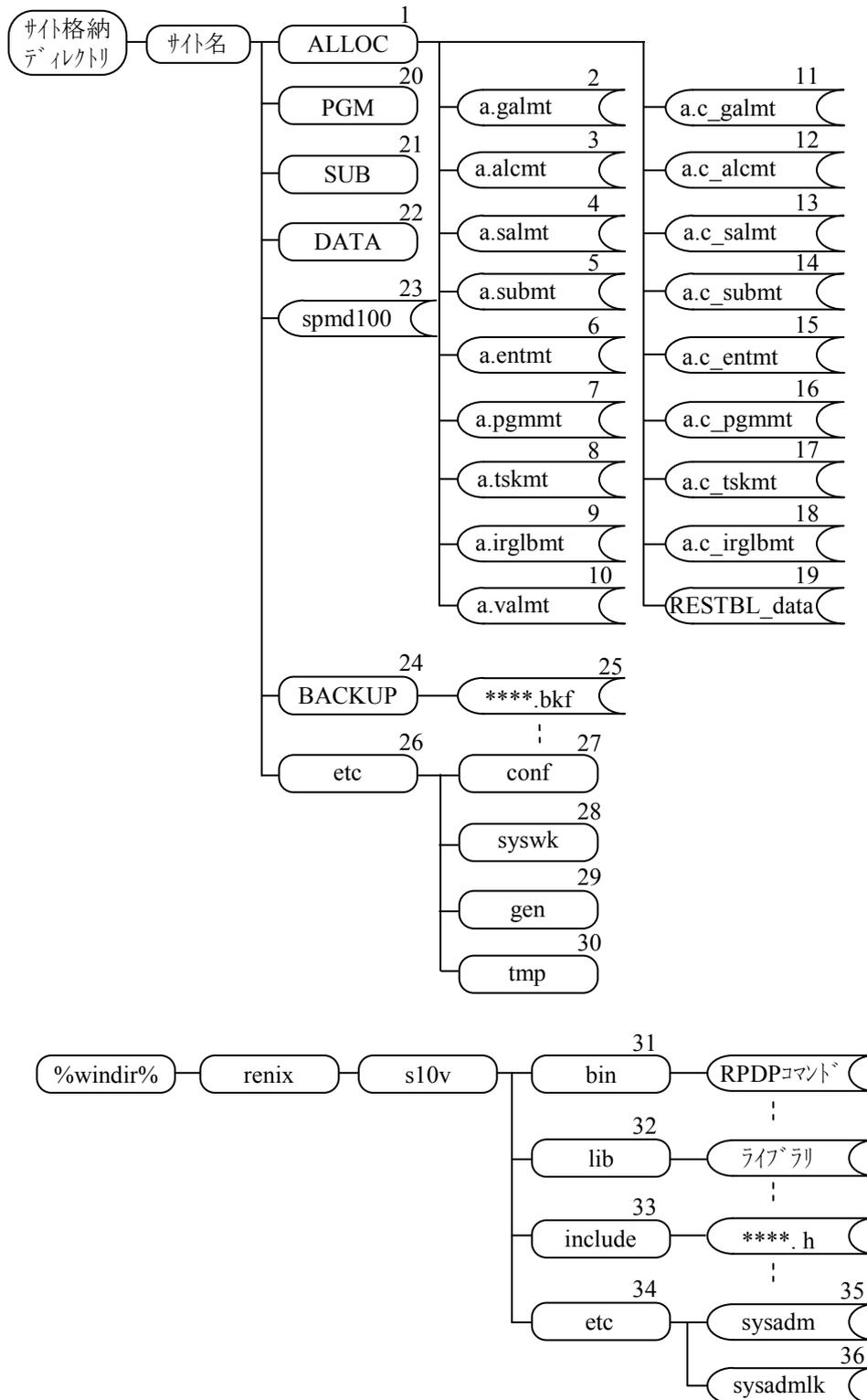
-2 : メッセージの書き込み異常で書き込みできませんでした。

## &lt;注意事項&gt;

- ・浮動小数点データを出力する場合は、sprintfライブラリ（C/C++）を使用して文字列に変換後、このルーチンでメッセージ書き込みを行ってください。
- ・デバッグ時だけ、使用してください。

## 付録C サイト管理ファイル

以下に、サイトを管理するファイルのディレクトリ構成とファイルの説明を示します。



No.	ファイル名/ ディレクトリ名	名称	説明	備考
1	ALLOC	アロケータ管理 テーブル格納 ディレクトリ	アロケータ管理テーブルを格納するディレクトリです。	
2	a.galmt	garea管理ファイル	論理空間内の領域の名称、大きさなどを管理します。	
3	a.alcmt	area管理ファイル	タスクのtext、dataサブプログラムのtext、dataグローバル内に確保したAREA（分割領域）を管理します。	system.uファイル内MAXAREAで定義した数のエントリを含みます。
4	a.salmt	sarea管理ファイル	グローバル内AREA（分割領域）に確保したSAREA（細分割領域）を管理します。	system.uファイル内MAXSAREAで定義した数を加えたエントリを含みます。
5	a.submt	サブプログラム管理ファイル	サブプログラム（IRSUB、組み込みサブルーチン）を管理します。	system.uファイル内ENTMT_MAXENTで定義した数に組み込みサブルーチンの最大数を加えた分のエントリを含みます。
6	a.entmt	IRSUB管理ファイル	間接リンクサブプログラム（IRSUB）を管理します。	system.uファイル内ENTMT_MAXENTで定義した数分のエントリを含みます。
7	a.pgmmt	プログラム管理ファイル	タスクとして登録されるプログラムを管理します。	system.uファイル内PGM_MAXNUMで定義した数分のエントリを含みます。
8	a.tskmt	タスク管理ファイル	タスクを管理します。	
9	a.irglbmt	間接リンクグローバル管理ファイル	間接リンクグローバルの登録を管理します。	system.uファイル内IRG_MAXENTで定義した数分のエントリを含みます。
10	a.valmt	バリュ管理ファイル	バリュの登録を管理します。	system.uファイル内VAL_MAXNUMで定義した数を加えたエントリを含みます。
11	a.c_galmt	PCs側 garea管理ファイル	PCs側の論理空間内の領域の名称、大きさなどを管理します。	svrplコマンドまたはsvdebugコマンドのldサブコマンドでPCsにダウンロードされます。
12	a.c_alcmt	PCs側 area管理ファイル	PCs側のタスクのtext、dataサブプログラムのtext、dataグローバル内に確保したAREA（分割領域）を管理します。	
13	a.c_salmt	PCs側 sarea管理ファイル	PCs側のグローバル内AREA（分割領域）に確保したSAREA（細分割領域）を管理します。	

No.	ファイル名／ディレクトリ名	名称	説明	備考
14	a.c_submt	PCs側サブプログラム管理ファイル	PCs側のサブプログラム（IRSUB、組み込みサブルーチン）を管理します。	svrplコマンドまたはsvdebugコマンドのldサブコマンドでPCsにダウンロードされます。
15	a.c_entmt	PCs側IRSUB管理ファイル	PCs側の間接リンクサブプログラム（IRSUB）を管理します。	
16	a.c_pgmmt	PCs側プログラム管理ファイル	PCs側のタスクとして登録されるプログラムを管理します。	
17	a.c_tskmt	PCs側タスク管理ファイル	PCs側のタスクを管理します。	
18	a.c_irlbmt	PCs側間接リンクグローバル管理ファイル	PCs側の間接リンクグローバルの登録を管理します。	
19	RESTBL_data	リソース管理テーブルデータファイル	サイトのリソースを管理するテーブルデータファイルです。	
20	PGM	プログラム格納ディレクトリ	プログラムのロードモジュールを格納するディレクトリです。	svloadにて-dオプションを指定した場合に格納されます。
21	SUB	サブプログラム格納ディレクトリ	サブプログラムのロードモジュールを格納するディレクトリです。	
22	DATA	グローバル初期値データ格納ディレクトリ	グローバルエリアの初期値データを格納するディレクトリです。	
23	spmd100	SPMファイル	PCsの主メモリ（SPM領域）内のOS、ドライバ用データファイルです。	このファイルは将来用として存在し、svrpl時のローディング対象外です。
24	BACKUP	バックアップファイル格納ディレクトリ	バックアップファイルを格納するディレクトリです。	svdfaにて作成したバックアップファイルが格納されます。
25	****.bkf	バックアップファイル	分割領域単位のPCsメモリ初期値ファイルです。	svrplコマンドでPCs主メモリにローディングされます。
26	etc	システムジェネレーションファイル格納ディレクトリ	システムジェネレーションファイルを格納するディレクトリです。	
27	conf	ユーザ用サイト定義情報ファイル格納ディレクトリ	システムジェネレーション時にユーザが定義するファイルを含むディレクトリです。	adapter.u、memory.uだけユーザ定義が可能です。それ以外のファイルに対して、内容の変更を行わないでください。

(3/3)

No.	ファイル名/ ディレクトリ名	名称	説明	備考
28	syswk	システム用 サイト定義情報 ファイル格納 ディレクトリ	システムジェネレーション時 にシステムが参照するファイ ルを含むディレクトリです。	このディレクトリ内のファイ ルに対して、内容の変更を行 わないでください。
29	gen	構成定義情報 ファイル格納 ディレクトリ	svconfコマンドが出力する構成 定義情報ファイルを格納する ディレクトリです。	
30	tmp	構成定義情報 出力ファイル 格納ディレクトリ	conf下の情報を基に作成した定 義情報出力ファイルの格納 ディレクトリです。	
31	bin	RPDPコマンド 格納ディレクトリ	RPDPコマンドが格納されてい るディレクトリです。	
32	lib	PCs用 ライブラリ格納 ディレクトリ	PCs用ライブラリ群が格納され ているディレクトリです。	
33	include	PCs用 インクルード ファイル格納 ディレクトリ	PCs用インクルードファイルが 格納されているディレクトリ です。	
34	etc	RPDPコマンド 管理ファイル	RPDPコマンド用管理ファイ ルが格納されているディレク トリです。	
35	sysadm	サイト管理ファイル	RPDPコマンド用サイト管理 ファイルです。	
36	sysadmlk	サイトロック ファイル	RPDPコマンド用サイトロック ファイルです。	

## 付録D エラーメッセージ

(1) システムジェネレーション関連コマンド

<svgen>

(1/2)

No.	エラーメッセージ	意味	ユーザの対応
1	Can't Find %s. (EC0102)	指定ファイルがありません。	指定ファイルを見直してください。
2	Can't Find SUBSYS_DIR. (EC0103)	SUBSYS_DIR構文がありません。	
3	Can't Find SITE_NAME. (EC0102)	SITE_NAME構文がありません。	
4	Invalid Site Directory Name. (%s)(EC0102)	サイトディレクトリ名称に異常があります。	
5	Invalid Machine Type. (%s)(EC0103)	マシンタイプが異常です。	
6	Can't Open %s. (EC0105)	ファイルがオープンできません。	システムの状態を確認してください。
7	Can't Write %s. (EC0106)	ファイルにデータが書き込めません。	
8	Can't Read %s. (EC0106)	ファイルからデータが読み出せません。	
9	Can't Link Site Directory. (EC0107)	サイトにリンクができません。	指定ファイルを見直してください。
10	Invalid data. (EC0108)	異常データを検出しました。	
11	Invalid Slot No. (%s)(EC0109)	スロット番号が異常です。	
12	Duplicate Slot No. (%s)(EC0110)	スロット番号が重複しています。	
13	Can't make directory. (%s)(EC0111)	ディレクトリを生成できません。	システムの状態を確認してください。
14	Can't make Sub-directory. (EC0112)	サブディレクトリを生成できません。	
15	Invalid Name. (%s)(EC0113)	異常名称です。	指定ファイルを見直してください。
16	Invalid File. (%s)(EC0115)	異常ファイルです。	指定ファイルを見直してください。 (*1)
17	Generation Data Illegal. (%s)(EC0116)	ジェネレーションのデータに異常があります。	ジェネレーション用データを確認してください。
18	Can't Creat Lock File. (%s)(EC0117)	ロックファイルを生成できません。	システムの状態を確認してください。
19	sysadm Busy. (EC0118)	他で利用者がいます。	再試行してください。
20	Not Super User. (EC0119)	スーパーユーザではありません。	スーパーユーザで再試行してください。
21	Over Max Site Count. (EC0120)	最大サイト数を超過しています。	サイト数を調整してください。 (*2)

(2/2)

No.	エラーメッセージ	意味	ユーザの対応
22	Can't Copy Define File. (EC0121)	定義ファイルのコピーに失敗しました。	システムの状態を確認してください。
23	RPDP Initialize Not Finish. (EC0122)	RPDPの初期化に失敗しました。	再立ち上げ後、再試行してください。
24	Invalid CPU Type. (%s). (EC0124)	CPUタイプ指定が異常です。	指定ファイルを見直してください。
25	Usage:svgen fname	—	—

(\*1) sysadmファイルが異常となった場合、同一ディレクトリにtmpsysadmが存在すればtmpsysadmに生成中のサイトがあることを確認し、存在すればtmpsysadmをsysadmファイル名に変更してください。

tmpsysadmにサイト名称がなく、sysadmファイルが存在する場合は、tmpsysadmを消去してください。

(\*2) 最大サイト個数は、デフォルトで512個です。

&lt;svconf&gt;

(1/4)

No.	エラーメッセージ	意味	ユーザの対応
1	Can't open file. (%s)(EC0203)	ファイルをオープンできません。	システムの状態を確認してください。
2	Invalid Site. (%s)(EC0205)	異常サイトを検出しました。	サイト名称を確認してください。
3	Invalid Slot No. (%d)(EC0207)	slot番号が異常です。	ジェネレーション情報を確認してください。
4	Over MAX ARP Entry. (EC0208)	ARP情報が最大エントリを超えました。	
5	Invalid IP Address. (%s)(EC0209)	IPアドレスが異常です。	
6	Invalid MAC Address. (%s)(EC0210)	MACアドレスが異常です。	
7	Over MAX Route Entry. (EC0211)	Route情報が最大エントリを超えました。	
8	Invalid Gate Way Address. (%s)(EC0212)	GateWayアドレスが異常です。	
9	Over MAX Adapter Entry. (EC0213)	Adapter情報が最大エントリを超えました。	
10	Invalid Station No. (%d)(EC0214)	Station番号が異常です。	
11	Invalid Interface Count. (%d)(EC0215)	Interface個数が異常です。	
12	Invalid UNO. (%s)(EC0216)	UNOが異常です。	
13	Invalid Interface No. (%d)(EC0217)	Interface番号が異常です。	
14	Invalid Subnet Mask. (%s)(EC0218)	サブネットマスクが異常です。	
15	Invalid BroadCast Address. (%s)(EC0219)	ブロードキャストアドレスが異常です。	
16	Over MAX Adapter Entry. (System)(EC0220)	最大Adapterエントリを超えました。	
17	Invalid Trace Mode. (%d)(EC0221)	トレースモードが異常です。	
18	Invalid TCP Port No. (%d)(EC0223)	TCPのポート番号が異常です。	
19	Invalid UDP Port No. (%d)(EC0224)	UDPのポート番号が異常です。	
20	Invalid IP Port No. (%d)(EC0225)	IPのポート番号が異常です。	
21	Invalid ARP Flag. (%d)(EC0226)	ARPフラグが異常です。	
22	Invalid ARP Send Count. (%d)(EC0227)	ARPの転送個数が異常です。	
23	Invalid TCP Window Size. (%d)(EC0228)	TCPのウィンドウサイズが異常です。	
24	Invalid TCP TIMEOUT. (%d)(EC0230)	TCPのタイムアウト時間が異常です。	
25	Invalid UDP Check SUM Flag. (%d)(EC0231)	UDPのSUMフラグが異常です。	
26	Invalid HOP No. (%d)(EC0232)	HOP番号が異常です。	
27	Invalid Remote CPU Control Port No. (%d)(EC0233)	リモートCPUのポート番号が異常です。	
28	Invalid Cyclic Port No. (%d)(EC0234)	CYCLEのポート番号が異常です。	
29	Invalid AI Report Time. (%d)(EC0235)	AIのレポート時間が異常です。	

(2/4)

No.	エラーメッセージ	意味	ユーザの対応
30	Invalid AI Report Count. (%d)(EC0236)	AIのレポート個数が異常です。	ジェネレーション情報を確認してください。
31	Invalid file. (%s)(EC0237)	異常ファイルを検出しました。	
32	Invalid Slot No. (%s)(EC0238)	スロット番号が異常です。	
33	Duplicate Slot No. (%s)(EC0239)	スロット番号が重複しています。	
34	Invalid Adapter Name. (%s)(EC0240)	アダプタ名称が異常です。	
35	Invalid Main Memory Size. (%s)(EC0241)	主メモリサイズが異常です。	
36	Invalid Common Memory Size. (%s)(EC0242)	共有メモリサイズが異常です。	
37	Invalid Duplex Common Memory Size. (%s)(EC0243)	二重化共用メモリサイズが異常です。	
38	Can't Copy Data File. (%s)(EC0244)	データファイルをコピーできませんでした。	
39	Can't Rename Data File. (%s)(EC0245)	データファイルの名称を変更できませんでした。	
40	RPDP Initialize Not Finish. (EC0246)	RPDPの初期化が終了しません。	
41	Invalid UCB No. (%s)(EC0256)	UCB番号が異常です。	
42	Invalid Channel No. (%s)(EC0257)	チャンネル番号が異常です。	
43	Invalid Timeout. (%s)(EC0258)	タイムアウト時間が異常です。	
44	Over MAX Host Entry. (EC0259)	Host数が最大値を超えました。	
45	Invalid Host Name. (%s)(EC0260)	Host名称が異常です。	
46	Over MAX Network Entry. (EC0261)	Network数が最大を超えました。	
47	Invalid Network Name. (%s)(EC0262)	Network名称が異常です。	
48	Over MAX Service Entry. (EC0263)	Service数が最大を超えました。	
49	Invalid Service Name. (%s)(EC0264)	Service名称が異常です。	
50	Invalid Port No. (%s)(EC0265)	ポート番号が異常です。	
51	Invalid Protocol No. (%s)(EC0266)	Rrotocol番号が異常です。	
52	Over MAX IP_route Entry.(EC0267)	P_route数が最大を超えました。	
53	Invalid UNO. (%d)(EC0216)	UNOが異常です。	
54	Can't Write file. (%s)(EC0269)	ファイル書き込みに失敗しました。	
55	Can't Find UNO in Network Information. (%d)(EC0270)	Network情報のUNOが見つかりません。	
56	Can't Get Memory Size. (EC0271)	メモリサイズを取り込めません。	
57	Over MAX Dual Network Entry. (EC0275)	重複するNetwork個数が最大を超えました。	
58	Invalid PATROL. (%d)(EC0278)	PATROL指定が異常です。	
59	Invalid SOFT_TO. (%d)(EC0279)	SOFT_TO時間が異常です。	
60	Invalid RT_BUFSIZE. (%d)(EC0280)	RT_BUFSIZEが異常です。	
61	Over MAX UCB Entry. (EC0273)	UCB個数が最大を超えました。	
62	Over MAX Interface Entry. (EC0276)	Interface個数が最大を超えました。	

No.	エラーメッセージ	意味	ユーザの対応
63	Invalid HI_WATER. (%d)(EC0281)	HI_WATER値が異常です。	ジェネレーション情報を確認してください。
64	Invalid LO_WATER. (%d)(EC0282)	LO_WATER値が異常です。	
65	Invalid CYC_SEND. (%s)(EC0283)	CYC_SEND値が異常です。	
66	Invalid CYC_WATCH. (%s)(EC0284)	CYC_WATCH値が異常です。	
67	Invalid CYC_WATCH2. (%s)(EC0285)	CYC_WATCH2値が異常です。	
68	Invalid RT_NETADDR. (%s)(EC0286)	RT_NETADDRが異常です。	
69	Invalid NET_SRBCNT. (%d)(EC0287)	NET_SRBCNTが異常です。	
70	Not Network Adapter. (Slot:%d)(EC0290)	スロットがNetworkのアダプタではありません。	
71	Not Network Adapter. (UNO:%d)(EC0291)	UCBがNetworkのアダプタではありません。	
72	Can't Find UNO in ucb.u. (UNO:%d)(%s)(EC0292)	ucb.u内にUNO番号が見当たりません。	
73	Generation Data Illegal. (%s)(EC0295)	全体的な情報が異常です。	システムの状態を確認してください。
74	Invalid MBUF_CNT. (%d)(EC0296)	MBUF_CNT値が異常です。	
75	Can't Creat Lock File. (%s)(EC0297)	ロックファイルを生成できませんでした。	再試行してください。
76	Site Busy. (EC0298)	サイトは他で利用されています。	
77	Not Super User. (EC0299)	スーパーユーザではありません。	スーパーユーザで再試行してください。
78	Cannot omit parameter(%s). (EC0701)	パラメータの省略はできません。	ジェネレーション情報を確認してください。
79	Invalid parameter specified(%s). (EC0702)	パラメータ指定に誤りがあります。	
80	Cannot create file(rms_data. c). (EC0703)	ファイル生成ができません。	
81	Systemcall error(%s). (EC0704)	システムコールのエラーです。	
82	Parameter unmatched between definition file and server configuration(%s). (EC0705)	パラメータが定義ファイルとサーバ情報で一致しません。	
83	GAREA size must multiple of 4 (%s).	GAREAのサイズは、4の倍数でなければなりません。	
84	GAREA size too large (%s).	GAREAサイズが大きすぎます。	
85	Total GAREA size too large.	全体GAREAサイズが大きすぎます。	
86	Old map range over on new map(%s). (EC0320)	新たなマップでは、旧マップの利用領域を超えています。	
87	Can not change oswork top in TCB(%s). (EC0321)	TCBのOSワークを変更できませんでした。	
88	Illigal site-information found(%s). (EC0323)	サイト情報が異常です。	

(4/4)

No.	エラーメッセージ	意味	ユーザの対応
89	LANCP count is over. (EC0325)	LANCPの設定情報が多すぎます。	ジェネレーション情報を確認してください。
90	Except CMU adapter not supported(%s). (EC0326)	CMU以外のアダプタはサポートしていません。	
91	Usage:svconf site	—	—
92	Usage:svconf sitename	—	—

(\*) No.92のエラーメッセージは、S-7895-63Pでだけ出力されます。

<svshconf>

No.	エラーメッセージ	意味	ユーザの対応
1	Invalid Site Directory Name. (%s)	サイトディレクトリ名称が異常です。	サイトを含むディレクトリ名称を確認してください。
2	Can't Open %s. (EC0405)	ファイルをオープンできませんでした。	ファイルを確認してください。 (*1)
3	Can't Read %s. (EC0406)	ファイルから読み込むことができませんでした。	ファイルを確認してください。
4	Illegal sitename. (%s)	サイト名称が異常です。	サイト名称を見直してください。
5	Can't Find specified sitename. (%s)	指定サイト名称が見つかりません。	サイト名称を確認してください。
6	Usage:svshconf sitename [-f ofile]	—	—
7	Usage:svshconf sitename [-f fname] (*2)	—	—

(\*1) memory.eファイル時、svconfコマンドを実施したか確認してください。

(\*2) No.7のエラーメッセージは、S-7895-63Pでだけ出力されます。

&lt;svsitecp&gt;

(1/2)

No.	エラーメッセージ	意味	ユーザの対応
1	Usage:svsitecp site1 site2	オプションに誤りがあります。	正しいオプションで起動してください。
2	svsitecp:Can't Define Copy Site Name. (サイト名) (EC0502)	指定されたコピー元サイト名称が定義されていません。	サイト名を見直してください。
3	svsitecp:Invalid Copy Site Number. (EC0504)	指定されたコピー先のサイト数と、コピー元のサイト数が一致しません。	コピー先サイト名称の数をコピー元サイト数分指定し、再起動してください。
4	svsitecp:Can't Updata sysadm. (EC0505)	PCsシステム管理ファイルを更新できませんでした。	サイトが1つも構築されていないか、sysadmファイルが消去された場合に発生します。サイト構築済みの場合は、システム管理者に連絡してください。 (* )
5	svsitecp:Invalid Copy Site Name. (サイト名) (EC0506)	指定されたコピー先サイト名称に誤りがあります。	サイト名を見直してください。
6	svsitecp:Can't Copy.(EC0508)	コピー処理ができませんでした。	コピー先サイトディレクトリが存在するディスクの空き容量が不足していると考えられます。空き容量を増やして再起動してください。
7	svsitecp:Can't Site Lock.(EC0509)	対象サイトのロックができませんでした。	対象サイトに対しRPDPのコマンドが実行されています。 RPDPコマンドを終了後、再起動してください。
8	svsitecp:File Access Error. (ファイル名) (EC0511)	ファイルに対するアクセスができませんでした。	ディスクの空き容量が不足していると考えられます。空き容量を増やして再起動してください。
9	svsitecp:Can't Open ファイル名.(EC0510)	ファイルをオープンできませんでした。	ファイルやディレクトリのセキュリティなどの状態を確認してください。
10	svsitecp:Can't Create Lock File. (ファイル名) (EC0512)	ロックファイルを作成できませんでした。	
11	svsitecp:sysadm Busy. (EC0513)	他ジェネレーションコマンドによりsysadmファイルを使用中です。	他ジェネレーションコマンド終了後に再起動してください。

No.	エラーメッセージ	意味	ユーザの対応
12	svsitecp:Not Super User. (EC0514)	administratorではありません。	administratorにて再起動してください。
13	svsitecp:Over Max Site Count. (EC0515)	PC上にて構築できる最大サイト数を超過しています。	%windir%\renix\usr¥s10v¥etc¥max_sites ファイルに定義された最大サイト数を増やすか不要サイトを削除した後、再起動してください。
14	svsitecp:Invalid File. (ファイル名) (EC0516)	ファイルに異常があります。	システム管理者に連絡してください。 (*)
15	svsitecp:Specified Name is Already Used (サイト名) . (EC0521)	指定されたサイト名はすでに使われています。	サイト名を見直してください。
16	svsitecp:Can't Read System Management File. (EC0522)	PCsシステム管理ファイルを読み込めませんでした。	サイトが1つも構築されていないかsysadmファイルが消去された場合に発生します。サイト構築済みの場合は、システム管理者に連絡してください。 (*)

(\*) 対処方法が「システム管理者に連絡してください。」となっているエラーに関しては、調査用のデータを収集してください。収集するデータファイルは、下記となります。

・ %windir%\renix¥etc¥log¥s10v¥1\_RPDP\_CTL, Rpdp\_ctl

## &lt;svsitedel&gt;

No.	エラーメッセージ	意味	ユーザの対応
1	Usage:svsitedel site	オプションに誤りがあります。	正しいオプションで起動してください。
2	svsitedel:Can't Define Delete Directory Name. (サイト名) (EC0602)	指定されたサイト名がsysadmに定義されていません。	サイト名を見直してください。
3	svsitedel:Can't Write ファイル名. (EC0605)	ファイルに書き込めませんでした。	ディスクの空き容量が不足していると考えられます。空き容量を増やして再起動してください。
4	svsitedel:Can't Delete Data in sysadm. (EC0608)	sysadmファイル内のデータを削除できませんでした。	
5	svsitedel:Invalid File ファイル名. (EC0606)	ファイル構成が異常です。	ファイル構成を正常に修正し、再起動してください。
6	svsitedel:Not Super User. (EC0607)	administratorではありません。	administratorにて再起動してください。
7	svsitedel:Can't Open ファイル名. (EC0603)	ファイルをオープンできませんでした。	ファイルやディレクトリのセキュリティなどの状態を確認してください。
8	svsitedel:Can't Creat Lock File. (ファイル名) (EC0611)	ロックファイルを作成できませんでした。	
9	svsitedel:sysadm Busy. (EC0612)	他ジェネレーションコマンドによりsysadmファイルを使用中です。	他ジェネレーションコマンド終了後に再起動してください。
10	svsitedel:Can't Delete Directory. (ディレクトリ名) (EC0613)	ディレクトリの削除に失敗しました。	ディレクトリへのアクセスを終了した後、再起動してください。 (*1)
11	svsitedel:Can't Site Lock. (EC0615)	対象サイトのロックができませんでした。	対象サイトに対しRPDPのコマンドが実行されています。RPDPコマンドを終了後、再起動してください。
12	Usage:svsitedel site [-d] (*2)	オプションに誤りがあります。	正しいオプションで起動してください。

(\*1) エクスプローラ等での参照を止め、svsitedelコマンドを“-d”オプション付きで再実行してください。

(\*2) No.12のエラーメッセージは、S-7895-63Pでだけ出力されます。

(2) アロケータ、ローダ、ビルダコマンドのエラーメッセージ

● シグナルの扱い

アロケータ、ローダ、ビルダおよびマップ表示コマンド処理中にシグナルを受信したときの扱いを表A-2に示します。

表A-2 シグナルの扱い

シグナル	シグナル受信時の動作
SIGINT	処理を中断します。
SIGQUIT	

● エラーメッセージ

アロケータ、ローダ、ビルダおよびマップ表示コマンドが出力するエラーメッセージを表A-3に示します。各コマンドは異常検出後、これらのメッセージを出力し終了します。

表A-3 エラーメッセージ

(1/8)

エラーコード	エラーメッセージ	意味	ユーザの対応
システム管理者による復旧が必要な内部エラー			
1001-1	Abnormal allocator management table(%s)	アロケータ管理テーブルが異常です。	システム管理者に連絡してください。 (*)
リソース不足に関連するエラー			
1002-4	Not enough physical memory allocated	物理メモリの容量が不足しています。	不足対象に応じて再度システムジェネレーションを実行してください。
5	Default garea is not defined	デフォルトGAREAが未定義です。	
6	Not enough area allocated(%s)	GLB領域の容量が不足しています。	
7	Not enough secondary storage area allocated(%s)	補助メモリの容量が不足しています。	
8	No task number available	タスク番号の空きがありません。	
10	No free table to make new entry(%s)	アロケータ管理テーブルの空きがありません。	
11	Too small system constants to resize management table(%s)	システム定数が小さ過ぎます。	
13	Cannot get RSSITE(%s)	環境変数のRSSITEが取り込めません。	
15	Please set environment variable(%s)	環境変数が設定されていません。	

(2/8)

エラーコード	エラーメッセージ	意味	ユーザの対応
システムリソース不足に関連するエラー			
1003-1	Memory allocation error (malloc,%s)	mallocでメモリが確保できません。	システム管理者に連絡してください。 (*)
3	Cannot create temporaries: %s	テンポラリファイルが作成できません。	
リトライにより復旧可能なエラー			
1004-1	Allocator management table is busy	アロケータ管理テーブルが他コマンドにより使用中です。	再試行してください。
原因を特定できるシステムコールエラー			
1005-1	Cannot open %s (%s)	ファイルを開けません。	システム管理者に連絡してください。 (*)
2	Cannot read %s (%s)	ファイルを読み出せません。	
3	Cannot write %s (%s)	ファイルに書き込めません。	
4	Can't exec %s	内部コマンドが実行できません。	
5	Lost %s - No child process!	子プロセスが消えました。	
6	Macro error (dtaskhr,%d)	dtaskhrが実行できません。	
7	Not dormant status	タスクがdormant状態ではありません(内部矛盾が発生しました)。	
8	Cannot unlink %s (Text file busy)	プログラムの削除に失敗しました。	
9	Cannot stat %s (%s)	ファイルのステータス情報が取り込めません。	
10	Cannot lseek %s (%s)	ファイルポインタをシークできません。	
原因を特定できないシステムコールエラー			
1006-1	systemcall error (%s,errno=%d)	システムコールのエラーが発生しました。	システム管理者に連絡してください。 (*)
2	コマンド名 : WIN32API error (API名称, EC=エラーコード)	API名称の関数でエラーコードのエラーが発生しました。	
パラメータ不足			
1007-3	Not enough parameter	引数が足りません。	引数を見直してください。

エラーコード	エラーメッセージ	意味	ユーザの対応
入力パラメータ範囲異常関連のエラー			
2001-2	Align number is out of range	アライン数値が範囲外です。	入力可能データを 確認後、再試行して ください。
3	Task number is out of range (1 to %d)	タスク番号が範囲外です。	
5	Priority level is out of range (%d to %d)	ユーザタスク用実行レベルが範囲外 です。	
6	Priority level for system is out of range (%d to %d)	システムタスク用実行レベルが範囲外 です。	
9	Bad align type	アライン種別が間違っています。	
10	Illegal point number (%d)	組み込みサブルーチン用ポイント番号 が間違っています。	
11	Entry number is out of range (1 to %d)	組み込みサブルーチン用エントリ番号 が範囲外です。	
12	Specified index number is out of range (1 to %d)	指定の番号が範囲外です。	
13	Invalid name (%s)	名称が間違っています。	
18	Illegal point name (%s)	組み込みサブルーチン用ポイント名称 が間違っています。	
19	Numeric value is out of range	指定の数値が範囲外です。	
23	Limit size for stack is out of range (0 to 2097152)	指定のスタックサイズが範囲外です。	
24	User task number is out of range (1 to 224)	指定のユーザタスク番号が範囲外で す。	
25	Specified system index number is out of range (1 to %d)	指定のシステム用番号が範囲外です。	
26	Specified number with -r is out of range	-rオプションで指定した番号が範囲外で す。	
27	Loading data is empty	ローディングデータがありません。	
28	Number of user task is over (タスク数)	ユーザタスク数がシステム定数を超え ました。	
33	ULSUB stack size (%d) is out of range (0 to 512)	組み込みサブプログラムのスタックサ イズが512バイトを超えています。	スタックサイズを見 直してください。
35	Program/IRSUB stack size (%d) is out of range (0 to 8388608)	タスクのスタックサイズが8MBを超え ています。	
入力パラメータ未定義関連のエラー			
2002-1	Specified name is undefined (%s)	指定の名称は未定義です。	入力可能データを確 認後、再試行して ください。
4	Specified point number in the entry number is empty	指定の組み込みサブルーチン用ポイン ト番号は未登録です。	
5	Specified IRSUB is not built (%s)	指定のIRSUBはビルドされていま せん。	
7	Specified IRSUB is already built (%s)	指定のIRSUBはビルド済みです。	
8	Specified number is undefined	指定の番号は未定義です。	
10	%s is undefined	オブジェクト内の名称が未定義です。	

(4/8)

エラーコード	エラーメッセージ	意味	ユーザの対応
入力パラメータ未定義関連のエラー (続き)			
11	Loading data is empty	ローディングするデータがありません。	オブジェクトファイル指定を見直してください。
13	Area (%s) kind is wrong	エリア種別が誤っています。	指定エリアを見直してください。
14	Can not load data in GLBW (%s)	初期値なしGLBに初期値をロードすることはできません。	
入力パラメータ二重定義関連のエラー			
2003-1	Specified name is already defined (%s)	指定名称は登録済みです。	入力可能データを確認後、再試行してください。
5	Task number is already defined	指定のタスク番号は登録済みです。	
7	Point number is already defined	指定の組み込みサブルーチン用ポイント番号は登録済みです。	
8	Specified IRSUB number is already defined	指定したIRSUB番号は使用済みです。	IRSUB番号を変えて再試行してください。
13	Unmatched reserved index number	指定の番号は不一致です。	入力可能データを確認後、再試行してください。
15	PN=%s is already defined	プログラム管理番号が重なっています。	
19	Specified number is already defined	指定の番号は登録済みです。	
21	Can not specify -s or -a with SAREA(%s)	細分割領域に-sまたは-aオプションは指定できません。	
22	Specified pgmname is already defined as TASK(%s)	指定のプログラム名称はタスクとして登録済みです。	
23	PN=%d is already loaded for single task	プログラム管理番号は定義済みです。	プログラム管理番号を変えて再試行してください。
入力パラメータ属性不一致関連のエラー			
2004-1	Unmatched owner type	所有者タイプが不一致です。	入力可能データを確認後、再試行してください。
2	Illegal user type (%s)	使用者タイプが不一致です。	
3	Specified area is not global (%s)	エリアの種別がGLB以外です。	
5	Illegal program type	プログラム種別が不一致です。	
8	Unmatched entry number	組み込みサブルーチン用エントリ番号が不一致です。	
10	Area type is not GLBI	エリア種別が初期値ありGLB以外です。	
11	Multi task attribute error	マルチタスク属性が不一致です。	
16	Unmatched entry type	組み込みサブルーチン用エントリセット番号が不一致です。	
17	Specified name is defined as GLB(%s)	指定の名称はGLBとして定義されています。	

エラーコード	エラーメッセージ	意味	ユーザの対応
入力パラメータ属性不一致関連のエラー (続き)			
19	Specified name is defined as VAL (%s)	指定の名称はVALとして定義されています。	入力可能データを確認後、再試行してください。
21	4096 aline error (%s)	指定アドレスが4096バイト境界ではありません。	
22	Physical address error (%s)	不当な論理アドレスを指定しています。	
23	Area (%s) kind is wrong	エリアの属性が誤っています。	
24	Loading data is too large (sname=%s)	データサイズがエリアのサイズを超えています。	
25	Inconsistent object was mixed	ライブラリ指定で-lsh4nbmzzと-lsh4nbmdn両方を指定しています。	
操作誤り関連のエラー			
2005-1	Cannot delete area which is already used	タスクまたはサブプログラムが登録済みのためエリアを削除できません。	svdloadしてから削除してください。
2	Cannot delete program which is registered as task	タスクとして登録済みのため削除できません。	svdtaskしてから削除してください。
3	Cannot delete built subprogram (%s)	ビルド済みのため削除できません。	svdbuildしてから削除してください。
6	Cannot delete defined %s (%s)	GLBまたはVALとして登録済みのため削除できません。	svdlsまたはsvdlvして削除ください。
7	Specified name (%s) is referenced by PROG or SUB	指定のリソースはプログラムまたはサブプログラムから参照されています。	参照しているプログラムまたはサブプログラムを削除してから再試行してください。
svloadコマンドでのパラメータ指定方法誤り関連のエラー			
2006-1	Invalid subargument: -W%s	使用できないサブ引数を指定しました。	入力可能データを確認後、再試行してください。
5	Too few arguments	引数の個数が不足しています。	
8	Missing operand (%s)	オペランドが不足しています。	
9	Bad option (%s)	使用できないオプションを指定しました。	
10	Invalid name (%s)	名称が間違っています。	
svloadコマンドの内部コマンドのエラー			
2008-1	Error in %s ; Status %d	内部コマンド (%s) でエラーが発生しました。	指定したライブラリが正しいか、およびオブジェクトファイルまたはそのソースを見直し後、再試行してください。
2	Fatal error in %s ; Status %d	内部コマンド (%s) で致命的なエラーが発生しました。	システム管理者に連絡してください。 (* )
アロケータ管理テーブルの異常			
200-1	RMphase, (0x%02x)	処理フェーズが異常です。	システム管理者に連絡してください。 (* )

(6/8)

エラーコード	エラーメッセージ	意味	ユーザの対応
入力データ誤りの関連のエラー			
—	Argument list too long	引数の数が多すぎます。	入力可能データを確認後、再試行してください。
—	Argument data too large	引数のデータ値が大きすぎます。	
—	File open error	指定ファイルが開けません。	
—	Illegal format in ファイル名 line 行	指定ファイルのline行目のフォーマットに誤りがあります。	
—	Illegal format of name	指定名称フォーマットに誤りがあります。	
—	Illegal format of numeric value	数値データの指定に誤りがあります。	
—	Illegal format of task name	タスク名称の指定フォーマットに誤りがあります。	
—	Illegal operand	オペランドの指定に誤りがあります。	
—	Program text is empty	プログラムテキストサイズが0です。	
—	%s is different from subprogram top name	サブプログラム名称が誤っています。	
—	%s is referred from system type	システムからユーザを参照しています。	
—	%s is referred from user type	ユーザからシステムを参照しています。	
—	Illegal option	許されないオプションを指定しました。	
—	Illegal option combination	オプションの組み合わせに誤りがあります。	
—	Missing option parameter	オプションパラメータの指定に誤りがあります。	
—	Numeric value is out of range	指定された数値が許容範囲外です。	
—	Parameter error	パラメータの指定誤りです。	
—	Specified number is undefined (指定name)	svmapコマンドでnameおよび-nオプションを指定したとき、指定name (エントリ番号) は未定義でした。	
—	Specified number is illegal (指定name)	svmapコマンドでnameおよび-nオプションを指定したとき、name指定 (エントリ番号) に誤りがありました。	
—	Specified name is undefined (指定name)	svmapコマンドでnameを指定したとき、指定nameは未定義でした。	
—	Task number error	タスク番号の指定方法に誤りがあります。	
—	Bad file name	svadmコマンドでオペレーション結果出力先のファイル名称が256文字以上です。	
—	Bad site name	svadmコマンドで指定サイト名称が14文字以上です。	

エラーコード	エラーメッセージ	意味	ユーザの対応
入力データ誤りの関連のエラー			
—	Parameter is too long	引数の数が多すぎます。	入力可能データを確認後、再試行してください。
—	Illegal parameter	引数のパラメータに誤りがあります。	
—	Sitename max length is 14 character (-u)	サイト名が14文字を超えています。	2～128の値を指定してください。
—	Multi task count is out of range (2 to 128)	マルチタスク数に誤りがあります。	
—	Can not get RSSITE	環境変数を取り込めません。	RSSITE環境変数を指定してください。
—	No such site (%s)	指定サイトはありません。	指定サイトを見直してください。
—	Some system constants are not defined	指定定数が、システム定数にありません。	指定定数を見直してください。
—	-w option argument is not 8 byte align	スタックサイズに8の倍数以外が指定されています。	指定サイズを見直してください。
—	Total stack size(%d) is too small	確保するスタックサイズがプログラムで使用するスタックサイズよりも小さな値です。	スタックサイズを見直してください。
—	-C option argument is not %d byte align	-Cオプションで指定された値が適切な値ではありません。	プログラムの場合は4096の倍数を、サブプログラムの場合は32の倍数を指定してください。
—	Bad realtime environment	環境変数が異常です。	環境変数を見直してください。
—	Program text is empty	プログラムのテキストサイズが0です。	指定オブジェクトファイルを見直してください。
—	%s is different from subprogram top name	サブプログラム名称が誤っています。	
—	.rodata cannot locate GLB area	const宣言されたデータは、ロードできません。	初期値データプログラムのconst宣言を見直してください。
—	%s is not defined (Sarea)	オブジェクトに未定義GLB/CM/DCMの初期値データが含まれます。	初期値データを見直してください。
—	Program has BSS area	マルチタスク・IRSUBがBSSエリアを持っています。	プログラムを確認してください。
—	Stack size (%s) = %d (%d) byte [MAX refered (%s) size %d byte] Err	累積スタックサイズが指定スタックサイズを超えました。	指定スタックサイズを見直してください。
—	Can not get site information	サイト情報が取り込めません。	サイト定義内容を確認のうえ、再試行してください。

(8/8)

エラーコード	エラーメッセージ	意味	ユーザの対応
実行環境異常関連のエラー			
—	Please set RSSITE	環境変数RSSITEが設定されていません。	RSSITEを設定後、再試行してください。
—	Unknown RSUTYP	環境変数RSUTYPに指定外のパラメータを設定しています。	RSUTYPに‘s’または‘u’を設定してください。
コマンド内部のエラー			
—	abnormal error (error=エラー番号)	内部エラーを検出しました。	システム管理者に連絡してください。 (*)
—	cannot perform malloc	mallocまたはreallocによる作業エリアの確保に失敗しました。	
—	Internal error (timeout detected)	通信タイムアウトを検出しました。	
—	Internal error (no valid data)	通信データに異常が発生しました。	
—	Internal error (select failed)	selectシステムコールエラーが発生しました。	
—	select error (errno=エラー番号)	selectシステムコールエラーが発生しました。	
—	cannot get filesize	svmapコマンドでバックアップファイルの空きエリア管理ファイルのサイズ取り込みに失敗しました。	
—	cannot open %s	svmapコマンドでアロケータ管理テーブルファイルのオープンに失敗しました。	
—	bad user name %s	パスワードファイルエントリにユーザ名が登録されていません。	
—	Specified site is undefined	指定サイトが存在しません。	

(\*) 対処方法が「システム管理者に連絡してください。」となっているエラーに関しては、調査用のデータを収集してください。収集するデータファイルは、下記となります。

・ %windir%\¥renix¥etc¥log¥s10v¥1\_RPDP\_CTL,Rpdp\_ctl

## (3) svdebugエラーメッセージ一覧

(1/5)

No.	エラーメッセージ	意味	ユーザの対応
1	File "ファイル名" already exists	指定したファイルはすでに存在しています。	正しいファイル名を指定してください。
2	Site "サイト名" not found	指定したサイトがありません。	サイト名を見直してください。
3	Cannot open "ファイル名"	ファイルのオープンに失敗しました。	正しいファイル名を指定してください。
4	No filename given for -i/-o/-r option	ファイル名の指定がありません。	ファイル名を指定してください。
5	No sitename given for -u option	サイト名の指定がありません。	サブコマンドを指定してください。
6	Task No error	タスク番号の指定が誤っています。	タスク番号を見直してください。
7	Task name error	タスク名の指定が誤っています。	タスク名を見直してください。
8	Factor error	起動要因の指定が誤っています。	起動要因を見直してください。
9	Cannot Specify RPC-server task	RPCサーバのタスクは指定できません。	タスク番号、タスク名を見直してください。
10	Unknown sub command	解釈不能なサブコマンド名が指定されています。	サブコマンドを見直してください。
11	Name error	解釈不能な名称が指定されています。	名称を見直してください。
12	Option error	解釈不能なオプションが指定されています。	オプションを見直してください。
13	Storage error	解釈不能な記憶装置が指定されています。	storage指定を見直してください。
14	Invalid address set	アクセス不可能なアドレスを指定しています。	アドレスを見直してください。
15	Misformed patch data	変更となるデータの設定に誤りがあります。	8進、10進、16進、実数のいずれかで入力してください。
16	Unknown RSSITE	RSSITE環境変数が設定されていません。	RSSITE環境変数を設定してください。
17	Unknown RSUTYP	ユーザモードでのシステムリソースへのアクセスはできません。	RSU TYP環境変数に 's' または 'u' を設定してください。
18	Break point already used by another process	別のデバッグプロセスにてブレークポイントを使用中です。	他のユーザの終了を待って、使用してください。
19	Input error	入力書式に誤りがあります。	入力書式を見直してください。
20	Type or length error	md、sdのオプション指定に誤りがあります。	オプションを見直してください。
21	RPDP library error (ライブラリ名:エラーコード)	RPDPのライブラリでエラーが発生しました。	システム管理者に連絡してください。
22	Allocator management table busy	アロケータ管理テーブルがbusy状態です。	コマンドを再実行してください。
23	Unmatch resource status	開発系マシンとコントローラ間でリソースの状態が不一致です。	名称指定を見直してください。
24	Id NO error	tmサブコマンドのidの指定が誤っています。	idを見直してください。

(2/5)

No.	エラーメッセージ	意味	ユーザの対応
25	Time error	tmサブコマンドのtの指定が誤っています。	tを見直してください。
26	Cycle time error	tmサブコマンドのcyctの指定が誤っています。	cyctを見直してください。
27	Initial/check data error	si、spサブコマンドの初期化/チェックパターンの指定が誤っています。	初期化/チェックパターンの指定を見直してください。
28	Addr error	br、rb、asサブコマンドのアドレスの指定が誤っています。	アドレスを見直してください。
29	Break point is used	ブレークポイントを使用中です。	ブレークポイントを解除してからデバッグを終了してください。
30	Bit data error	bs、bgサブコマンドのビットの指定が誤っています。	ビットの指定を見直してください。
31	Sht sub command is already executed by another process	別のデバッグプロセスにてshtサブコマンドを実行中です。	他のユーザの終了を待って、使用してください。
32	Task no error (NO.2100-01)	タスク番号の指定が誤っています。	タスク番号を見直してください。
33	Task name error (NO.2100-02)	タスク名の指定が誤っています。	タスク名を見直してください。
34	Factor error (NO.2100-03)	起動要因の指定が誤っています。	起動要因を見直してください。
35	Cannot Specify RPC-server task (NO.2100-04)	RPCサーバのタスクは指定できません。	タスク番号、タスク名を見直してください。
36	Specified task is dormant (NO.2100-05)	指定されたタスクは、DORMANT状態です。	タスクの状態を確認し、再試行してください。
37	Specified task is not dormant (NO.2100-06)	指定されたタスクは、DORMANT状態ではありません。	
38	Specified task is already suspend (NO.2100-07)	指定されたタスクは、すでに実行抑止状態です。	
39	Specified task is not suspend (NO.2100-08)	指定されたタスクは、すでに実行抑止状態ではありません。	タスク番号、タスク名を見直してください。
40	Specified task is not registered (NO.2100-09)	指定されたタスクは、未登録です。	
41	Backup file access error (NO.2100-10)	バックアップファイルのアクセスに失敗しました。	システム管理者に連絡してください。(*1)
42	Unmatched RSUTYP (NO.2100-11)	ユーザモードでのシステムリソースへのアクセスはできません。	システムモードでアクセスしてください。
43	Unmatch resource status (NO.2100-12)	開発系マシン、コントローラ間でリソースの状態が不一致です。	名称指定を見直してください。
44	Specified task is undefined (NO.2100-13)	指定したタスク名が未定義です。	タスクの指定を見直してください。
45	User Task not running (NO.2100-14)	ユーザタスクがRUNNING状態ではありません。	ユーザタスクをRUNNING状態にしてください。
46	Task is not dormant (tn=%d) (NO.2100-15)	タスクがDORMANT状態ではありません。	タスクをDORMANT状態にして再試行してください。
47	Invalid address error (NO.2100-16)	アクセス不可能なアドレスをアクセスしようとしてしました。	アドレスを見直してください。
48	Invalid address set (NO.2100-17)	アクセス不可能なアドレスを指定しています。	

No.	エラーメッセージ	意味	ユーザの対応
49	Cannot open "ファイル名" (NO.2100-18)	ファイルのオープンに失敗しました。	正しいファイル名を指定してください。
50	Processor connection table is full (NO.2100-19)	プロセス間連絡テーブルが満杯です。	他ユーザの使用終了を待ち、再試行してください。
51	Specified task is not idle (NO.2100-20)	指定タスクがIDLE状態ではありません。	タスクの状態を確認し、再試行してください。
52	Timer event is not registered (NO.2100-21)	タイマイベントが登録されていません。	指定タスクを見直してください。
53	Time error (NO.2100-22)	tmサブコマンドのtの指定が誤っています。	tを見直してください。
54	Cycle time error (NO.2100-23)	tmサブコマンドのcyctの指定が誤っています。	cyctを見直してください。
55	Cannot get system constant (NO.2100-24)	システム定数の取り出しに失敗しました。	システム管理者に連絡してください。 (*1)
56	Ent error (NO.2100-25)	ldサブコマンドのentの指定が誤っています。	entの指定を見直してください。
57	Irsb No error (NO.2100-26)	間接リンクサブルーチン番号が誤っています。	間接リンクサブルーチン番号を見直してください。
58	Irglobal No error (NO.2100-27)	間接リンクグローバル番号の指定が誤っています。	間接リンクグローバル番号を見直してください。
59	Task suspend failed (NO.2100-28)	taサブコマンドでタスクの実行抑止に失敗しました。	タスクの状態を確認し、再試行してください。
60	Point error (NO.2100-29)	ldサブコマンドのポイントの指定が誤っています。	pointを見直してください。
61	Cannot get register information (NO.2100-30)	ta、rrサブコマンドでレジスタの取り出しに失敗しました。	システム管理者に連絡してください。 (*1)
62	Specified name (名称) is undefined (NO.2100-31)	指定された名称が未定義です。	名称指定を見直してください。
63	Cannot register timer event in TRB (NO.2100-32)	タイマイベントの登録に失敗しました。	システム管理者に連絡してください。 (*1)
64	File "ファイル名" already exists (NO.2100-33)	ファイルは、すでに存在しています。	ファイル名の指定を見直してください。
65	File "ファイル名" create error (NO.2100-34)	ファイルの作成に失敗しました。	
66	Cannot save "ファイル名" (NO.2100-35)	ファイルの保存に失敗しました。	
67	File "ファイル名" read error (NO.2100-36)	ファイルの読み出しに失敗しました。	
68	File "ファイル名" format error (NO.2100-37)	ld、cmサブコマンドで指定したファイルのフォーマットが誤っています。	
69	Pname "ファイル名" not found (NO.2100-38)	プログラム名が見つかりません。	プログラム名を見直してください。
70	Must specify address in text space (NO.2100-39)	text空間内のアドレスを指定してください。	アドレス指定を見直してください。
71	Specified address is already set (NO.2100-40)	指定アドレスは、すでに設定済みです。	
72	Must specify break point address (NO.2100-41)	ブレークポイントのアドレスを指定してください。	

(4/5)

No.	エラーメッセージ	意味	ユーザの対応
73	Cannot get TCB (NO.2100-42)	taサブコマンドでTCBの取り出しに失敗しました。	システム管理者に連絡してください。 (*1)
74	Cannot set break point beyond the max (NO.2100-43)	すでにブレークポイントの最大設定可能数分 (5つ) を指定しています。	ブレークポイントを解除してから再試行してください。
75	Cannot use ld sub command after RSSRCV set (NO.2100-44)	RSSRCVの設定後にldサブコマンドは使用できません。	svrplコマンドにて一括ダウンロードしてください。
76	Inconsistency detected %s (NO.2100-46)	アロケータ管理テーブルに不整合があります。	システム管理者に連絡してください。 (*1)
77	Specified area is not defined for glb (NO.2100-50)	指定した分割領域はグローバルのエリアではありません。	指定分割領域名を見直してください。
78	Specified pgm number is out of range (1 to 255) (NO.2100-52)	指定プログラム番号は、範囲外です。	指定パラメータを見直してください。
79	ADT channel is already set (NO.2100-56)	ADTはすでに設定されています。	ADTを解除してから実行してください。
80	Illegal break point address (laddr=論理アドレス) (NO.2100-58)	ブレークポイントに設定されている論理アドレスは、プログラムとして登録されているアドレスではありません。	CPUをリスタートし、ブレークポイントの設定を解除してください。
81	Specified raddr is not break point address (raddr=相対アドレス) (NO.2100-59)	指定したraddrはブレークポイントが設定されているアドレスではありません。	指定アドレスを見直してください。
82	Break task is not found (NO.2100-60)	ブレーク中のタスクが見つかりませんでした。	ブレークポイントの設定を確認してください。
83	Specified name (プログラム名称) is used break point (NO.2100-61)	ldサブコマンドで指定したプログラム名称は、ブレークポイントが設定されています。	ブレークポイントを解除してから再試行してください。
84	Specified area is not initialize data area (NO.2100-62)	指定したアドレスはバックアップファイルの存在しない領域を含んでいます。	指定アドレスを見直してください。
85	Specified address is not initialize data area (NO.2100-63)	指定したエリアはバックアップファイルの存在しないエリアです。	指定エリアを見直してください。
86	Specified name (%s) is not GLB area (NO.2100-66)	指定した名称は、GLBエリアではありません。	GLBエリアを指定してください。
87	Communication error (catch signal) (NO.2101-01)	シグナルを受信しました。	システム管理者に連絡してください。 (*1)
88	Communication error (connection timeout) (NO.2101-02)	タイムアウトが発生しました。	
89	Communication error (connection refused) (NO.2101-03)	RPCサーバが不在です。	
90	Communication error (connection cut) (NO.2101-04)	RPCサーバが切断されています。	
91	Communication error (connection reset) (NO.2101-05)	コネクションがリセットされています。	
92	Communication error (server closed) (NO.2101-06)	RPCサーバがクローズしています。	

No.	エラーメッセージ	意味	ユーザの対応	
93	Communication error (port busy) (NO.2101-07)	回線ポートがbusy状態です。	他ユーザの使用終了を待ち、再実行してください。 システム管理者に連絡してください。 (*1)	
94	Communication error (bad socket specified) (NO.2101-08)	指定ソケットが異常です。		
95	Communication error (socket creat error) (NO.2101-09)	ソケットの生成に失敗しました。		
96	Communication error (no buffer) (NO.2101-10)	メモリ確保に失敗しました。		
97	Communication error (network not reached) (NO.2101-11)	ネットワークが未接続状態です。		
98	Communication error (network down) (NO.2101-12)	ネットワーク接続インタフェースがダウンしています。		
99	Communication error (port No error) (NO.2101-13)	ポート番号の取り込みに失敗しました。		
100	Communication error (IP address error) (NO.2101-14)	IPアドレスの取り込みに失敗しました。		
101	Communication error (memory attach failed) (NO.2101-15)	共用メモリのアタッチに失敗しました。		
102	Communication error (trace file cannot open) (NO.2101-16)	トレースファイルのオープンに失敗しました。		
103	Communication error (trace file cannot copy) (NO.2101-17)	トレースファイルのコピーに失敗しました。		
104	Communication error (fatal error) (NO.2101-18)	致命的エラーが発生しました。		
105	Communication error (ライブラリ名:エラーコード) (NO.2101-19)	RPL/RRBライブラリでエラーが発生しました。		
106	Communication error (inter PU communication time out) (NO.2101-20)	内部タイムアウトエラーが発生しました。		
107	Communication error (rc=%d) (NO.2101-21)	RPCライブラリエラーが発生しました。		システム管理者に連絡してください。 (*1) (*2)

(\*1) 対処方法が「システム管理者に連絡してください。」となっているエラーに関しては、調査用のデータを収集してください。収集するデータファイルは、以下のファイルです。

- ・ %windir%\¥renix¥etc¥log¥s10vc¥RDPDP\_CTL
- ・ %windir%\¥renix¥etc¥log¥s10v¥1\_RDPDP\_CTL

(\*2) Communication error発生時は、下記を参照してください。

<Communication errorエラーコードの意味>

- |                              |                                |
|------------------------------|--------------------------------|
| 0x11 : ソケット異常                | 0x04 : フレーム作成用メモリ確保失敗          |
| 0x12 : IPアドレス異常              | 0x05 : データ送信失敗                 |
| 0x14 : 格納エリアアドレス異常 (0指定、dta) | 0x06 : レスポンス受信待ち異常             |
| 0x15 : 格納エリアアドレス異常 (0指定、wka) | 0x07 : レスポンス未受信でリトライオーバー       |
| 0x16 : サイズ異常 (0以下または16KB以上)  | 0x08 : データ受信失敗                 |
| 0x17 : サイズ異常 (非ロングワードサイズ)    | 0x18 : 格納エリアアドレス異常 (0指定、dmaia) |
| 0x03 : 相手アダプタ種別異常            | 0x19 : 格納エリアアドレス異常 (0指定、reta)  |

## (4) svrplコマンドエラーメッセージ一覧

No.	エラーメッセージ	意味	ユーザの対応
1	No sitename given for -u option	サイト名称が指定されていません。	サイト名称を指定してください。
2	No unitname given for -U option	ユニット名称が指定されていません。	ユニット名称を指定してください。
3	unknown RSSITE	指定サイト名称がありません。	指定サイトを確認してください。
4	Site=%s not found		
5	Unit=%s not found	指定ユニット名称がありません。	指定ユニットを確認してください。
6	%s cannot open	ファイルがオープンできません。	ファイルが正常か確認してください。
7	%s file access error	ファイルに対するアクセスができません。	
8	Internal error (%s)	内部エラーが発生しました。	再試行してください。
9	download file (%s) not found	ダウンロードするバックアップファイルが見つかりません。	環境を確認してください。
10	site (%s) lock busy	サイトは、他の処理で使われています。	再試行してください。
11	site (%s) lock error		
12	communication error (%s,RC=0x%x,0x%x)	通信エラーが発生しました。	RCコードを基に原因を確認してください。 (*1)
13	communication error (%s,RC=0x%x)		
14	site (%s) allocator management tables modify error	アロケータ管理テーブル更新でエラーが発生しました。	svmkrestblコマンドでアロケータ管理テーブルを修復してください。
15	IP ADDRESS SET ERROR (RC=0x%x)	IPアドレスを設定時にエラーが発生しました。	RCコードを基に原因を確認してください。
16	%s (slot=%d) NON EXIST	スロットが存在しません。	システムジェネレーション情報を確認してください。
17	File mapping error (%s)	RPDP用リソーステーブルがセッティングできませんでした。	再試行してください。
18	site (%s) unlock error	サイトの占有解除でエラーが発生しました。	
19	Can not specified NP site(%s)	指定サイトは、NPのサイトです。	CPのサイトを指定してください。
20	Usage:svrplhr [-u site {-s}] [-r] [-time   -notime]	—	—

(\*1) Communication error発生時は、下記を参照してください。

<Communication errorエラーコードの意味>

0x11 : ソケット異常	0x04 : フレーム作作用メモリ確保失敗
0x12 : IPアドレス異常	0x05 : データ送信失敗
0x14 : 格納エリアアドレス異常 (0指定、dta)	0x06 : レスポンス受信待ち異常
0x15 : 格納エリアアドレス異常 (0指定、wka)	0x07 : レスポンス未受信でリトライオーバー
0x16 : サイズ異常 (0以下または16KB以上)	0x08 : データ受信失敗
0x17 : サイズ異常 (非ロングワードサイズ)	0x18 : 格納エリアアドレス異常 (0指定、dmaia)
0x03 : 相手アダプタ種別異常	0x19 : 格納エリアアドレス異常 (0指定、reta)
0xffffffff : CPU状態取り込み異常 (*2)	

(\*2) ネットワークが未接続の場合や、RPDPインストール後にログインしなおしていない場合も発生します。

(5) svcpuctlコマンドエラーメッセージ一覧

No.	エラーメッセージ	意味	ユーザの対応
1	No sitename given for -u option	ユニット名称が指定されていません。	ユニット名称を指定してください。
2	unknown RSSITE	指定サイト名称がありません。	指定サイトを確認してください。
3	Site=%s not found		
4	Internal error (%s)	内部エラーが発生しました。	再試行してください。
5	site (%s) lock busy	サイトは、他の処理で使われています。	
6	site (%s) lock error		
7	communication error (%s, RC=0x%x, 0x%x)	通信エラーが発生しました。	RCコードを基に原因を確認してください。
8	communication error (%s, RC=0x%x)		
9	%s (slot=%d) NON EXIST	スロットが存在しません。	システムジェネレーション情報を確認してください。
10	site (%s) unlock error	サイトの占有解除でエラーが発生しました。	再試行してください。
11	Usage:svcpuctl [{-u site} {-s {-stop   -run}}] [-time] Usage:svcpuctl [-u site] -ss	—	—

## (6) svelogコマンドエラーメッセージ一覧

No.	エラーメッセージ	意味	ユーザの対応
1	usage: svelog [-u site] [-f format] [-logno] [+case] [-d fname] [-o fname]	オプションに誤りがあります。	正しいオプションで起動してください。
2	Unknown RSSITE	RSSITE環境変数が設定されていません。	RSSITE環境変数を設定してください。
3	logno error. logno is 1-999	指定したログ番号が範囲外です。	ログ番号を見直してください。
4	unknown site (サイト名)	指定されたサイトがありません。	サイト名を見直してください。
5	communication error (エラーコード)	開発系マシンとコントローラ間の送受信に失敗しました。	システム管理者に連絡してください。 (*)
6	memory allocate error	メモリが確保できませんでした。	
7	logno "ログ番号": not found	指定されたログ番号のエラーログがありません。	ログ番号を見直してください。
8	no error log.	エラーログがありません。	エラーは発生していません。
9	cannot open "ファイル名"	ファイルのオープンに失敗しました。	システム管理者に連絡してください。 (*)
10	cannot read "ファイル名"	ファイルの読み出し時に異常を検出しました。	
11	specified logno is not found	指定したログ番号のエラーログがありません。	ログ番号を見直してください。
12	svelog : Invalid file name (XXXX)	XXXXに出力されたファイル名称に誤りがあります。	正しいファイル名称で再起動してください。

(\*) 対処方法が「システム管理者に連絡してください。」となっているエラーに関しては、調査用のデータを収集してください。収集するデータファイルは、以下のファイルです。

- %windir%\¥renix¥etc¥log¥s10v¥RPDP\_CTL
- %windir%\¥renix¥etc¥log¥s10v¥1\_RPDP\_CTL

(7) svdhpコマンドエラーメッセージ一覧

(1/2)

No.	エラーメッセージ	意味	ユーザの対応
1	usage: svdhp [-u site] [+count] [-on -off] [-d fname] [-o fname] [-f fname]	オプションに誤りがあります。	正しいオプションで起動してください。
2	Unknown RSSITE	RSSITE環境変数が設定されていません。	RSSITE環境変数を設定してください。
3	No such site (サイト名)	指定されたサイトがありません。	サイト名を見直してください。
4	Some system constants are not defined	定義されていないシステム定数があります。	システム定数の定義を見直してください。
5	Bad realtime environment	システム環境が異常です。	システム管理者に連絡してください。 (*)
6	memory allocate error	メモリが確保できませんでした。	一時的に空きメモリが不足しエラーが発生した可能性があります。メモリに空きがあることを確認後、再実行してください。
7	cannot open "ファイル名"	ファイルのオープンに失敗しました。	ファイルやディレクトリのセキュリティなどの状態を確認してください。
8	cannot read "ファイル名"	ファイルの読み出し時に異常を検出しました。	
9	cannot write "ファイル名"	ファイルの書き込み時に異常を検出しました。	
10	No such AREA (DHP_RD) in salmt	グローバルエリアにDHP読み出しエリア (DHP_RD) がありません。	システム管理者に連絡してください。 (*)
11	Memory access error	コントローラメモリのアクセスに失敗しました。	
12	Memory allocation error (malloc, dhp read area)	dhp読み出しエリアが確保できませんでした。	一時的に空きメモリが不足しエラーが発生した可能性があります。メモリに空きがあることを確認後、再実行してください。
13	Communication error (catch signal)	シグナルを受信しました。	システム管理者に連絡してください。 (*)
14	Communication error (connection timeout)	コネクションタイムアウトが発生しました。	
15	Communication error (connection refused)	RPCサーバが不在です。	
16	Communication error (connection cut)	コネクションが切断されました。	
17	Communication error (connection reset)	コネクションがリセットされました。	
18	Communication error (server closed)	RPCサーバがクローズされました。	

(2/2)

No.	エラーメッセージ	意味	ユーザの対応
19	Communication error (port busy)	回線ポートがbusy状態です。	他の通信の終了を待ち、再実行してください。
20	Communication error (socket create error)	ソケットの生成に失敗しました。	システム管理者に連絡してください。 (*)
21	Communication error (no buffer)	メモリ獲得に失敗しました。	
22	Communication error (network not reached)	ネットワークが未接続状態です。	
23	Communication error (network down)	ネットワーク接続インタフェースがダウンしました。	
24	Communication error (port No error)	ポート番号の取り込みに失敗しました。	
25	Communication error (IP address error)	IPアドレスの取り込みに失敗しました。	
26	Communication error (memory attach failed)	共用メモリのアタッチに失敗しました。	
27	Communication error (trace file cannot open)	トレースファイルのオープンに失敗しました。	
28	Communication error (trace file cannot copy)	トレースファイルのコピーに失敗しました。	
29	Communication error (fatal error)	致命的なエラーが発生しました。	
30	dhp data read error	dhpトレースデータを読み出し時にエラーを検出しました。	
31	Cannot dhp trace ON/OFF	dhpトレース制御時にエラーを検出しました。	
32	svdhp : Invalid file name (XXXX)	XXXXに出力されたファイル名称に誤りがあります。	

(\*) 対処方法が「システム管理者に連絡してください。」となっているエラーに関しては、調査用データを収集してください。収集するデータファイルは、下記となります。

・%windir%\¥renix¥etc¥log¥s10¥¥1\_RPDP\_CTL, Rpdp\_ctl

(8) svcpunowコマンドエラーメッセージ一覧

(1/2)

No.	エラーメッセージ	意味	ユーザの対応
1	Usage:svcpunow [-u site] [-t second]	オプションに誤りがあります。	正しいオプションで起動してください。
2	Unknown RSSITE	RSSITE環境変数が設定されていません。	環境変数RSSITEを設定後、再試行してください。
3	memory allocation error (malloc, puloadinfo area)	PU負荷率情報の読み出しエリア確保に失敗しました。	一時的に空きメモリが不足しエラーが発生した可能性があります。メモリに空きがあることを確認後、再実行してください。
4	cannot get PU load information	PU負荷率の取り込みに失敗しました。	システム管理者に連絡してください。 (*)
5	Not available parameter	使用できないパラメータを検出しました。	
6	No such AREA (puloadinfo) in salmt	PU負荷率読み出しエリアが検索できませんでした。	
7	Communication error (catch signal)	シグナルを受信しました。	
8	Communication error (connection timeout)	タイムアウトが発生しました。	
9	Communication error (connection refused)	RPCサーバが存在しません。	
10	Communication error (connection cut)	RPCサーバが切断されています。	
11	Communication error (connection reset)	コネクションがリセットされています。	
12	Communication error (server closed)	RPCサーバがクローズしています。	
13	Communication error (port busy)	回線ポートがbusy状態です。	
14	Communication error (bad socket specified)	ソケット記述子の指定に誤りがあります。	
15	Communication error (socket create error)	ソケットの生成に失敗しました。	
16	Communication error (no buffer)	メモリ獲得に失敗しました。	一時的に空きメモリが不足しエラーが発生した可能性があります。メモリに空きがあることを確認後、再実行してください。
17	Communication error (network not reached)	ネットワークが未接続状態です。	システム管理者に連絡してください。 (*)

(2/2)

No.	エラーメッセージ	意味	ユーザの対応	
18	Communication error (network down)	ネットワーク接続インタフェースがダウンしています。	システム管理者に連絡してください。 (*)	
19	Communication error (port No error)	ポート番号の取り込みに失敗しました。		
20	Communication error (IP address error)	IPアドレスの取り込みに失敗しました。		
21	Communication error (memory attach failed)	共用メモリのアタッチに失敗しました。		
22	Communication error (trace file cannot open)	トレースファイルのオープンに失敗しました。		
23	Communication error (trace file cannot copy)	トレースファイルのコピーに失敗しました。		
24	Communication error (fatal error)	致命的エラーが発生しました。		
25	Communication error (cannot connection errno = %x)	通信回線の確立に失敗しました。		
26	Communication error (RRB errno = %x)	メモリの読み出しに失敗しました。		
27	Memory access error	S10Vメモリの読み出し、書き込みに失敗しました。		
28	target status error	S10Vコマンドサポートタスクの起動に失敗しました。		
29	Cannot get TCB	TCB情報の読み出しに失敗しました。		
30	command is already execution	他でPU負荷率を測定中のため実行できません。		再試行してください。
31	No sitename given for -u option	サイト名称の指定がありません。		入力可能データを確認後、再試行してください。
32	Site=%s not found	指定したサイトがありません。		
33	PU load measuring period error [second = 1-3600]	測定時間の指定に誤りがあります。		

(\*) 対処方法が「システム管理者に連絡してください。」となっているエラーに関しては、調査用のデータを収集してください。収集するデータファイルは、下記となります。

・ %windir%\¥renix¥etc¥log¥s10v¥1\_RPDP\_CTL, Rpdp\_ctl

(9) svtimexコマンドエラーメッセージ一覧

(1/2)

No.	エラーメッセージ	意味	ユーザの対応	
1	Usage:svtimex [-u site] [tname] [-t second] [tn]	オプションに誤りがあります。	正しいオプションで起動してください。	
2	Unknown RSSITE	RSSITE環境変数が設定されていません。	環境変数RSSITEを設定後、再試行してください。	
3	memory allocation error (malloc, taskloadinfo area)	PU負荷率情報の読み出しエリア確保に失敗しました。	一時的に空きメモリが不足しエラーが発生した可能性があります。メモリに空きがあることを確認後、再実行してください。	
4	cannot get task load information	タスク稼働率の取り込みに失敗しました。	システム管理者に連絡してください。 (*)	
5	cannot get task load information Taskname=%s (%s)	タスク稼働率の取り込みに失敗しました。(タスクが特定できる場合)		
6	Not available parameter	使用できないパラメータを検出しました。		
7	No such AREA (puloainfo) in salmt	タスク稼働率読み出しエリアが検索できませんでした。		
8	Communication error (catch signal)	シグナルを受信しました。		
9	Communication error (connection timeout)	タイムアウトが発生しました。		
10	Communication error (connection refused)	RPCサーバが不在です。		
11	Communication error (connection cut)	RPCサーバが切断されています。		
12	Communication error (connection reset)	コネクションがリセットされています。		
13	Communication error (server closed)	RPCサーバがクローズしています。		
14	Communication error (port busy)	回線ポートがbusy状態です。		
15	Communication error (bad socket specified)	ソケット記述子の指定に誤りがあります。		
16	Communication error (socket creat error)	ソケットの生成に失敗しました。		
17	Communication error (no buffer)	メモリ獲得に失敗しました。		一時的に空きメモリが不足しエラーが発生した可能性があります。メモリに空きがあることを確認後、再実行してください。

(2/2)

No.	エラーメッセージ	意味	ユーザの対応
18	Communication error (network not reached)	ネットワークが未接続状態です。	システム管理者に連絡してください。 (*)
19	Communication error (network down)	ネットワーク接続インタフェースがダウンしています。	
20	Communication error (port No error)	ポート番号の取り込みに失敗しました。	
21	Communication error (IP address error)	IPアドレスの取り込みに失敗しました。	
22	Communication error (memory attach failed)	共用メモリのアタッチに失敗しました。	
23	Communication error (trace file cannot open)	トレースファイルのオープンに失敗しました。	
24	Communication error (trace file cannot copy)	トレースファイルのコピーに失敗しました。	
25	Communication error (fatal error)	致命的エラーが発生しました。	
26	Communication error (cannot connection errno = %x )	通信回線の確立に失敗しました。	
27	Communication error (RRB errno = %x)	メモリの読み出しに失敗しました。	
28	Memory access error	S10Vメモリの読み出し／書き込みに失敗しました。	
29	target status error	S10Vコマンドサポートタスクの起動に失敗しました。	
30	Cannot get TCB	TCB情報の読み出しに失敗しました。	
31	command is already execution	他でタスク稼働率測定中なため実行できません。	
32	No sitename given for -u option	サイト名称の指定がありません。	入力可能データを確認後、再試行してください。
33	Site=%s not found	指定したサイトがありません。	
34	Task measuring period error [second = 1-86400]	測定時間の指定に誤りがあります。	
35	taskname or number set data over (max=10)!!	タスク名称または番号の指定が10個を超えました。	
36	taskname or number error (%s)	タスク名称または番号の指定に誤りがあります。	タスクを開発系マシンとPCs両方に登録してください。
37	%s (%s) task non exist or unmatched	指定されたタスクは開発系マシンとPCs両方に登録されていません。	

(\*) 対処方法が「システム管理者に連絡してください。」となっているエラーに関しては、調査用データを収集してください。収集するデータファイルは、下記となります。

・ %windir%\¥renix¥etc¥log¥s10v¥1\_RPDP\_CTL, Rpdctl

## 付録E RPD P使用上の注意事項

### (1) svdebug ldサブコマンド中断時のリカバリー処理

RPDPでは、svdebugのldサブコマンド中断時（通信エラー等による）のPCs主メモリと開発系マシンの対象サイトのバックアップファイルの不整合を防止するため、ldサブコマンドのリカバリー処理を行います。ここでいうリカバリー処理とは、ldサブコマンドを実行した後の状態に移行するため、中断したldサブコマンドを再実行することをいいます。リカバリー処理は、RPDPコマンド実行時または開発系マシンの立ち上げ時に行われます。ただし、通信エラー等が回復しない場合、

“Communication error (connection timeout)”、“Communication error (inter PU communication time out)”等のエラーメッセージが出力されRPDPコマンドが使用できなくなります。

この場合の対処法として以下のオペレーションがあります。

(オペレーション)

```
svsitecntl -rssrcv サイト名称 [Enter]
```

このオペレーション後、RPDPのリカバリー処理を実行しなくてもコマンドの実行を行えるようになります。ただし、ldサブコマンドのリカバリー処理が未実行の状態ですので、このオペレーション以降、対象サイトに対しldサブコマンドの実行は禁止されます。

エラー内容は、下記となります。

Cannot use ld sub comannd after RSSRCV set (NO.2100-44)

この状態は、svrplコマンドにより対象サイトに対しダウンロードが行われるまで継続されます。

## (2) 使用上の注意事項

## [運用上の注意事項]

- デバッガコマンド (svdebug) に入力するテキストファイルを他マシンからftp転送して使用する場合、asciiモード指定でファイルを転送してください。
- svstedelコマンド使用時に、削除対象となるディレクトリ、ファイルに対して、ファイルマネージャやコマンドプロンプトにてアクセスしている場合、削除できません。また、ファイルマネージャについては、削除対象となるディレクトリ、ファイルに対してアクセスした後、別なディレクトリ、ファイルに対してアクセス (ダブルクリック) するか、ファイルマネージャを閉じない限り削除できません。
- システムジェネレーションコマンド (svgen、svconf、svsitecp、svstedel) 実行中にCtrl+C (SIGINT) を発行した場合、発行するタイミング (各コマンド内にて別コマンドを実行中) によりコマンドが異常終了します。その場合、処理対象となったサイトは無効となるため、svstedelコマンドにてそのサイトを削除してください。

## [制限事項]

- 以下の名称は、予約名であるため、サイト名、ユニット名、ディレクトリ名、ファイル名に使用できません。また、以下の名称に拡張子 (.c .obj .txtなど) を付加した場合も同様です。

• AUX	• CON	• NUL
• COM1	• LPT1	• PRN
• COM2	• LPT2	
• COM3	• LPT3	
• COM4	• LPT4	
• COM5	• LPT5	
• COM6	• LPT6	
• COM7	• LPT7	
• COM8	• LPT8	
• COM9	• LPT9	

- svdebugのld、cm、svサブコマンドの-fオプションで、他のディスク上のファイルを指定できません。

## 付録F svmapの表示フォーマット

マップ情報は、以下に示す情報を出力します。

- (1) ヘッダー、フッター
- (2) 大分割領域情報
- (3) 分割領域情報
- (4) 細分割領域情報
- (5) プログラム情報
- (6) サブプログラム情報
- (7) タスク情報
- (8) グローバル情報
- (9) VAL情報
- (10) IRSUBエントリ情報
- (11) IRGLBエントリ情報
- (12) ULSUBエントリ情報
- (13) 物理メモリの空き情報

### <マップ情報の出力形式>

マップ情報は、以下に示す形式で出力することができます。

- (1) 階層マップ出力
- (2) アドレス順リスト出力
- (3) 名称順リスト出力
- (4) 番号順リスト出力
- (5) 名称指定出力

階層マップ出力は、指定大分割領域、分割領域単位に、論理空間上に配置されるリソースのマップ情報を階層的に出力します。

リスト出力は、指定情報を、アドレス順、名称順、番号順に並べて出力します。

また、リソースの名称を指定し、その名称単独の情報を出力することもできます。

マップ情報の出力フォーマットを以下に示します。

以下に示す表示フォーマット中の下線 (   ) 部分は、出力するマップ情報であり、マップ出力対象により変化することを表します。

(1) ヘッダー、フッター

マップ情報は、出力情報の前後にヘッダー、フッターを出力します。

ヘッダーおよびフッターのフォーマットは下記のとおりです。

(a) ヘッダー

```

** allocator map **                               YYYY/MM/DD hh:mm:ss
site name = site

```

**\*\* allocator map \*\* :**

ヘッダースtringを表示します。

**\*\* allocator map \*\*** : 通常のマップ出力時のヘッダーです。

**\*\* allocator map (CON) \*\*** : PCs側の論理空間マップ出力 (-CON指定) 時のヘッダーです。

**YYYY/MM/DD hh:mm:ss :**

マップ出力コマンド (svmap) を起動した時刻を表示します。

YYYY : 年 (西暦4桁)

MM : 月

DD : 日

hh:mm:ss : 時分秒

site : マップ情報を表示するサイト名を表示します。

(b) フッター

```

** map output end **

```

(2) 大分割領域情報

システムジェネレーションで定義した大分割領域のマップを表示します。

大分割領域の論理空間上の先頭アドレスは固定です。

```

< garea >
gname      laddr      paddr      size
$MAP       20000000  paddr     size
$TASK      30000000  paddr     size
$GLBR      40000000  paddr     size
$GLBRW     50000000  paddr     size
$IRSUB     60000000  paddr     size
    
```

**gname** : 大分割領域の名称です。

**laddr** : 大分割領域先頭の論理アドレスです。

**paddr** : 大分割領域先頭の物理アドレスです。

**size** : 大分割領域のサイズです。

表 A-4 リアルタイムリソースの管理状態

シンボル	状態	意味
@	not-build	バックアップファイルにだけロードされている状態です。
+	defined-POC	
.	defined	バックアップファイルにも実機メモリにもロードされている状態です。
-	defined-CON	実機メモリにだけロードされている状態です。 ダウンロード後、開発系側だけ削除した状態です。
*	unmatch	バックアップファイルにも実機メモリにもロードされているが、整合の取れていない状態です。
_	non_exist2	dbuildしたIRSUB、組み込みサブプログラムをdloadせずにダウンロードした状態です。

(-CON指定時、s、date、lddate、svdateは表示されません。)

## (3) 分割領域情報

svdfaで確保した分割領域の情報を表示します。

```
< area >
garea/aname      raddr      size      laddr      kind bkupfile      date      lddate      svdate
garea/aname      s k raddr    size      laddr      kind bkupfile    YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
garea/_____  raddr      size      laddr
```

**garea** : 親大分割領域のGAREA名称を表示します。

**\$TASK** : タスク格納領域を表します。

**\$GLBR** : 読み出し専用GLB格納領域を表します。

**\$GLBRW** : 読み書き両用GLB格納領域を表します。

**\$IRSUB** : サブプログラム格納領域を表します。

**\$MAP** : マップ情報格納領域を表します。

**aname** : 分割領域名称を表示します。

分割領域名称の空白は、空き領域であることを表します。

**s** : リソースの状態を表します。

リソースの状態については、表A-4を参照してください。

**k** : 所有者種別を表示します。(s: システム、u: ユーザ)

**raddr** : 分割領域先頭の大分割領域先頭からの相対アドレスを16進8桁  
固定で表示します。

**size** : 分割領域のサイズを16進8桁固定で表示します。

**laddr** : 分割領域の先頭論理アドレスを16進8桁固定で表示します。

**kind** : 分割領域種別

**glbi** : 初期値あり読み書き両用グローバル領域を表します。

**glbw** : 初期値なし読み書き両用グローバル領域を表します。

**glbr** : 初期値あり読み出し専用グローバル領域を表します。

**task** : タスク格納領域を表します。

**sub** : サブプログラム格納領域を表します。

**ostbl** : OSリザーブ領域を表します。

**bkupfile** : 分割領域の初期値を格納したバックアップファイル名を表示  
します。

初期値なしGLBのエリアの場合は空白で表示します。

**date** : svdfaで分割領域を生成した時刻を表示します。

**lddate** : PCsメモリにダウンロードした時刻を表示します。

ダウンロードされていない場合は空白で表示します。

**svdate** : デバッグのsvサブコマンドでバックアップファイルに保存し  
た時刻を表示します。

保存されていない場合は空白で表示します。

**date**, **lddate**, **svdate**は、-fオプション指定時だけ表示します。

(4) 細分割領域情報

細分割領域の情報を表示します。

```

< sarea >
garea/aname/sname          raddr      size      date      svdate
garea/aname/sname          laddr      size      YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
garea/aname/              raddr      size      laddr
    
```

**garea** : 親大分割領域のGAREA名称を表示します。

表示するGAREA名称の意味は、「(3) 分割領域情報」の表示と同様です。

**aname** : 分割領域名称を表示します。

**sname** : 細分割領域名称を表示します。

細分割領域名称の空白は、空き領域であることを表します。

**s** : リソースの状態を表します。

リソースの状態については、表A-4を参照してください。

**k** : 所有者種別を表示します。(s: システム、u: ユーザ)

**raddr** : 細分割領域の先頭アドレスを、分割領域先頭からの相対バイトアドレス (16進8桁固定) で表示します。

**size** : 細分割領域のサイズを16進8桁固定で表示します。

**laddr** : 細分割領域の先頭論理アドレスを16進8桁固定で表示します。

**date** : svdfsで細分割領域を確保した時刻、またはsvloadでプログラムをバックアップファイルにロードした時刻を表示します。

**lddate** : PCsメモリにダウンロードした時刻を表示します。

ダウンロードされていない場合は空白で表示します。

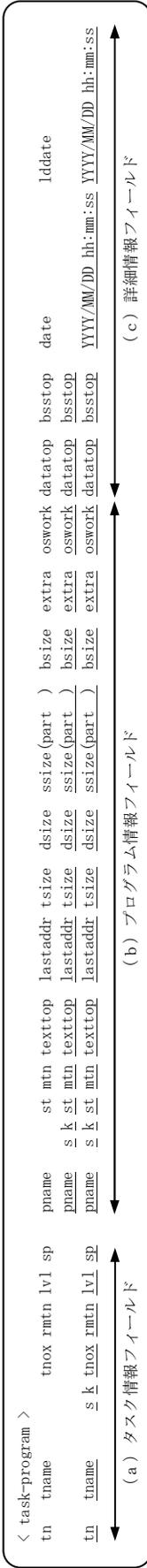
**svdate** : デバッグのsvサブコマンドでバックアップファイルに保存した時刻を表示します。

保存されていない場合は空白で表示します。

**date, lddate, svdate**は、-fオプション指定時だけ表示します。( -CON指定時、s、date、lddate、svdateは表示されません。)

(5) プログラム情報

プログラムに関する情報を表示します。



プログラムの表示フォーマットは、「(7) タスク情報」に示すフォーマットと同じです。

各フィールドの意味は、「(7) タスク情報」を参照してください。

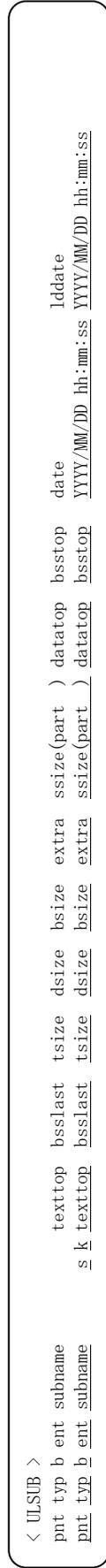
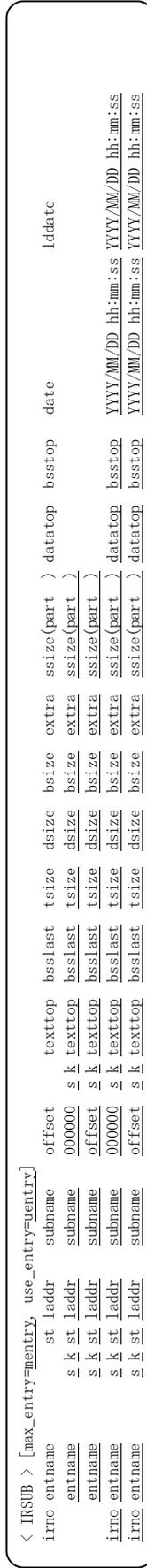
プログラム情報表示とタスク情報表示の相違点は以下の2点です。

- ・デフォルトのソート順がプログラム指定 (-p) ではプログラムの名称順に、タスク情報指定 (-t) ではタスク番号順となります。
- ・名称指定の表示で、プログラム指定 (-p) では指定名称をプログラム名として扱います。

タスク情報指定 (-t) では指定名称をタスク名として扱います。

(6) サブプログラム情報

サブプログラムに関する情報を表示します。



サブプログラム情報の表示フォーマットは、「(10) IRSUB エントリ情報」および「(12) ULSUB エントリ情報」に示すフォーマットと同じです。

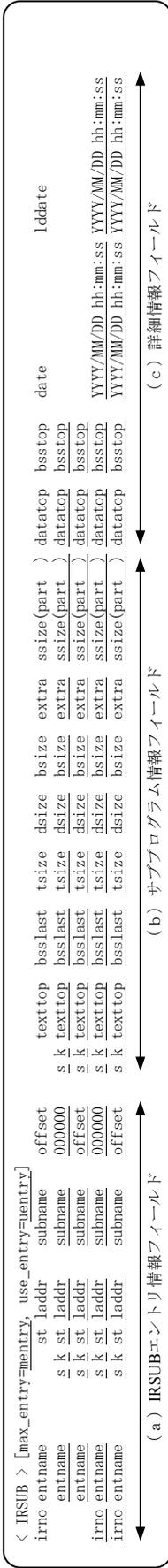
IRSUB と組み込みサブプログラムの両方の情報を表示します。

各フィールドの意味は、「(10) IRSUB エントリ情報」および「(12) ULSUB エントリ情報」を参照してください。





(10) IRSUBエン트리情報  
IRSUBのエントりに関する情報を表示します。



mentry : IRSUBの登録可能エントリ数を10進6桁固定で表示します。  
 uentry : IRSUBの使用中のエントリ数を10進6桁固定で表示します。

(a) IRSUBエン트리情報フィールド

irno : 間接リンクテーブルのエントリ番号  
 : IRSUBがビルドされていない場合、irnoは空白となります。

entname : IRSUBのエントリ名称  
 s : IRSUBのエントリの状態を表します。  
 : リソースの状態については、表A-4を参照してください。

k : 所有者種別を表示します。(s: システム、u: ユーザ)  
 (IRSUBがビルドされていない場合、空白を表示します。)

st : IRSUBのエントリ種別および割り当て状態を表示します。  
 il : ビルドされていないIRSUBのトップエントリ関数を表します。  
 ml : ビルドされていないIRSUBのマルチエントリ関数を表します。  
 ib : ビルドされているIRSUBのトップエントリ関数を表します。  
 mb : ビルドされているIRSUBのマルチエントリ関数を表します。

laddr : エントリポイントの論理空間上のアドレスを16進8桁固定で表示します。

subname : エントリの含まれるサブプログラムの名称を表示します。

offset : サブプログラムの先頭からエントリポイントまでの相対値を16進6桁固定で表示します。

(b) サブプログラム情報フィールド

s : サブプログラムの状態を表します。  
 : リソースの状態については、表A-4を参照してください。

k : 所有者種別を表示します。(s: システム、u: ユーザ)

texttop : テキスト部の先頭論理アドレスを16進8桁固定で表示します。

bsslast : bss部の最終論理アドレスを16進8桁固定で表示します。  
 : テキスト部の先頭論理アドレス (extra) を含んだ値です。

tsize : テキスト部のサイズを16進6桁固定で表示します。  
 dsize : データ部のサイズを16進6桁固定で表示します。  
 bsize : bss部のサイズを16進6桁固定で表示します。  
 extra : load時に指定した冗長バイトサイズを16進6桁固定で表示します。

ssize (part) : stackのサイズを16進6桁固定で表示します。  
 (part) にはローダに指定したサブプログラム自身の使用するスタックサイズを表示します。

datatop : データ部の先頭論理アドレスを16進8桁固定で表示します。

bsstop : BSS部の先頭論理アドレスを16進8桁固定で表示します。

date : svbuildでビルドした時刻を表示します。  
 ビルドされていない場合は空白で表示します。

lddate : PCsメモリにダウンロードした時刻を表示します。  
 ダウンロードされていない場合は空白で表示します。

詳細情報フィールドは、-fオプションを指定した場合だけ表示します。

(11) IRGLBエン트리情報



**mentry** : GLBの登録可能エントリ数を10進6桁固定で表示します。  
**umentry** : GLBの使用中のエントリ数を10進6桁固定で表示します  
 (OS使用分の13個を含みます)。

(a) IRGLBエン트리ファイルド

**irno** : 間接リンクテーブルのエントリ番号 (IRGLBとしてビルドされていない場合は空白を表示します。)  
**entname** : IRGLBのエントリ名称 (IRGLBとしてビルドされていない場合は空白を表示します。)

**s** : IRGLBエントリの状態を表示します。  
 リソースの状態については、表A-4を参照してください。  
**k** : 所有者種別を表示します。(s: システム、u: ユーザ)

**laddr** : エントリポイントの論理空間上のアドレスを16進8桁固定で表示します。  
 (IRGLBとしてビルドされていない場合は空白を表示します。)

**salname** : エントリの含まれる細分割領域の名称を表示します。  
 エントリのアドレスが絶対アドレスで与えられている場合は\* (アスタリスク) で表示します。  
 16進6桁固定で表示します。

**offset** : 細分割領域の先頭からエントリポイントまでの相対値を16進6桁固定で表示します。

(b) GLBエリアファイルド  
**kind** : 細分割領域種別  
**glbi** : 初期値あり読み書き両用グローバル領域を表示します。  
**glbw** : 初期値なし読み書き両用グローバル領域を表示します。  
**glbr** : 初期値あり読み出し専用グローバル領域を表示します。

**s** : GLBの状態を表示します。  
 リソースの状態については、表A-4を参照してください。  
**k** : 所有者種別を表示します。(s: システム、u: ユーザ)  
**aname** : 親分割領域の名称を表示します。  
**sasize** : 領域のサイズを16進8桁固定で表示します。  
**datasize** : データのサイズを16進8桁固定で表示します。  
**aoff** : 分割領域先頭からの相対バイトアドレスを16進8桁固定で表示します。

(c) 詳細情報ファイルド  
**date** : svirglbまたはsvdfs-eでIRGLBとしてビルドした時刻を表示します。  
**IRGLB**としてビルドされていない場合は空白で表示します。  
**ldddate** : PCsメモリにダウンロードした時刻を表示します。  
 ダウンロードされていない場合は空白で表示します。  
 詳細情報ファイルドは、-fオプションを指定した場合だけ表示します。

(12) ULSUBエンタリ情報

```

< ULSUB >
pnt typ ent b subname      texttop bsslast tsize dsize bsize extra ssize (part ) datatop      date      lddate
pnt typ ent b subname      s k texttop bsslast tsize dsize bsize extra ssize (part ) datatop      hh:mm:ss YYYY/MM/DD hh:mm:ss

```

- pnt : 組み込みサブルーチンの組み込みポイントを表示します。
- typ : 組み込みサブルーチンのタイプ (osまたはuser) を表示します。
- ent : 組み込みサブルーチンのエントリ番号を表示します。
- b : 組み込みサブルーチンのビルド状態を表します。  
リソースの状態については、表A-4を参照してください。
- subname : 組み込みサブルーチン名称を表示します。
- s : 組み込みサブルーチンの状態を表します。  
リソースの状態については、表A-4を参照してください。
- k : 所有者種別を表示します。(s: システム、u: ユーザ)
- texttop : テキスト部の先頭論理アドレスを16進8桁固定で表示します。
- bsslast : bss部の最終論理アドレスを16進8桁固定で表示します。  
最終論理アドレスは冗長バイトサイズ (extra) を含んだ値です。
- tsize : テキスト部のサイズを16進6桁固定で表示します。
- dsize : データ部のサイズを16進6桁固定で表示します。
- bsize : bss部のサイズを16進6桁固定で表示します。
- extra : load時に指定した冗長バイトサイズを16進6桁固定で表示します。
- ssize(part) : stackのサイズを16進6桁固定で表示します。  
(part) にはローダに指定したサブプログラム自身の使用するスタックサイズを表示します。
- datatop : データ部の先頭論理アドレスを16進8桁固定で表示します。
- bssstop : BSS部の先頭論理アドレスを16進8桁固定で表示します。
- date : svbuildで組み込みサブルーチンとしてビルドした時刻を表示します。  
ビルドされていない場合はloadした時刻を表示します。
- lddate : PCsメモリにダウンロードした時刻を表示します。  
ダウンロードされていない場合は空白で表示します。

詳細情報フィールドは、-fオプションを指定した場合だけ表示します。

(13) 物理メモリの空き情報

```

< physical memory >
garea      use      free      total
$TASK     xxx Kbyte yyy Kbyte zzz Kbyte
$GLBR     xxx Kbyte yyy Kbyte zzz Kbyte
$GLBRW    xxx Kbyte yyy Kbyte zzz Kbyte
$IRSUB    xxx Kbyte yyy Kbyte zzz Kbyte
    
```

- use : GAREAごとに使用している物理メモリのサイズ
- free : GAREAごとの空き物理メモリのサイズ
- total : サイト構築時に割り当てたGAREAごとの物理メモリのサイズ

(14) 階層マップ出力

階層マップ出力は以下に示す形式で出力します。

(a) garea/areaの階層マップ (+gn gname -G -a指定)

```

** allocator map **
site name = site
< garea >
gname      laddr      paddr      size      bkupfile
gname     laddr     paddr     size     bkupfile
< area >
garea/area raddr      size      laddr      date      svdate
garea/area s_k raddr      size      laddr      YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
garea/_____ raddr      size      laddr
** map output end **
    
```

(注) date、lddate、svdateは、-fオプション指定時だけ表示します。

(b) garea/area/sareaの階層マップ (+gn gname -G -a -e指定)

```

** allocator map **
site name = site
< garea >
gname      laddr      paddr      size
gname      laddr      paddr      size
< area >
garea/aname raddr      size
garea/      raddr      size
           s k raddr size
           raddr size
           date      lddate      svdate
           YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
< sarea >
garea/aname/sname raddr      size
garea/aname/      raddr      size
           s k raddr size
           raddr size
           date      lddate      svdate
           YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
** map output end **

```

(注) date、lddate、svdateは、-fオプション指定時だけ表示します。

(c) area/sareaの階層マップ (+gn aname -a -e指定)

```

** allocator map **
site name = site
< area >
garea/aname raddr      size
garea/aname      raddr      size
           s k raddr size
           raddr size
           date      lddate      svdate
           YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
< sarea >
garea/aname/sname raddr      size
garea/aname/      raddr      size
           s k raddr size
           raddr size
           date      lddate      svdate
           YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
** map output end **

```

(注) date、lddate、svdateは、-fオプション指定時だけ表示します。

(15) デフォルトの表示フォーマット

(a) デフォルトの表示フォーマット (詳細情報なし)

-u オプション以外のすべてのオプションを省略した場合は、分割領域・細分割領域のアドレス順リスト、タスク・プログラム、IRSUB、組み込みサブルーチン、IRGLBの番号順リスト、VALの名称順リストを詳細情報なしで出力します。表示形式を以下に示します。

```

** allocator map **
      site name = site
      < garea >
      gname laddr paddr size
      gname laddr size
      < area >
      garea/aname raddr size kind bkupfile
      garea/aname s k raddr laddr kind bkupfile
      garea/_____ raddr size
      < sarea >
      garea/aname/aname raddr size laddr
      garea/aname/aname s k raddr size laddr
      garea/aname/_____ raddr size
      < task-program >
      tn tname
      tn tname tnox rmtn lvl sp
      tn tname s k tnox rmtn lvl sp
      < IRSUB > [max_entry=entry, use_entry=entry]
      irno entname
      entname st laddr subname offset
      entname s k st laddr subname 000000
      irno entname
      entname s k st laddr subname offset
      irno entname
      entname s k st laddr subname 000000
      < ULSUB >
      pnt typ ent b subname
      pnt typ ent b subname s k texttop bsslast tsize dsize bsize extra ssize (part)
      < global, irglb > [max_entry=entry, use_entry=entry]
      irno entname
      laddr salname offset kind s k aname sasize datasize aoff
      irno entname
      salname s k laddr salname 000000 kind s k aname sasize datasize aoff
      irno entname
      salname s k laddr salname 000000 kind s k aname sasize datasize aoff
      irno entname
      salname s k laddr salname ***** kind s k aname sasize datasize aoff
      < value > [max_entry=entry, use_entry=entry]
      ename valhex valdec
      ename k valhex valdec
      ** map output end **
  
```

(b) デフォルトの表示フォーマット (詳細表示)

-u、-fオプション以外のすべてのオプションを省略した場合は、分割領域・細分割領域のアドレス順リスト、タスク・プログラム、IRSUB、組み込みサブルーチン、IRGLBの番号順リスト、VALの名称順リストを詳細情報付きで出力します。表示形式を以下に示します。

```

** allocator map **
site name = site
< garea >
  gname      laddr      paddr      size
  gname      laddr      paddr      size
< area >
  garea/_____ raddr      size      kind      bkupfile      svdate
  garea/_____ s k raddr  size      laddr      kind      bkupfile      YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
  garea/_____ raddr      size      laddr
  garea/_____ s k raddr  size      laddr
  garea/_____ raddr      size
< sarea >
  garea/aname/_____ raddr      size      laddr      svdate
  garea/aname/_____ s k raddr  size      laddr      YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
  garea/aname/_____ raddr      size
< task-program >
tn tname      tnox      rmtn      lvl      sp      pname      pname      pname      st      mtm      texttop      lastaddr      tsize      dsize      ssize      (part )      bsize      extra      oswork      datatop      bsstop      date      lddate
tn tname      s k tno x  rmtn  lvl  sp      pna me      pna me      pna me      s k st  mtm  texttop  lastaddr  tsize  dsize  ssize  (part )  bsize  extra  oswork  datatop  bsstop  YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
< IRSUB > [max_entry=entry, use_entry=entry]
irno entname      laddr      subname      offset
irno entname      s k laddr  subname  000000
irno entname      s k laddr  subname  offset
irno entname      s k laddr  subname  000000
irno entname      s k laddr  subname  offset
< ULSUB >
pnt typ ent b subname      texttop      bsslast      tsize      dsize      bsize      extra      ssize      (part )      datatop      bsstop      date      lddate
pnt typ ent b subname      s k texttop  bsslast  tsize  dsize  bsize  extra  ssize  (part )  datatop  bsstop  YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
< global, irglb > [max_entry=entry, use_entry=entry]
irno entname      laddr      salname      offset      kind      s k name      sasize      datasize      aoiff      date
irno entname      s k laddr  salname  000000  kind  s k name  sasize  datasize  aoiff  YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
irno entname      s k laddr  salname  offset      kind      s k name      sasize      datasize      aoiff      YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
irno entname      s k laddr  salname  *****  kind      s k name      sasize      datasize      aoiff      YYYY/MM/DD hh:mm:ss YYYY/MM/DD hh:mm:ss
< value > [max_entry=entry, use_entry=entry]
ename      valhex      valdec      date
ename      k valhex  valdec  YYYY/MM/DD hh:mm:ss
** map output end **

```

## 付録G svdebug (オンラインデバッガ) md、sdの表示フォーマット

### (1) mdサブコマンドの表示フォーマット

- データ表示 (プリント) 時の表示フォーマット

```
0xaaaaaaaa dddddddd dddddddd dddddddd dddddddd `.....`
  アドレス      表示データ      文字コード
```

アドレス : 表示データの先頭アドレスを16進で表示します。

表示データ : 指定されたデータ出力形式・データ長に従い、アドレスの内容を表示します。

1行で最大16バイト分のデータを表示します。

データ出力形式が浮動小数点 (-f-l, -fd) のとき、アドレスの内容が下記データである場合は、16進数に変換して表示します。また、16進数表示の後に対応する文字列を表示します。

浮動小数点データ	文字列	表示例	
		単精度	倍精度
非数	Na	0x7fffffff : Na	0xfff00000 0x00000001 : Na
無限大	In	0x7f800000 : In	0xfff00000 0x00000000 : In
表現できる最大値	Ma	0x7f7fffff : Ma	0x7fefffff 0xffffffff : Ma
表現できる最小値	Mi	0xff7fffff : Mi	0xffefffff 0xffffffff : Mi

文字コード : 表示データを2バイトコードで表示します。

直前に表示した行の内容と同一データが行単位で連続する場合、上記フォーマットにかわって下記のメッセージを表示します (-allオプション指定時は、連続するデータすべてを表示します)。

```
0xaaaaaaaa-0xaaaaaaaa as previous
  |
  |-----> 同一データの先頭アドレス
  |
  |-----> 同一データの最終アドレス
```

- データ変更 (パッチ) 時の表示フォーマット

```
0xaaaaaaaa dddddddd :
  アドレス 表示データ
```

● 表示例

mdの表示例を、データ出力形式とデータ長の両オプションの組み合わせにより下図に示します。

```

*****|*****|*****|*****|*****|*****|*****|*****
16進4バイト 0x00ec0000 0000000a 00000064 000003e8 00002710 `.....d.....'.`
表示(-h、-l) 0x00ec0010 000186a0 `.....'
                ....

16進2バイト 0x00ec0000 0000 000a 0000 0064 0000 03e8 0000 2710 `.....d.....'.`
表示(-h、-w) 0x00ec0010 0001 86a0 `.....'
                ....

16進1バイト 0x00ec0000 00 00 00 0a 00 00 00 64 00 00 03 e8 00 00 27 10 `.....d.....'.`
表示(-h、-b) 0x00ec0010 00 01 86 a0 `.....'
                ....

10進4バイト 0x00ec0000          10          100          1000          10000
表示(-d、-l) 0x00ec0010          100000

10進2バイト 0x00ec0000          0   10          0   100          0   1000          0 10000
表示(-d、-w) 0x00ec0010          1 -31072

10進1バイト 0x00ec0000          0   0   0 10          0   0   0 100          0   0   3 -24          0   0
表示(-d、-b)          39  16
                0x00ec0010          0   1 -122 -96

単精度実数 0x00ec0020          1. 1200000          2. 1229999          10. 1230001          20. 1233997
表示(-f、-l) 0x00ec0030          100. 123451

倍精度実数 0x96000000          1. 0000000000000000E+00          2. 0000000000000000E+00
表示(-fd) 0x96000010          -1. 0000000000000000E+00          1. 0000000000000000E+100
                0x96000020          0x7fefffff 0xfffffff:Ma          0xffefffff 0xfffffff:Mi
    
```

(2) sdの表示フォーマット

- データ表示 (プリント) 時の表示フォーマット

```
0xaaaaaaaa(0x111111) dddddddd dddddddd dddddddd dddddddd `.....`
  アドレスオフセット      表示データ      文字コード
```

アドレス : 表示データの先頭アドレスを16進で表示します。

オフセット : データ先頭からのオフセットを表示します。

表示データ : 指定されたデータ出力形式・データ長に従い、アドレスの内容を表示します。

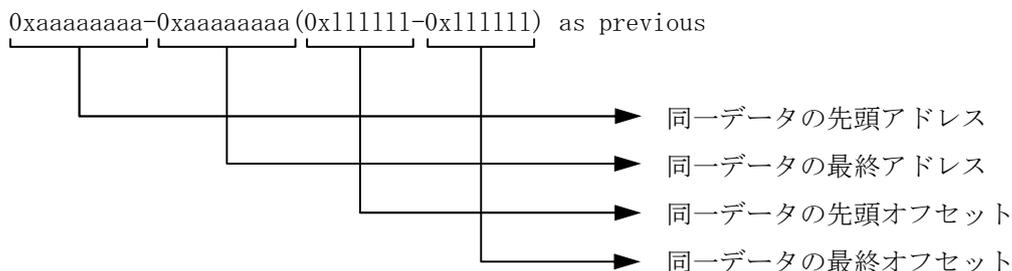
1行で最大16バイト分のデータを表示します。

データ出力形式が浮動小数点 (-f-l, -fd) のとき、アドレスの内容が下記データである場合は、16進数に変換して表示します。また、16進数表示の後に対応する文字列を表示します。

浮動小数点データ	文字列	表示例	
		単精度	倍精度
非数	Na	0x7fffffff : Na	0xffff00000 0x00000001 : Na
無限大	In	0x7f800000 : In	0xffff00000 0x00000000 : In
表現できる最大値	Ma	0x7f7fffff : Ma	0x7fefffff 0xffffffffff : Ma
表現できる最小値	Mi	0xff7fffff : Mi	0xffefffff 0xffffffffff : Mi

文字コード : 表示データを2バイトコードで表示します。

直前に表示した行の内容と同一データが行単位で連続する場合、上記フォーマットにかわって下記のメッセージを表示します (-allオプション指定時は連続するデータすべてを表示します)。



- データ変更 (パッチ) 時の表示フォーマット

```
0xaaaaaaaa(0x111111) dddddddd :
  アドレス オフセット 表示データ
```

● 表示例

sdの表示例を、データ出力形式とデータ長の両オプションの組み合わせにより下図に示します。

	*****	*****	*****	*****	*****	*****	*****	*****	*****									
16進4バイト 表示(-h、-l)	0x00ec0000(0x000000)	0000000a	00000064	000003e8	00002710	.....d.....'	.....											
	0x00ec0010(0x000010)	000186a0				.....												
16進2バイト 表示(-h、-w)	0x00ec0000(0x000000)	0000	000a	0000	0064	0000	03e8	0000	2710									
	0x00ec0010(0x000010)	0001	86a0															
16進1バイト 表示(-h、-b)	0x00ec0000(0x000010)	00	00	00	0a	00	00	00	64	00	00	03	e8	00	00	27	10	.....d
	0x00ec0010(0x000010)	00	01	86	a0													.....
10進4バイト 表示(-d、-l)	0x00ec0000(0x000000)		10		100		1000		10000									
	0x00ec0010(0x000010)		100000															
10進2バイト 表示(-d、-w)	0x00ec0000(0x000000)	0	10	0	100	0	1000	0	10000									
	0x00ec0010(0x000010)	1	-31072															
10進1バイト 表示(-d、-b)	0x00ec0000(0x000000)	0	0	0	10	0	0	0	100	0	0	3	-24					
	0x00ec0010(0x000010)	0	0	39	16													
単精度実数 表示(-f、-l)	0x00ec0020(0x000000)	1.1200000	2.1229999	10.1230001	20.1233997													
	0x00ec0030(0x000010)	100.123451																
倍精度実数 表示(-fd)	0x96000000(0x000000)	1.0000000000000000E+00	2.0000000000000000E+00															
	0x96000010(0x000010)	-1.0000000000000000E+00	1.0000000000000000E+100															
	0x96000020(0x000020)	0x7fffffff	0xffffffff:Ma	0xffefffff	0xffffffff:Mi													

## 付録H ライブラリの使用するスタックサイズ一覧

ライブラリが使用するスタックサイズの一覧を以下に示します。

(1/2)

ライブラリ	関数名	スタックサイズ
C標準ライブラリ (libsh4nbmdn.lib libsh4nbmzz.lib)	atof	408
	freexp	8
	ldexp	20
	memchr	0
	memset	12
	modf	40
	sscanf	528
	sprintf	752
	strcat	0
	strchr	0
	strcmp	20
	strcpy	24
	strncpy	4
	strlen	0
	strncat	4
	strncmp	4
	strncpy	0
	strpbrk	4
	strchr	12
	strspn	4
	strtod	408
	strtol	68
	vsprintf	752
	acos	196
	asin	184
	atan	156
	atan2	176
	ceil	28
	exp	92
	fabs	0
	floor	28
	fmod	40

(2/2)

ライブラリ	関数名	スタックサイズ
C標準ライブラリ (libsh4nbmdn.lib libsh4nbmzz.lib)	log	60
	log10	72
	pow	132
	cos	84
	sin	84
	cosh	112
	sinh	144
	sqrt	8
	tan	132
	tanh	156
	libcrs.lib	fpgetmask
fpgetround		0
fpgetsticky		0
fpsetmask		0
fpsetround		0
fpsetsticky		0
fpcheck		0
fpchecko		0
libfirad.lib	irglbad	0
	irsubad	0
libcpms.lib	memcpy (*)	28

(\*) ローダでロードしたプログラム、サブプログラムにおいて、memcpy()はC標準ライブラリではなく、CPMSライブラリのmemcpy()が使用されます。