「ITRON4.0®仕様に完全準拠し拡張を含む
オープンソース ITRON®仕様 の開発」
ダイナミックローディング機能 取り扱い説明書

発行年 月 日 第 版

編集：株式会社エーアイコーポレーション
目次

1. 基本概念及び用語の説明
   - はじめに
   - 基本スキーム
   - ソフトウェア構成の説明
   - 用語の説明

2. 導入編
   - 準備
     - ソフトウェアのインストール
     - ソフトウェアを用いたクロスコンパイル環境の導入
   - ダイナミックダウンロードソフトウェアのインストール
   - コンフィギュレータのセットアップ
     - ビルド方法
     - 起動方法
   - 拡張した静的機能

3. ダウンローディングサーバのセットアップ
   - ダウンローディングサーバビルド方法
     - 起動方法
     - ダウンローディングサーバ設定ファイル
   - サーバ作業ディレクトリの作成
   - モジュール管理データベースファイル

4. サーバダウンロードシェル
   - ダウンロードシェルプログラムのビルト方法
     - 起動方法
     - シェルコマンド一覧
     - シェルコマンドリファレンス

5. ターゲットダウンロードシェル
   - 起動方法
   - シェルコマンドリファレンス

6. ダウンローディングターゲットのセットアップ
   - ターゲットシステムのセットアップ
   - ターゲットソフトウェアの構成
   - ローダ機能のコンフィグレーション説明
# 目次

1 章 チュートリアル編

<table>
<thead>
<tr>
<th>ダウンローディングターゲットシステムの準備</th>
<th>ペースモジュールのコンフィグレーション及びビルド</th>
<th>ロードモジュールの作成</th>
<th>ダウンロードの操作</th>
</tr>
</thead>
</table>

2 章 補足資料編

<table>
<thead>
<tr>
<th>他のシステムへの移植ガイド</th>
<th>フル機能 [カーネルサービス]</th>
<th>ネットワーク サービス</th>
<th>フラッシュメモリドライバ</th>
</tr>
</thead>
</table>

3 章 [使用したダウンロードモジュールのデバッグ方法の紹介]
1章 基本概念及び用語の説明

はじめに

組み込みシステムの多くは保守が困難な場所に隠蔽され設置されています。またこのようなシステムは一旦稼働すると、保守目的で電源をなかなか落とせない状況になります。日本の組み込みシステムにはコードサイズがコンパクトでリアルタイム応答特性に優れた Venue 頭 topo 仕様が多く採用されています。しかし従来の Venue 頭 topo 仕様では、機能を含めた全てのプログラムが 1 リングされ、機器に頭 topo として組み込まれる事を前提としていました。頭 topo 仕様は自身に動的なプログラム更新機能を持っていないため、機器の設置後バグが発見されても、容易にプログラムの差し替えが出来ませんでした。特に設備に隠蔽された機器の場合、head 交換によるプログラム交換は大掛かりな保守工事を必要としました。ダイナミックローディング機関は Venue 頭 topo 仕様フルセットカーネル上でミドルウェアとして動作し、ネットワークを介してシステムを止めることなくプログラムの更新や新規追加を可能にするものです。従来保守が難しかった設備に隠蔽された機器もこの機能を使用する事により、ネットワークに接続されていれば、Venue 頭 topo のリアルタイム特性を損なうことなく、汎用のような動的なプログラム更新の利便性を得ることが可能となります。

図 1.1.1 サーバに最新のソフトウェアをダウンロード
携帯電話に代表されるように、何らかのユーザ操作インタフェースを持った機器においても仕様が広く使われています。ダイナミックダウンローディング機能は今まで不可能であったユーザ自身の操作による（クライアントデマンド）、機器へのソフトウェアアップグレードやソフトウェア不具合の改修を可能とします。

図 1.1.2 ユーザインタフェース機能を持った機器のソフトウェアアップデート

昨今、多くの機器がインターネットなどのネットワーク接続機能を内蔵したものが増えてきてしまった。しかし重要な機能のみを実装する組込み機器の世界においても依然ネットワーク機能を持たない機器が存在することも事実です。このような機器において仕様や仕様のリムーバブルストレージデバイスを使用して修正が必要な不具合ソフトウェアモジュールのみ更新できれば保守の容易性が向上します。なお全体書き換え更新の場合、システム全体を止める必要がありますが、部分更新方式の場合、更新機能のみの一時的停止で、カーネル機能を含む他の機能を動作させたままでソフトウェアの更新が可能となります。ダイナミックダウンローディング機能はネットワークで接続された機器はもとより、このようなネットワーク機能を持たない機器においてもダイナミックダウンロードによるソフトウェア更新を可能とします。

図 1.1.3 リムーバブルストレージを使った機器のソフトウェアアップデート


## 2. 基本スキーム

パーソナルコンピュータやワークステーションにおいて動作する汎用OSでは、OS自身が動的なアドレス解決機能を持っています。一方、機器専用のアプリケーションプログラムを動作させる事を目的とした組み込みシステムでは、ランタイムに冗長なプログラムリンク機能を搭載するのはハードウェアコストの面で許されないので一般的です。そこでダイナミックダウンロード機構では下図のようにシンボルアドレス解決機能をパーソナルコンピュータシステム等の汎用プラットホームで行わせ、ターゲット機器ではローダ機能のみを搭載し、ネットワーク連携で動的にアドレス解決する機構を採用し実現しています。

![基本スキーム](image)

図 2-1 基本スキーム
図 1.3.1 ソフトウェア構成の説明

図 1.3.1 ソフトウェア構成
ダイナミックローディング機構は以下のソフトウェア部品で構成されます。

ターゲットシステム サイド
ローダ機能はターゲット組込み機能でフロールセットカーネル上で動作し、カーネルにプログラムのダイナミックダウンローディング機能を追加するミドルウェアです。ローダ機能はロー - ディングエージェント機能、モジュール管理機能、転送プロトコル機能で構成されます。また外部のソフトウェアとしてネットワーク機能と言語バンドムライブラリ機能を必要とします。以降それぞれの機能ブロックの役割を説明します。

ローディングエージェント機能
ローディングエージェントはダウンロードサーバとのセッションを管理し、トランスアクション要求を処理します。

モジュール管理機能
ダイナミックローディングされたロードモジュールの管理を行います。ローディングエージェントに対して、ロードモジュールの問合せ、削除、追加のサービスを提供します。

転送プロトコル機能
転送プロトコル機能はネットワークやストレージデバイス等の情報伝達手段を抽象化します。ローディングエージェント機能と情報媒体アクセス機能の間に立脚し、ダウンロードトランスアクションサービスを提供します。情報媒体アクセス機能として外部プロダクトを製造、外部の柔軟性を移植し実現しています。

メモリドライバ機能
メモリドライバ機能はフラッシュメモリ等の不揮発性メモリへの書き込み操作を抽象化するドライバ機能です。カードボードに搭載されるフラッシュメモリに対して消去、プログラムデータ書き込みを実現します。

検証機能
検証機能はダイナミックダウンローディング機構が実現する動的なプログラム更新機能をデモンストレーション実行するサンプルロードモジュールです。検証機構でサーバを使用し、動的なホームページ閲覧サービス機能のブラウジングデモンストレーションを行います。
6. 外部プロダクト
以下は、外部プロダクトを利用します。
・新制品 他社製社製　プロトコルスタック

△ダウンロードサーバ・サイト

ダウンロードサーバシステムは、以下のサーバで動作します。オペレーテータからのダウンロードデマンドを受け付け、ターゲット機器に接続し、動的なプログラムのダウンロードを遂行します。ダウンロードサーバシステムはダウンローディングサーバ機能、モジュールビルダ機能、ダウンローディングシェル機能にて構成されます。また外部のソフトウェアとしてRedHat Linux 9が必要となります。以降それぞれの機能ブロックの役割を説明します。

△ ダウンローディングサーバ機能
ダウンローディングサーバ機能は、サーバ上のプロセスとして動作し、ダウンローディングシェルプログラムからの処理要求に対してダウンローディングトランザクション処理を行うプログラムです。

△ モジュールビルダ機能
モジュールビルダ機能はダウンローディングサーバ機能の最終ロードモジュールを作成する処理を行います。

△ ダウンローディングシェル機能
ダイナミックダウンローディング機構におけるユーザインタフェース機能です。サーバ上のプロセスとして動作し、ダウンローディングサーバ機能とはドメインのソケットで接続します。ダウンローディングトランザクションを発生させるコマンドラインインタフェースのユーザインタフェースを提供します。

△ 外部プロダクト
・新製品 他社製 他社　動作
ツール
ツールシステムは ダイナミックダウンロード機構におけるロードモジュール作成をバックエンドで支援します。

コンフィギュレータ機能
コンフィギュレーションファイルを解釈し、カーネル機能、モジュール管理機能へモジュールの初期化情報を出力するツール機能です。

外部プロダクト
以下の外部プロダクトを利用します。
・RedHat Linux 9 --- "®ìOS
用語の説明

フルセットカーネル
仕様で定義された拡張プロファイル機能を全て実装した仕様カーネルプログラムの通称。

モジュール
リンクによってアドレス解消され、出力されるファイルです。モジュールには動的に装着できないベースモジュールと装着可能なロードモジュールが存在します。

ベースモジュール
ダイナミックローディング機能を実現するための機能(フルセットカーネル、ダイナミックダウンローディング機構等)が実装されたリンクモジュール。ダイナミックローディング機能ではこれらは従来の仕様を用いて作業する事が想定しています。ベースモジュールはダイナミックローディング機能では更新できません。

ロードモジュール
動的に機器に組み込み可能なリンクモジュールです。ロードモジュールの属性は基幹属性とオプション属性のモジュールを作成する事が可能です。基幹属性ロードモジュールはシステムにおいて必要不可欠な機能が実装されることを想定していますので、単独でアンロードは出来ません。また、ダウンロード途中で機器の電源を切ってしまった場合も、新旧どちらかのロードモジュールが存在する事を保証しています。一方、オプション属性のロードモジュールはシステムにおいて必須ではない機能を想定しています。単独でアンロードは可能ですが、ダウンロードキャンセルで新旧とも失われてしまう可能性があります。この場合再ダウンロードが必要となります。
シンボルエクスポート
モジュールはリンク単位です。他のモジュールのシンボル(関数名、変数名)を直接参照することは出来ません。他のモジュールのシンボル参照を可能とする仕組がシンボルエクスポートです。モジュールの開発者ユーザはダイナミックダウンロード機能で拡張されたシンボルを静的で、内部のシンボルを外部モジュールに公開する事が可能です。ベースモジュールのコンフィグレーションとロードモジュールのコンフィグレーションファイルどちらにも記述可能です。ベースモジュールに含まれるカーネルのサービスコールはコンフィギュレータツールがベースモジュールをビルドする際、内部で自動エクスポート処理をしていますので、カーネルサービスコールについて明示的な記述は不要です。

コンフィグレータ
ユーザ定義したシンボルを解釈し、カーネル及びダイナミックローディング機能に構成情報を出力するツールです。コンフィグレータの機能としてベースモジュールをコンフィギュレーションする機能、ロードモジュールをコンフィギュレーションする機能、エクスポートシンボルプロセッサ機能を有しています。コンフィグレータはの一連のモジュールビルド処理のコマンド処理として記述します。
2章 導入編

本章ではダイナミックダウンロード機能の導入方法を説明します。

■■準備

■■■■■■のインストール

まずダウンローディングサーバに■■■■をインストールします。ダウンローディングサーバの実行を確認しているのは■■■■配布セットです。ネットワークダウンロードした■■イメージ、購入パッケージ、書籍の付録等から■■のインストールを実行して下さい。■■イメージは■■■■からダウンロード可能です。
■■■■の詳細なインストール方法は本書では説明を省略しております。■■■■要件は以下の通りです。

ダイナミックダウンロードネットワーク運用を実行するための■■■■システム要件
• 1つ以上の■■■■ネットワークインタフェースがインストールされている事
• ■■■■サーバサービスがインストールされている事

以下に■■■■サーバコンフィグレーション■■■■■■■■例を示します。

リストアート：■■■■サーバコンフィグレーションファイル
ダイナミックダウンロードオフライン運用を実行するためのシステム要件

- またはフライバーが組み込まれていること
- カードリードライターを使用する場合、ホストドライバーとマスストレージクラスドライバーが組み込まれている事

通常のインストールを行うと、これらの機能は組み込まれています。

なおダウンロードサーバオフライン動作の確認は以下の機器、記憶媒体を使用致しました。

カードリードライタ
- 製

記憶媒体
- レキサーメディア社
ツールのダウンロード

ダイナミックダウンロード機関はクロスコンパイラとして以下のツールを使用します。本節ではこのツールのインストール方法を説明します。

<table>
<thead>
<tr>
<th>ツール種類</th>
<th>アーカイブファイル名</th>
<th>バージョン番号</th>
</tr>
</thead>
<tbody>
<tr>
<td>コンパイラ</td>
<td>gcc-core-3.2.tar.gz</td>
<td>3.2</td>
</tr>
<tr>
<td>リンク、アッセンブラ等</td>
<td>binutils-2.13.2.1.tar.gz</td>
<td>2.13.2.1</td>
</tr>
<tr>
<td>ライブラリ</td>
<td>newlib-1.11.0.tar.gz</td>
<td>1.11.0</td>
</tr>
</tbody>
</table>

上記ファイルは以下のウェブサイトから入手可能です。2003年11月現在。

GNU     http://www.gnu.org/
Ring Server http://www.ring.gr.jp/
newlib: Red Hat http://sources.redhat.com/newlib/

上記で示したアーカイブファイルをダウンロードし、tarコマンドを使用して解凍します。
解凍例:

```
$ tar xvf gcc-core-3.2.tar.gz
```

ここでは、アーカイブファイル名の出力ディレクトリのディレクトリをアーカイブ解凍ディレクトリ、実行ファイルの作成ディレクトリを実行ファイルの作成ディレクトリに仮定し説明します。以下で解凍した場合、以下のソースディレクトリがそれぞれ作成されます。

```
/home/someone/gnu-work
/home/someone/gnu-src
```

...
次に実行ファイルビルド作業ディレクトリを作成します。

mkdir /home/someone/gnu-work/binutils-2.13.2.1
mkdir /home/someone/gnu-work/gcc-3.2
mkdir /home/someone/gnu-work/newlib-1.11.0

configure -> make -> install

binutils -> gcc -> newlib

(2) binutils

$ ./../../gnu-src/binutils-2.13.2.1/configure --target=sh-hitachi-elf
$ make
$ make install

(3) gcc

$ ../../gnu-src/gcc-3.3/configure --target=sh-hitachi-elf --prefix=/usr/local --with-gnu-as --with-gnu-ld --with-newlib --with-headers=./../../gnu-src/newlib-1.11.0/newlib/libc/include
$ make
$ make install

上記ではクロス環境ビルドターゲット名として sh-hitachi-elf を、インストール先ディレクトリとして /usr/local 以下を指定しています。$ make install の実行で /usr/local/ への実行ファイルのコピーが行われます。

注：$ make configure を実行するには、./configure ディレクトリの下の書き込み権限が必要です。

(1) ビルド作業

作業ディレクトリ以下のコマンドを実行します。

$ make
$ make install

注：ローダ機能はライブラリ機能を一部使用するため、ローダのライブラリは
この文脈で、協調ビルドを指定しています。この指定では、時先ディレクトリにヘッダファイルがコピーされるため、先ディレクトリ（上記では）への書き込み権限が必要です。

実行でビルトした実行ファイルがインストールディレクトリにコピーされます。この時に先ディレクトリへの書き込み権限が必要です。

ビルト作業

作業ディレクトリで以下のコマンドを実行します。

$ ./configure --target=sh-hitachi-elf --prefix=/usr/local

$ make

$ make install

注：実行するには、ディレクトリへの書き込み権限が必要です。
ダイナミックダウンロードソフトウェアのインストール

以下にダイナミックダウンローディングの配布セットのディレクトリ構造を示します。

リストバインダー：ダイナミックローディング機構ディレクトリ階層
表 ダイヤ ミックローディングディレクトリ構成

<table>
<thead>
<tr>
<th>ディレクトリー</th>
<th>ディレクトリー</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>idle</td>
<td>idl_shell</td>
<td>サーバソフトウェアディレクトリ</td>
</tr>
<tr>
<td></td>
<td>idlsrv</td>
<td>ダウンローディングシェルプログラム</td>
</tr>
<tr>
<td></td>
<td>config.data</td>
<td>ダウンローディングサーバプログラム</td>
</tr>
<tr>
<td></td>
<td>target</td>
<td>サンプルサーバコンフィグレーションデータ</td>
</tr>
<tr>
<td></td>
<td>idl_kernel</td>
<td>サーバソフトウェアディレクトリ</td>
</tr>
<tr>
<td></td>
<td>loader</td>
<td>ダウンローディング機能</td>
</tr>
<tr>
<td></td>
<td>network</td>
<td>ダウンローディングシェル機能</td>
</tr>
<tr>
<td></td>
<td>filesystem</td>
<td>ネットワーク機能</td>
</tr>
<tr>
<td></td>
<td>module</td>
<td>ファイルシステム機能</td>
</tr>
<tr>
<td></td>
<td>script.sample</td>
<td>デモ用モジュール</td>
</tr>
<tr>
<td></td>
<td></td>
<td>サンプルスクリプト</td>
</tr>
</tbody>
</table>
<p><b>リスト(後): フルセットカーネルディレクトリ階層</b></p>

<table>
<thead>
<tr>
<th>項目</th>
</tr>
</thead>
<tbody>
<tr>
<td>フルセットカーネルディレクトリ</td>
</tr>
<tr>
<td>サブディレクトリ</td>
</tr>
<tr>
<td>config</td>
</tr>
<tr>
<td>include</td>
</tr>
<tr>
<td>kernel</td>
</tr>
<tr>
<td>library</td>
</tr>
<tr>
<td>pdic</td>
</tr>
<tr>
<td>systask</td>
</tr>
<tr>
<td>utils</td>
</tr>
<tr>
<td>ms7727cp01</td>
</tr>
</tbody>
</table>

---

*注: 以下では、上記上記の適当なディレクトリに展開します。ここでは、以下のディレクトリとして展開したと仮定し説明を行います。別途フルセットカーネルのディレクトリより、ディレクトリをコピーしてフルセットカーネル機能とダイナミックローディング機能のソースをマージします。*
コンフィギュレータのセットアップ

ビルド方法
ソースを展開した際に出来る、コンフィギュレータディレクトリ下 『idl-opsw』 をディレクトリとして展開します。これにより、実行ファイル 『idlconfig』 が作成されます。次に 『idlconfig』 を実行します。実行ファイル 『idlconfig』が『idl-opsw』 へコピーされます。

注: 『idlconfig』 を実行するには 『idl-opsw』 への書き込み権限が必要です。

起動方法
コンフィグレータの起動は通常 『idlconfig』 の一連のビルド処理中に記述します。『idl-opsw』 へ展開したディレクトリに格納している『idlconfig』を参照してください。
以降にコンフィグレータの起動オプションの説明を行います。

起動書式
- 『idlconfig』 『オプション』

コンフィグレータ全体のオプション
- 『途中経過などを表示します』
- 『起動時のパナーを表示しません』

シンボルローダ
- 『ローダブルモジュールからシンボルを抜きます』
- 『『idlconfig』 の出力ファイルを解析してシンボルを得る』

静的 『パーサ』
- 『入力ファイル名を指定します』
- 『指定したファイルにオブジェクト情報を出力します』
- 『割当の割当順序を指定します』
- 『割当順序は次の 『つの組合せで指定します』
- 『デフォルト』


モジュール名
指定したモジュールを構成する

プレフィックス
プレフィックスを指定します

ファイル生成
ファイル生成モジュール

更新が無くてもファイルを作成します

ベースモジュールコンフィギュレーション

カーネル構成ファイルの名前を指定します

オブジェクトが格納されている場所を指定します

ローダコンフィギュレーション

ローダ構成ファイルを生成します

ローダブルモジュールコンフィギュレーション

対象となるモジュール名を設定します

モジュールを含むオブジェクトファイルを指定します

パス
### 2.3.3 ITRON API

ここではダイナミックダウンロードのコンフィグレータ機能が解釈できる静的APIを説明します。静的APIは，在来型APIの仕様で定義されているものと、ダウンロード用に拡張したものがコンフィグレーションファイルに記述可能です。仕様準拠静的APIとしては以下のAPIを解釈し、カーネル機能及びローダ機能に構成情報を出力します。下記の仕様は、ITRON II DDII仕様書を参照してください。

表 2.3.3.1 仕様 静的API

<table>
<thead>
<tr>
<th>機能分類</th>
<th>定義名</th>
<th>定義クラス</th>
<th>内容</th>
<th>呼出制約</th>
</tr>
</thead>
<tbody>
<tr>
<td>タスク管理</td>
<td></td>
<td></td>
<td>タスクの生成</td>
<td></td>
</tr>
<tr>
<td>タスク例外処理機能</td>
<td></td>
<td></td>
<td>タスク例外処理ルーチンの定義</td>
<td></td>
</tr>
<tr>
<td>同期通信機能</td>
<td></td>
<td></td>
<td>セマフォの生成</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>イベントフラグの生成</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>データキューエ生成</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>メールボックスの生成</td>
<td></td>
</tr>
<tr>
<td>拡張同期通信機能</td>
<td></td>
<td></td>
<td>ミューテックスの生成</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>メッセージバッファの生成</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ランデブポートの生成</td>
<td></td>
</tr>
<tr>
<td>メモリプール管理機能</td>
<td></td>
<td></td>
<td>固定長メモリプールの生成</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>可変帳メモリプールの生成</td>
<td></td>
</tr>
<tr>
<td>時間管理機能</td>
<td></td>
<td></td>
<td>周期ハンドラの生成</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>アラームハンドラの生成</td>
<td></td>
</tr>
<tr>
<td>機能</td>
<td>項目</td>
<td>記号</td>
<td>内容</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>割込み管理機能</td>
<td>割込みハンドラーの定義</td>
<td>〇</td>
<td>〇</td>
<td></td>
</tr>
<tr>
<td></td>
<td>割込みサービスルーチンの追加</td>
<td>〇</td>
<td>〇</td>
<td></td>
</tr>
<tr>
<td></td>
<td>例外ハンドラの定義</td>
<td>〇</td>
<td>〇</td>
<td></td>
</tr>
<tr>
<td>サービスコーレル管理機能</td>
<td>拡張サービスコールの定義</td>
<td>〇</td>
<td>〇</td>
<td></td>
</tr>
<tr>
<td>システム構成管理機能</td>
<td>初期化ルーチンの追加</td>
<td>〇</td>
<td>〇</td>
<td></td>
</tr>
<tr>
<td>ファイル包含機能</td>
<td>外部ファイルのインクルード</td>
<td>〇</td>
<td>〇</td>
<td></td>
</tr>
</tbody>
</table>
拡張した静的API機能

次にダウンロード機能で拡張した、仕様範囲外の静的APIを説明します。

<table>
<thead>
<tr>
<th>機能分類</th>
<th>例名</th>
<th>内容</th>
<th>呼出制約</th>
</tr>
</thead>
<tbody>
<tr>
<td>ベースモジュール</td>
<td></td>
<td>ベースモジュールの属性定義</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>装着可能なロードモジュールの定義</td>
<td>□ □</td>
</tr>
<tr>
<td>ダウンロードモジュール</td>
<td></td>
<td>ロードモジュールの属性定義</td>
<td>□ □</td>
</tr>
<tr>
<td>管理機能</td>
<td></td>
<td>ロードモジュール初期処理の定義</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ロードモジュール終了処理の定義</td>
<td>□ □</td>
</tr>
<tr>
<td>エクスポート機能</td>
<td></td>
<td>シンボルの宣言</td>
<td>□ □</td>
</tr>
<tr>
<td>管理機能</td>
<td></td>
<td>動的生成可能なタスク最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能なセマフォ最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能なイベントフラグ最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能なデータキューマックス数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能なメールボックス最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能な固定長メモリプール最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能な可変長メモリプール最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能なメッセージバッファ最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能なランデブポート最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能なミューテックス最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能な周期ハンドラの最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能なアラームハンドラの最大数</td>
<td>□ □</td>
</tr>
<tr>
<td></td>
<td></td>
<td>動的生成可能な割り込みサービスルーチンの最大数</td>
<td>□ □</td>
</tr>
<tr>
<td>VRID_TSK</td>
<td></td>
<td>ダウンロードの予約</td>
<td>□ □</td>
</tr>
<tr>
<td>VRID_SEM</td>
<td></td>
<td>タスクの予約</td>
<td>□ □</td>
</tr>
<tr>
<td>VRID_MBP</td>
<td></td>
<td>セマフォの予約</td>
<td>□ □</td>
</tr>
<tr>
<td>予約コード</td>
<td>予約内容</td>
<td>イベントフラグ</td>
<td>データキューピー</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>VRID_FLG</td>
<td>予約コード</td>
<td>イベントフラグ</td>
<td>データキューピー</td>
</tr>
</tbody>
</table>
(1) ベースモジュール属性の設定

機能:
ベースモジュールのバージョン番号、属性の設定。

書式:
```
VATR_BSM( module_description, { attribute,  version })
```

<table>
<thead>
<tr>
<th>戸籍項目</th>
<th>定義</th>
</tr>
</thead>
</table>
| attribute | 属性 M
| version | バージョン番号 M

```
VATR_BSM( "uITRON FullSpec. with loader", { TA_NULL, 01.00 });
```

(2) ロードモジュール属性の設定

機能:
ロードモジュールのバージョン番号、属性の設定。

書式:
```
VATR_LDM( Identifier, module_description, { attribute,  version })
```

<table>
<thead>
<tr>
<th>戸籍項目</th>
<th>定義</th>
</tr>
</thead>
</table>
| Identifier | ロードモジュール識別子 M
| module_description | モジュール説明文字列最大 32 文字 M
| attribute | 属性
| TYPE_STANDARD: 基幹属性
| TYPE_OPTION | オプション属性
| version | バージョン番号 M

```
VATR_LDM( Identifier, module_description, { attribute,  version });
```

記述例:
```
"Foo", VATR_BSM("module", { "attr", "1.2.0" });
```

(1) シンボルのエクスポート宣言

機能:
モジュールの内部シンボルを外部モジュールに公開する。

書式:
EXPORT( "シンボル名" );

記述例:
EXPORT( "int  func1( int )" );
EXPORT( "int  val1" );
EXPORT( "ID  mbxid1" );

(2) ロードモジュール初期化処理の登録

機能:
ロードモジュール初期化時に呼び出される、ユーザ初期化ルーチンを登録する。
本定義はロードモジュールのコンフィグレーションファイルに必須である。

書式:
EXPORT_INI( "初期化名" );

記述例:
EXPORT_INI( "初期化処理関数名" );
(ⅰ) ロードモジュール終了処理の登録

機能：
ロードモジュール削除時に呼び出される、ユーザ初期化ルーチンを登録する。
本定義はロードモジュールのコンフィグレーションファイルに必須である。

書式：

ロードモジュール初期化処理関数

記述例：

(ⅱ) ロードモジュールのアタッチ

機能：
ベースモジュールコンフィグレーションにダウンロード可能なロードモジュールを追加する。

書式：

ロードモジュール識別子
ロードモジュールコンフィグレーションファイルワルパス指定

記述例：

パラメータ部

機能:
自動 探番で動的に生成されるカーネルオブジェクトの最大数を定義する。

書式:
VAID_xxx( max_number );
max_number : カーネルオブジェクトの最大数

記述例:
VAID_xxx(123);

機能:
カーネルオブジェクトの 探番 を予約する。
予約された 探番 は、静的 探番 もしくは動的なサービスコールにおいて 探番 を自動割り当て検査、使用しないことを保証する。

書式:
VRID_xxx( reserved_id );
reserved_id : カーネルオブジェクト 探番

記述例:
VRID_xxx(123);
シンボルlekスポート最大数の定義

機能:
ロードモジュールがエクスポートするシンボルの最大数を定義する。

書式:
VNUM_EXP(max_export)

記述例:
VNUM_EXP(50);

ロードモジュールの装着最大数の定義

機能:
ベースモジュールに装着可能なロードモジュールの最大数を定義する。

書式:
VNUM_LDM(max_load_module)

記述例:
VNUM_LDM(50);
11) VDEF_KMB  
heap_size:

12) VATT_TER  
terminate_routine:

システム終了処理の定義

機能:
システム終了時に呼び出すユーザ定義の終了処理を定義する。

記述例:

システム終了処理関数名
コンフィグレータが自動処理する静的 API

下表はロードモジュールがエクスポートしたシンボルを管理するためにコンフィギュレータが内部的に使用する拡張 APIです。ユーザ（開発者）はこの API を使って記述することはありません。

新たなロードモジュールがベースモジュールコンフィグレーションファイルに登録されると、設定されたAPI定義がベースモジュールコンフィギュレーションファイル末尾に自動追加され、設定されたAPI定義が更新されることがあります。

<table>
<thead>
<tr>
<th>定義名</th>
<th>定義内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>グローバル参照シンボル次割り当て（自動生成）</td>
</tr>
<tr>
<td>API</td>
<td>グローバル参照シンボル割り当て定義（自動生成）</td>
</tr>
</tbody>
</table>

書式

API定義の書式は以下のようにします。

<table>
<thead>
<tr>
<th>キー</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>定義名</td>
<td>グローバル参照テーブル 次登録インデックス</td>
</tr>
<tr>
<td>API</td>
<td>エクスポートシンボル名</td>
</tr>
<tr>
<td>定義</td>
<td>登録インデックス</td>
</tr>
<tr>
<td>名前</td>
<td>ロードモジュール識別子</td>
</tr>
</tbody>
</table>
２フレームダウンローディングサーバのセットアップ

２.4 ダウンローディングサーバビルド方法

ダイナミックダウンローディングソースを設定したディレクトリで以下の作業を行い、ダウンロードサーバ実行ファイルの生成、インストールを行います。

- `cd <install_dir>/idl-opsw/server/idlsrv`  ソースディレクトリへ移動
- `make` 実行ファイルの作成
- `make install` 実行ファイルをコピ

この操作は`<install_dir>/idl-opsw/server/idlsrv`への書き込み権限が必要です。

２.4.2 起動方法

ダウンローディングサーバ (idl-opsw) はデーモンのプロセスとして動作します。コマンドラインからタイプする事で起動が可能ですが、自動起動時に実行されるシェルスクリプトに記述する事でデーモン動作も可能となります。ダウンローディングサーバは実行環境設定ファイル、モジュール管理ファイルの読み出し権限、作業ディレクトリロードモジュール作成ディレクトリ、ログ出力ディレクトリへの書き込み権限が必要です。通常`<install_dir>` 権限で実行させます。以降に起動式例、オプション説明をします。

起動式例

- `idlsrv` フォアグランドプロセスとして起動
- `idlsrv &` バックグラウンドプロセスとして起動

オプション説明

- `D` デーモンとして動作します。
- `debug` デバッグ用のオプションです。デバッグメッセージをコンソール出力します。
- `logging` ダウンローディング実行ログを`<install_dir>/var/log/IDL/idlsrv.log`ファイルに出力します。
ファイル名
ダウンローディングサーバーのコンフィグレーションファイル名を指定することで実行環境を切り替えます。
詳細は/etc/IDL/2.0/idlsrv.confを参照します。

起動時にダウンローディングサーバをデーモンとして初期起動させるには、イニシャルシェルスクリプトの1つである、/etc/rc.d/rc.localに以下の1行を追加します。
### ダウンローディングサーバ設定ファイル

ダウンローディングサーバ設定ファイルはダウンローディングサーバ動作を指定するパラメータファイルです。起動引数で格納ディレクトリを指定する事は可能ですが、指定されない場合はデフォルトディレクトリのディレクトリを参照します。本書ではデフォルトディレクトリに格納するものとして説明します。

- ダウンローディングサーバ設定ファイルを作成する。
- ダウンローディング機能ソース配布セット 
  より、設定に含まれる設定を設定ファイル設定にコピーする。
- 動作環境に合わせてエディタで内容を修正する。

<table>
<thead>
<tr>
<th>IDLSRV_IP</th>
<th>192.168.0.207</th>
<th># サーバアドレス設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_IDL_PORT</td>
<td>49512</td>
<td># IDLサーバポート番号設定</td>
</tr>
<tr>
<td>IDLSRV_MAX_SESSION</td>
<td>30</td>
<td># 最大同時セッション数設定(最大2147483647)</td>
</tr>
<tr>
<td>IDLSRV_TIMEOUT</td>
<td>3600</td>
<td># タイムアウト時間の設定[秒]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_TFTP_IP</th>
<th>192.168.0.207</th>
<th># TFTPサーバアドレス設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_TFTP_PORT</td>
<td>69</td>
<td># TFTPポート番号設定</td>
</tr>
<tr>
<td>IDLSRV_TFTP_PATH</td>
<td>/tftpboot</td>
<td># TFTPパス設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_DATABASE_FILE</th>
<th>/etc/IDL/database.txt</th>
<th># データベースファイル設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_AUXINFO_FILE</td>
<td>build.inf</td>
<td># 辅助情報ファイル設定</td>
</tr>
<tr>
<td>IDLSRV_LINKER_OUTPUT</td>
<td>a.out</td>
<td># ルナーライナーオリジナル</td>
</tr>
<tr>
<td>IDLSRV_MODULE_BINARY</td>
<td>module.bin</td>
<td># モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODULE_FILE</td>
<td>module.idl</td>
<td># モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>.module_header</td>
<td># ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>header.bin</td>
<td># ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>header.idl</td>
<td># ヘッダファイル設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MAKE</th>
<th>/usr/bin/make</th>
<th># メインビルド設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_OBJCOPY</td>
<td>/usr/local/bin/objcopy</td>
<td># オブジェクトコピー設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_FD</td>
<td>/dev/fd0</td>
<td># メディア設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MEDIA_FD</th>
<th>/dev/fd0</th>
<th># メディア設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MODEL</th>
<th>モジュール設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MODEL_BINARY</td>
<td>モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODEL_FILE</td>
<td>モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>ヘッダファイル設定</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>IDLSRV_MEDIA_FD</th>
<th>/dev/fd0</th>
<th># メディア設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MODEL</th>
<th>モジュール設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MODEL_BINARY</td>
<td>モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODEL_FILE</td>
<td>モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>ヘッダファイル設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MEDIA_FD</th>
<th>/dev/fd0</th>
<th># メディア設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MODEL</th>
<th>モジュール設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MODEL_BINARY</td>
<td>モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODEL_FILE</td>
<td>モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>ヘッダファイル設定</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>IDLSRV_MEDIA_FD</th>
<th>/dev/fd0</th>
<th># メディア設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MODEL</th>
<th>モジュール設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MODEL_BINARY</td>
<td>モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODEL_FILE</td>
<td>モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>ヘッダファイル設定</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>IDLSRV_MEDIA_FD</th>
<th>/dev/fd0</th>
<th># メディア設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MODEL</th>
<th>モジュール設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MODEL_BINARY</td>
<td>モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODEL_FILE</td>
<td>モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>ヘッダファイル設定</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>IDLSRV_MEDIA_FD</th>
<th>/dev/fd0</th>
<th># メディア設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MODEL</th>
<th>モジュール設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MODEL_BINARY</td>
<td>モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODEL_FILE</td>
<td>モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>ヘッダファイル設定</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>IDLSRV_MEDIA_FD</th>
<th>/dev/fd0</th>
<th># メディア設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MODEL</th>
<th>モジュール設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MODEL_BINARY</td>
<td>モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODEL_FILE</td>
<td>モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>ヘッダファイル設定</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>IDLSRV_MEDIA_FD</th>
<th>/dev/fd0</th>
<th># メディア設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MODEL</th>
<th>モジュール設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MODEL_BINARY</td>
<td>モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODEL_FILE</td>
<td>モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>ヘッダファイル設定</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>IDLSRV_MEDIA_FD</th>
<th>/dev/fd0</th>
<th># メディア設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MEDIA_HDD</td>
<td>/dev/hda</td>
<td># ハードディスク設定</td>
</tr>
<tr>
<td>IDLSRV_MEDIA_USB</td>
<td>/dev/sda1</td>
<td># USBスティック設定</td>
</tr>
<tr>
<td>IDLSRV_MOUNT_POINT</td>
<td>/mnt/IDL</td>
<td># マウントディレクトリ設定</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDLSRV_MODEL</th>
<th>モジュール設定</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MODEL_BINARY</td>
<td>モジュールバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_MODEL_FILE</td>
<td>モジュールファイル設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_SECTION</td>
<td>ヘッダセクション設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_BINARY</td>
<td>ヘッダバイナリ設定</td>
</tr>
<tr>
<td>IDLSRV_HEADER_FILE</td>
<td>ヘッダファイル設定</td>
</tr>
</tbody>
</table>
ダウンロードサーバ関連の設定説明

ダウンロードサーバは TCP/IP ソケットにて、アプリケーションであるダウンロードシェルにサービスを提供します。以下はダウンロードサーバ関連の設定項目です。

<table>
<thead>
<tr>
<th>設定項目</th>
<th>設定内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_HOST</td>
<td>ダウンローディングサーバが動作するホストのアドレスを設定します。</td>
</tr>
<tr>
<td>IDLSRV_PORT</td>
<td>ダウンローディングサーバがダウンローディングシェルにサービスを提供するポート番号を設定します。</td>
</tr>
<tr>
<td>IDLSRV_MAX_SESSION</td>
<td>ダウンローディングサーバの最大セッション数（シェル接続数）を設定します。</td>
</tr>
<tr>
<td>IDLSRV_TIMEOUT</td>
<td>シェル無処理タイムアウト（単位：秒）を設定します。</td>
</tr>
</tbody>
</table>

ファイル転送サーバ関連の設定説明

ダウンロード機能はファイル転送サービスとしてファイル転送サーバサービスを利用します。以下はファイル転送サーバサービスの設定項目です。

<table>
<thead>
<tr>
<th>設定項目</th>
<th>設定内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_HOST</td>
<td>ダウンロードサーバの動作するホストのアドレスを設定します。</td>
</tr>
<tr>
<td>IDLSRV_PORT</td>
<td>ダウンロードサーバがファイル転送サービスを提供するポート番号を設定します。</td>
</tr>
<tr>
<td>IDLSRV_PATH</td>
<td>サーバのルート位置を設定します。</td>
</tr>
</tbody>
</table>
### モジュールビルダ処理 関連の設定

<table>
<thead>
<tr>
<th>設定項目</th>
<th>設定内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_DATABASE_FILE</td>
<td>モジュール管理データベースのファイル名情報をフルパスで設定します。 例: <code>./database.txt</code></td>
</tr>
<tr>
<td>IDLSRV_AUXINFO_FILE</td>
<td>コンフィギュレータが出力するビルド補助情報ファイル名を指定します。本システムでは<code>build.inf</code>と設定します。</td>
</tr>
<tr>
<td>IDLSRV_LINKER_OUTPUT</td>
<td>リンカに出力させる、最終リンクオブジェクト名を指定します。 例: <code>build</code></td>
</tr>
<tr>
<td>IDLSRV_MODULE_FILE</td>
<td>ロードモジュールファイル名(最終ビルドファイル)を指定します。</td>
</tr>
<tr>
<td>IDLSRV_MAKE</td>
<td>ロードモジュールヘッダ情報をセクション名を指定します。</td>
</tr>
<tr>
<td>IDLSRV_MAKEOBJCOPY</td>
<td>リンカ出力ファイルからロードモジュールヘッダ情報を切り出し出力する中間ファイル名を指定します。<code>build.info</code>を特に修正する必要はありません。</td>
</tr>
<tr>
<td>IDLSRV_MODULE_FILEOBJCOPY</td>
<td>ロードモジュールヘッダ情報ファイル名(最終ビルドファイル)を指定します。<code>build.info</code>を特に修正する必要はありません。</td>
</tr>
</tbody>
</table>

### モジュールビルダ処理で起動する外部実行ファイル

<table>
<thead>
<tr>
<th>設定項目</th>
<th>設定内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLSRV_MAKEEXEC</td>
<td>コマンド実行ファイルをフルパスで指定します。 例: <code>./make</code></td>
</tr>
<tr>
<td>IDLSRV_MAKEOBJCOPYEXEC</td>
<td>コマンド実行ファイルに含まれる、コマンド実行ファイルをフルパスで指定します。<code>build.info</code>はダウンロードサーバがサポートする全てのターゲットに対して操作可能なマルチターゲットビルトされたものである必要があります。 例: <code>./make</code></td>
</tr>
</tbody>
</table>
オフラインモードで使用するデバイス情報

<table>
<thead>
<tr>
<th>設定項目</th>
<th>設定内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>フロッピディスクデバイス例</td>
<td>常に存在</td>
</tr>
<tr>
<td>ハードディスクデバイス例</td>
<td>ハードディスク</td>
</tr>
<tr>
<td>ストレージデバイス例</td>
<td>USB</td>
</tr>
<tr>
<td>デバイスマウントディレクトリ</td>
<td>/dev/fd0</td>
</tr>
<tr>
<td>オフライン操作時のデフォルトメディア</td>
<td>フロッピディスク</td>
</tr>
<tr>
<td></td>
<td>ハードディスク</td>
</tr>
<tr>
<td></td>
<td>ストレージデバイス</td>
</tr>
</tbody>
</table>

サーバ作業ディレクトリの作成

ダウンローディングサーバを作動させる前に以下のディレクトリを手作業で作成しなくてはなりません。実行環境に合わせて以下のディレクトリを作成します。

<table>
<thead>
<tr>
<th>内容</th>
<th>本書で対象となっているディレクトリ名・ファイル名</th>
<th>特記</th>
</tr>
</thead>
<tbody>
<tr>
<td>ダウンローディングサーバ</td>
<td>ディレクトリ名</td>
<td>起動パラメータで変更可能</td>
</tr>
<tr>
<td>実行環境設定ファイル</td>
<td>ファイル名</td>
<td></td>
</tr>
<tr>
<td>格納ディレクトリ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>モジュール管理データベース</td>
<td>ディレクトリ名</td>
<td>実行環境設定ファイルで変更可能</td>
</tr>
<tr>
<td>格納ディレクトリ</td>
<td>ファイル名</td>
<td></td>
</tr>
<tr>
<td>モジュールビルド作業ディレクトリ</td>
<td>ディレクトリ名</td>
<td>実行環境設定ファイルで変更可能。時期はサービスのディレクトリ階層を設定する事。</td>
</tr>
<tr>
<td>ログ出力ディレクトリ</td>
<td>ディレクトリ名</td>
<td>変更は出来ない</td>
</tr>
</tbody>
</table>
モジュール管理データベースファイル

モジュール管理データベースはダウンロードサーバが接続するターゲットシステムの情報、作成したダウンロードモジュールの情報を登録するテキストファイルです。設定フォーマットはカンマ区切り文字とする形式となっています。データはコメント記述行を表します。
本節では、モジュール管理データベースに登録するデータ項目の説明を行います。

ターゲットシステム登録情報

<table>
<thead>
<tr>
<th>T,3,1,0x00000000:0x007fffff,0x0c000000:0x0dffffff,0x00100000:0x007effff,0x000010000,0xe0000000,1,0x00000000:0x007fffff,0x00010000</th>
</tr>
</thead>
</table>

登録行書式：

- 事務コード
- ターゲット ID
- エンディアンタイプ
- ROM-START:ROM-END
- RAM-START:RAM-END
- LDMROM-START:LDMROM-END
- ROUND-SIZE
- ADDRESS-MASK
- FlashBankNum
- Flash-Bank-Info
- Erase-Block-Size
<table>
<thead>
<tr>
<th>設定項目</th>
<th>設定内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetID</td>
<td>ターゲットシステムを一意に識別する任意のID番号を設定します。</td>
</tr>
<tr>
<td>EndianType</td>
<td>ターゲットシステムのエンディアンタイプ（バイプレインド）を設定します。 設定値 1:リトルエンディアン</td>
</tr>
<tr>
<td>ROM-Area-Info</td>
<td>全体のアドレス範囲を開始アドレス〜終了アドレスの書式で設定します。</td>
</tr>
<tr>
<td>RAM-Area-Info</td>
<td>全体のアドレス範囲を開始アドレス〜終了アドレスの書式で設定します。</td>
</tr>
<tr>
<td>LDROM-Area-Info</td>
<td>ROM-Area-Infoの中でのロードモジュールを割り付ける可能なアドレス範囲を開始アドレス〜終了アドレスの書式で設定します。</td>
</tr>
</tbody>
</table>
| ADDRESS_MASK | CPU設定例  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z  
| CPU設定例 | SH3: 0xe0000000  M68K: 0x0 |
| Flash-Bank-Info | Flash-Areaの情報数を設定します。 |
| Flash-Bank-Info | 消去バンク定義情報の情報数を設定します。 |
| Flash-Bank-Info | 消去バンク定義情報を開始アドレス〜終了アドレス〜消去ブロックサイズの書式で設定します。 |
### ロードモジュール登録情報

<table>
<thead>
<tr>
<th>号</th>
<th>内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>ロードモジュールの定義</td>
</tr>
<tr>
<td>(2)</td>
<td>定義行書式</td>
</tr>
<tr>
<td>(3)</td>
<td>定義項目説明</td>
</tr>
</tbody>
</table>

#### 定義行書式

- モジュール定義行キーワードタグ
- モジュール
- 使用可能機器
- ロード可能ベースモジュールバージョン範囲
- ロードモジュールバージョン
- モジュール属性
- モジュール説明
- システムコンフィグレーション ディレクトリ
- モジュールコンフィグレーション ディレクトリ

#### 登録行書式

- [WEB](standard module)
- モジュール名
- バージョン
- タグ
- ディレクトリ

<table>
<thead>
<tr>
<th>号</th>
<th>内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>IPA web server (OPTION)</td>
</tr>
<tr>
<td>3</td>
<td>IPA web server (OPTION)</td>
</tr>
<tr>
<td>設定項目</td>
<td>設定内容</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>コンフィギュレータが割り付けたロードモジュール番号を設定します。</td>
<td></td>
</tr>
<tr>
<td>登録するロードモジュールをダウンロード可能なシステムをターゲットシステム登録情報で設定したターゲット番号で設定します。</td>
<td></td>
</tr>
<tr>
<td>登録するロードモジュールが動作可能なベースモジュールバージョン番号範囲を以下の桁数桁の数値書式で設定します。開始バージョン 終了バージョン</td>
<td></td>
</tr>
<tr>
<td>登録するロードモジュールのバージョン番号を設定します。本バージョン番号はロードモジュールコンフィギュレーションファイルに設定したバージョン番号と一致させる必要があります。</td>
<td></td>
</tr>
<tr>
<td>中でロードモジュールを割り付けるアドレス範囲を開始アドレス・終了アドレスの書式で設定します。</td>
<td></td>
</tr>
<tr>
<td>ロードモジュールの領域を割り付ける際の、アドレス境界整合値を指定します。</td>
<td></td>
</tr>
<tr>
<td>によっては、論理的なアドレス空間を識別する情報を論理アドレスに持つものの存在します。ここでは論理アドレスを物理アドレス情報へ変換するためのマスク情報を設定します。論理アドレス → 物理アドレスとなるのでオール設定を設定します。設定例 設定例</td>
<td></td>
</tr>
<tr>
<td>消去バンク定義情報の情報数 中の消去バンク定義情報の情報数を設定します。</td>
<td></td>
</tr>
<tr>
<td>消去バンク定義情報を開始アドレス・終了アドレスを消去ブロックサイズの書式で設定します。</td>
<td></td>
</tr>
</tbody>
</table>
### ターゲット機器 ネットワークノード登録情報

#### 行書式

<table>
<thead>
<tr>
<th>設定項目</th>
<th>設定内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>ノード名</td>
<td>ターゲットノードの識別名称を記述します。</td>
</tr>
<tr>
<td>ノードID</td>
<td>ノードのシステム種別をターゲットにで設定します。</td>
</tr>
<tr>
<td>アドレス</td>
<td>ノードのアドレス情報設定します。</td>
</tr>
<tr>
<td>ポート番号</td>
<td>ターゲットで動作するロータ機能のポート番号を指定します。</td>
</tr>
</tbody>
</table>
### ターゲット機器 ネットワークノードグループ登録情報

#### 同じシステムタイプを持ったターゲット機器の総称を登録する

**書式:**

`G,<group-name>,<node-name1>,<node-name2>`

**登録行書式:**

`G,DL-DemoSystem,TE1,TE2`

### 設定項目

<table>
<thead>
<tr>
<th>設定項目</th>
<th>設定内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>グループ識別名</td>
<td>グループ識別名称を記述します。</td>
</tr>
<tr>
<td>グループ所属ノード名リスト</td>
<td>グループに所属するノード名リストを記述します。</td>
</tr>
</tbody>
</table>
サーバダウンロードシェル

ダウンロードシェルプログラムのビルト方法

ダイナミックダウンローディングソースをディレクトリで以下の作業を行い、ダウンロードシェル実行ファイルの生成、インストールを行います。

1. $ cd <install_dir>/idl-opsw/server/idl_shell
2. $ make
3. $ make install

上の操作は、$USER$への書き込み権限が必要です。
起動方法

起動パラメータ

シールドプログラム起動時にダウンローディングサーバに接続します。起動パラメータは下表の通りです。オプションを省略した場合はローカルホスト指定になります。（ポートはdefault固定）ダウンローディングサーバ接続後は、プロンプト(>は)が表示されてコマンド入力待ちになりますので、コマンド名と必要なパラメータを入力し、「Enter」キーを押下すればそのコマンドが実行されます。（コマンド名は大文字で指定することも出来ます）

コマンドでパラメータ

<table>
<thead>
<tr>
<th>起動パラメータ</th>
<th>メッセージの日本語表示</th>
</tr>
</thead>
<tbody>
<tr>
<td>さ</td>
<td>メッセージの日本語表示</td>
</tr>
<tr>
<td>と</td>
<td>メッセージの日本語表示</td>
</tr>
<tr>
<td>さ</td>
<td>メッセージの日本語表示</td>
</tr>
</tbody>
</table>

使用例

メッセージ

接続成功：

server接続エラー

接続失敗：

サーバ接続エラー

備考

接続に失敗した場合はシールドプログラムは終了します。
### シェルコマンド一覧

<table>
<thead>
<tr>
<th>コマンド名</th>
<th>省略形</th>
<th>コマンド説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect</td>
<td>c</td>
<td>ターゲットへ接続</td>
</tr>
<tr>
<td>status</td>
<td>sta</td>
<td>サーバに登録されているロードモジュールの一覧表示</td>
</tr>
<tr>
<td>unload</td>
<td>uld</td>
<td>ターゲットに実装されているロードモジュールの一覧表示</td>
</tr>
<tr>
<td>load</td>
<td>ld</td>
<td>ターゲットに実装されているロードモジュールの削除</td>
</tr>
<tr>
<td>target</td>
<td>ta</td>
<td>ターゲットにモジュールをロードする</td>
</tr>
<tr>
<td>group</td>
<td>gr</td>
<td>接続中のターゲット一覧表示</td>
</tr>
<tr>
<td>member</td>
<td>mem</td>
<td>データベースに登録されているグループの一覧表示</td>
</tr>
<tr>
<td>target</td>
<td>tg</td>
<td>グループに所属しているターゲットの一覧表示</td>
</tr>
<tr>
<td>disconnect</td>
<td>disc</td>
<td>接続可能ターゲットの一覧表示</td>
</tr>
<tr>
<td>reload</td>
<td>rel</td>
<td>ターゲットの切断</td>
</tr>
<tr>
<td>rebuild</td>
<td>b</td>
<td>ターゲットにモジュールを再ロードする</td>
</tr>
<tr>
<td>help</td>
<td>h</td>
<td>コマンドの使用法の説明</td>
</tr>
<tr>
<td>quit</td>
<td>q</td>
<td>シェルプログラム終了</td>
</tr>
<tr>
<td>offline</td>
<td></td>
<td>オフライン運用時のモジュールビルト指定コマンド</td>
</tr>
</tbody>
</table>
シェルコマンドリファレンス

1. おおまかな:ターゲットの接続

ダウンローディングサーバに対して指定したターゲットへの接続を指示します。接続先としてグループ名を指定することで、そのグループに所属しているターゲット全てに対して一括接続を行います。グループ名を指定する場合は -gオプションを指定してください。

<table>
<thead>
<tr>
<th>コマンド名（省略形）</th>
<th>コマンド書式</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>コンポーネント</td>
</tr>
<tr>
<td>パラメータ</td>
<td>モジュール管理</td>
</tr>
<tr>
<td>使用例：IP指定</td>
<td></td>
</tr>
<tr>
<td>ターゲット名指定</td>
<td></td>
</tr>
<tr>
<td>グループ名指定</td>
<td></td>
</tr>
<tr>
<td>正常時メッセージ</td>
<td></td>
</tr>
<tr>
<td>エラーメッセージ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

エラーメッセージ（起動オプションで--が指定された時は日本語表示）

|                  |                   |                                                                                  |
|                  |                   |                                                                                  |

備考

接続が成功した場合、プロンプトがidea>になる。
2. モジュールリストの取得

モジュール管理で登録されているロードモジュールのリストを取得し、一覧表示します。

<table>
<thead>
<tr>
<th>コマンド名 (省略形)</th>
<th>mdlist</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド書式</td>
<td>mdlist</td>
</tr>
<tr>
<td>パラメータ</td>
<td>なし</td>
</tr>
<tr>
<td>使用例</td>
<td>mdlist</td>
</tr>
<tr>
<td>正常時表示例</td>
<td>mdlist</td>
</tr>
<tr>
<td>エラーメッセージ (起動オプションで)</td>
<td>mdlist</td>
</tr>
<tr>
<td>エラーメッセージ (日本語で)</td>
<td>mdlist</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>備考</th>
<th>モジュールリスト取得エラー</th>
<th>モジュールリスト取得が失敗した</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>サーバ内で致命的な障害発生 (サーバを直ちに切断する)</td>
<td></td>
</tr>
</tbody>
</table>

正常時に表示されるモジュール内容は以下の通りです。

モジュール ID  モジュールバージョン 属性  モジュール名
### 3. モジュール状態問い合わせ

ターゲットに装着されているロードモジュールのうち、指定したモジュール-IDのロードモジュール情報を取得し、表示します。モジュール-IDを省略した場合、または、ワイルドカード(*)を指定した場合は装着されている全てのロードモジュール情報を一覧表示します。このコマンドは接続されている全てのターゲットに対して、ロードモジュールの状態を問い合わせします。

<table>
<thead>
<tr>
<th>コマンド名 (省略形)</th>
<th>コマンド書式</th>
<th>パラメータ</th>
<th>使用例</th>
<th>ワイルドカード指定</th>
</tr>
</thead>
</table>
| モジュール状態問い合わせ | モジュール状態問い合わせ | モジュール-ID | ワイルドカード (*** ***)

**正常時表示例**

<table>
<thead>
<tr>
<th>エラーメッセージ (起動オプションで西が指定された時には日本語表示)</th>
<th>パラメータが不正</th>
<th>パラメータが指定されていない</th>
</tr>
</thead>
<tbody>
<tr>
<td>グループ所属ターゲット取得エラー</td>
<td>指定グループ名がデータベースに登録されていない</td>
<td></td>
</tr>
<tr>
<td>ターゲットリストなし</td>
<td>指定グループのターゲットリストが見つからなかった。</td>
<td></td>
</tr>
<tr>
<td>ターゲット名不正</td>
<td>指定ターゲット名がデータベースに登録されていない</td>
<td></td>
</tr>
<tr>
<td>接続済みターゲット</td>
<td>指定したターゲット名または、IPのターゲットと既に接続されている。</td>
<td></td>
</tr>
</tbody>
</table>

**備考**

正常時に表示されるモジュール内容は以下の通りです。
モジュール ID モジュールバージョン 属性 モジュール名
4. モジュールの削除

ターゲットシステムに装着されているロードモジュールのうち、指定したモジュールIDのロードモジュール削除コマンドを実行します。このコマンドは接続されている全てのターゲットに対して、指定IDのロードモジュールを削除します。

<table>
<thead>
<tr>
<th>コマンド名（省略形）</th>
<th>モジュール削除</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド書式</td>
<td><code>unload</code></td>
</tr>
<tr>
<td>パラメータ</td>
<td>モジュールID</td>
</tr>
<tr>
<td>使用例</td>
<td>モジュール削除成功</td>
</tr>
<tr>
<td>正常時表示例</td>
<td>レポートを出力</td>
</tr>
<tr>
<td>正常時メッセージ</td>
<td>指定したIDのモジュールを削除した。</td>
</tr>
<tr>
<td>エラーメッセージ</td>
<td>パラメータが指定されていない</td>
</tr>
<tr>
<td>(起動オプションで@が指定された時は日本語表示)</td>
<td>パラメータが不正</td>
</tr>
<tr>
<td></td>
<td>ターゲット未接続</td>
</tr>
<tr>
<td></td>
<td>ターゲットエラー</td>
</tr>
<tr>
<td></td>
<td>アンロードエラー</td>
</tr>
</tbody>
</table>
| 備考             | 本コマンドを実行する前に、ターゲットと接続しておく必要があります。
5. トレース: モジュール追加、更新

指定したIDとバージョンのロードモジュールを、ターゲットシステムのメモリマップに合わせて再ビルド（リンク）し、ターゲットシステムにロードします。ロード中はキーによるキャンセルが可能です。このコマンドは接続されている全てのターゲットに対して、指定IDのロードモジュールをロードします。

<table>
<thead>
<tr>
<th>コマンド名（省略形）</th>
<th>コマンド書式</th>
</tr>
</thead>
<tbody>
<tr>
<td>トレース</td>
<td>ID 希望バージョン</td>
</tr>
<tr>
<td>パラメータ1</td>
<td>モジュールID</td>
</tr>
<tr>
<td>パラメータ2</td>
<td>モジュールバージョン</td>
</tr>
<tr>
<td>使用例</td>
<td></td>
</tr>
<tr>
<td>正常時表示例</td>
<td></td>
</tr>
<tr>
<td>正常時メッセージ</td>
<td>モジュールローディング</td>
</tr>
<tr>
<td>エラーメッセージ (起動オプションで -e が指定された時は日本語表示）</td>
<td>パラメータが不正</td>
</tr>
<tr>
<td>ターゲット未接続</td>
<td>パラメータが不正</td>
</tr>
<tr>
<td>ターゲットエラー</td>
<td>ターゲット未接続</td>
</tr>
<tr>
<td>ローディングエラー</td>
<td>ターゲット未接続</td>
</tr>
<tr>
<td>ローディングキャンセル</td>
<td>ローディングエラー</td>
</tr>
<tr>
<td>ビルドエラー</td>
<td>ローディングキャンセル</td>
</tr>
<tr>
<td>備考</td>
<td>本コマンドを実行する前に、ターゲットと接続しておく必要があります。</td>
</tr>
</tbody>
</table>
6. 再ロード: モジュールの再ロード(デバッグ支援コマンド)

指定したセッションIDのターゲットシステムへロードしたビルド済みロードモジュールを、現在接続されている全てのターゲットシステムへ再ロードします。

注

本コマンドはロードモジュールデバッグする際、再ビルドを行いたくないケースを支援するコマンドです。ターゲットに再ダウンロードして正常に動作するかはオペレータ責任となります。実運用時の使用は避けて下さい。

<table>
<thead>
<tr>
<th>コマンド名（省略）</th>
<th>パラメータ数（個）</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド書式</td>
<td>パラメータ1→パラメータ2→パラメータ3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>パラメータ1</th>
<th>パラメータ2</th>
<th>パラメータ3</th>
</tr>
</thead>
<tbody>
<tr>
<td>セッションID</td>
<td>モジュールID</td>
<td>モジュールバージョン</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>使用例</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>正常時表示例</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>正常時メッセージ</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>モジュールローディング</td>
<td></td>
</tr>
<tr>
<td>モジュールロード成功</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>エラーメッセージ</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>パラメータが指定されていない</td>
<td></td>
</tr>
<tr>
<td>バラメータが不正</td>
<td></td>
</tr>
<tr>
<td>ターゲット未接続</td>
<td></td>
</tr>
<tr>
<td>ターゲットエラー</td>
<td></td>
</tr>
<tr>
<td>ターゲットが応答しない（タイムアウト）</td>
<td></td>
</tr>
<tr>
<td>ローディングエラー</td>
<td></td>
</tr>
<tr>
<td>モジュールロード失敗</td>
<td></td>
</tr>
<tr>
<td>ローディングキャンセル</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>備考</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>本コマンドを実行する前に、ターゲットを接続しておく必要があります。</td>
<td></td>
</tr>
</tbody>
</table>
7. 接続中のターゲットを表示

現在接続されているターゲット(IPアドレス)の一覧を表示します。

<table>
<thead>
<tr>
<th>コマンド名 (省略形)</th>
<th>コマンド形式</th>
<th>パラメータ</th>
<th>使用例</th>
<th>正常時表示例</th>
<th>出力メッセージ</th>
<th>備考</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect Target</td>
<td></td>
<td>なし</td>
<td></td>
<td></td>
<td></td>
<td>本コマンドはサーバへの問い合わせは行いません。</td>
</tr>
</tbody>
</table>

8. 接続可能ターゲット表示

ローディングサーバ内のデータベースから接続可能なターゲットシステムのリストを取得し、一覧を示します。

<table>
<thead>
<tr>
<th>コマンド名 (省略形)</th>
<th>コマンド形式</th>
<th>パラメータ</th>
<th>使用例</th>
<th>正常時表示例</th>
<th>出力メッセージ</th>
<th>備考</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tglist</td>
<td></td>
<td>なし</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9. グループのリスト表示
ローディングサーバ内のデータベースに登録されているグループのリストを取得し、一覧を表示します。

<table>
<thead>
<tr>
<th>コマンド名 (省略形)</th>
<th>$group$</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド書式</td>
<td>$group$</td>
</tr>
<tr>
<td>パラメータ</td>
<td>なし</td>
</tr>
<tr>
<td>使用例</td>
<td>$group$</td>
</tr>
<tr>
<td>正常時表示例</td>
<td>なし</td>
</tr>
<tr>
<td>出力メッセージ</td>
<td>なし</td>
</tr>
<tr>
<td>備考</td>
<td>なし</td>
</tr>
</tbody>
</table>

10. グループに所属するターゲットのリスト表示
指定したグループがローディングサーバ内のデータベースに登録されているか検索し、登録されていた場合は、そのグループに所属しているターゲット（IPアドレス）のリストを取得し、一覧表を示します。

<table>
<thead>
<tr>
<th>コマンド名 (省略形)</th>
<th>$mem$</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド書式</td>
<td>$mem$</td>
</tr>
<tr>
<td>パラメータ</td>
<td>グループ名</td>
</tr>
<tr>
<td>使用例</td>
<td>グループ名</td>
</tr>
<tr>
<td>正常時表示例</td>
<td>なし</td>
</tr>
<tr>
<td>エラーメッセージ（起動オプションで指定された時は日本語表示）</td>
<td>バラメータが指定されていない</td>
</tr>
<tr>
<td></td>
<td>パラメータが不正</td>
</tr>
<tr>
<td></td>
<td>ターゲットリスト取得失敗</td>
</tr>
<tr>
<td>備考</td>
<td>なし</td>
</tr>
</tbody>
</table>

Parameter error
11．ターゲットの切断

指定したターゲット名または、IPのターゲットを切断します。またグループ名を指定することで、そのグループに所属しているターゲット全てに対して一括切断を行います。グループ名を指定する場合は -g オプションを指定してください。

<table>
<thead>
<tr>
<th>コマンド名（省略形）</th>
<th>disconnect(ディスコン)</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド書式</td>
<td>disconnect [パラメータ]</td>
</tr>
<tr>
<td>パラメータ</td>
<td>ターゲットのIPアドレスとポート番号</td>
</tr>
<tr>
<td></td>
<td>ターゲット名</td>
</tr>
<tr>
<td></td>
<td>グループ名</td>
</tr>
<tr>
<td>使用例: IP指定</td>
<td>disconnect 192.168.0.205:9876</td>
</tr>
<tr>
<td>ターゲット名指定</td>
<td>disconnect TE5</td>
</tr>
<tr>
<td>グループ名指定</td>
<td>disconnect ?gDemoSystem</td>
</tr>
<tr>
<td>接続中の全ターゲット指定</td>
<td>disconnect *</td>
</tr>
<tr>
<td>（省略可）</td>
<td>disconnect</td>
</tr>
<tr>
<td>出力メッセージ</td>
<td>ダイアログを表示します</td>
</tr>
<tr>
<td>備考</td>
<td>全てのターゲットを切断した場合は、プロンプトが “disconnect” になります。</td>
</tr>
</tbody>
</table>

12．サーバーの切断（終了）

シェルプログラムを終了します。

<table>
<thead>
<tr>
<th>コマンド名（省略形）</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>パラメータ</td>
<td>なし</td>
</tr>
<tr>
<td>出力メッセージ</td>
<td>なし</td>
</tr>
<tr>
<td>使用例</td>
<td>q</td>
</tr>
</tbody>
</table>
13. オフライン運用時のモジュールビルド指定

ビルド要求パケットファイルが存在する、ダウンローディングサーバのデフォルトデバイス（FD、HDD、USBストレージデバイス）を指定し、ビルド要求パケットファイルに記述されているモジュールIDとモジュールバージョンのロードモジュールをビルドします。ビルド実行後、ビルド応答（結果）を応答パケットファイルに出力し、生成したロードモジュールを（モジュールヘッダパケット）、（モジュール本体パケット）ファイルとして、メディアに書き込みます。

<table>
<thead>
<tr>
<th>コマンド名（省略形）</th>
<th>コマンド書式</th>
</tr>
</thead>
<tbody>
<tr>
<td>パラメータ</td>
<td>パケットファイル接頭メディア</td>
</tr>
<tr>
<td>使用例</td>
<td></td>
</tr>
<tr>
<td>正常時表示例</td>
<td></td>
</tr>
</tbody>
</table>

エラーメッセージ（起動オプションでが指定された時は日本語表示）

| マウントエラー（不正メディア） | 指定メディアのマウントが出来ない |
| ファイル作成失敗 | パケットファイルの作成が出来ない |
| ファイルオープンエラー | 要求パケットが存在しない、またはファイルリードエラーが発生 |
| 不正なビルド要求 | 要求パケットの書式が不正 |

備考 | コマンドはオフライン運用専用のコマンドです。 |
ターゲットダウンロードシェル

起動方法

ネットワーククライアントデマンド運用またはオフライン運用でダウンロード機能を実行するには機器側にダウンロードシェル機能が必要となります。ベースモジュールにクライアントデマンドもしくはオフライン運用を選択しビルドした場合、以下のようにダウンロードシェル機能が自動起動されます。

**** TOPPERS/IDL TAR GET SHELL ****

Version: $Revision: 1.14 $
シェルコマンドリファレンス

2.6.2 ダウンロードサーバーの接続

<table>
<thead>
<tr>
<th>コマンド説明</th>
<th>ローディングサーバに接続を行う。</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド名</td>
<td>$connect $IP :ポート $port</td>
</tr>
<tr>
<td>省略形</td>
<td>$connect $IP :ポート $port</td>
</tr>
<tr>
<td>パラメータ</td>
<td>ローディングサーバのIPアドレスとポート番号</td>
</tr>
<tr>
<td></td>
<td>パラメータを省略した場合はローカルサーバへの接続を行う</td>
</tr>
<tr>
<td>書式</td>
<td>$connect $IP :ポート $port</td>
</tr>
<tr>
<td>使用例</td>
<td>$connect 192.168.0.205 :9876</td>
</tr>
<tr>
<td>オフライン運用時</td>
<td>無効</td>
</tr>
<tr>
<td>備考</td>
<td>接続後のプロンプト表示を変える</td>
</tr>
<tr>
<td></td>
<td>[ ] [ ] [ ]</td>
</tr>
<tr>
<td></td>
<td>本コマンドは、オフライン運用では使用しない。</td>
</tr>
<tr>
<td></td>
<td>（ターゲットの起動後、サーバと接続している状態になる）</td>
</tr>
</tbody>
</table>

メッセージ出力

エラー：

接続中エラー」
指定したIPのサーバと既に接続されている

交換エラー」
指定したIPのサーバが存在しない

サーバ接続エラー」
サーバとの接続に失敗した

接続成功：

「接続成功」[IP :ポート] : 既に接続されています。
モジュールリストの取得

<table>
<thead>
<tr>
<th>コマンド説明</th>
<th>ローディングサーバに登録されているロードモジュールのリストを取得し、一覧表示する。</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド名</td>
<td>コマンド名</td>
</tr>
<tr>
<td>省略形</td>
<td>省略形</td>
</tr>
<tr>
<td>パラメータ</td>
<td>なし</td>
</tr>
<tr>
<td>使用例</td>
<td>使用例</td>
</tr>
<tr>
<td>オフライン運用時</td>
<td>無効</td>
</tr>
</tbody>
</table>

メッセージ出力

エラー：
「モジュールリスト取得エラー」

取得成功：モジュール一覧表示

モジュール一覧表示

.".
".".
".".
## モジュール状態問い合わせ

<table>
<thead>
<tr>
<th>コマンド説明</th>
<th>ターゲットに実装されているロードモジュールのうち、指定したモジュール ID のロードモジュール情報を表示する。モジュール ID に 0* または 0* を指定した場合、実装されている全てのロードモジュール情報を一覧表示する。</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド名</td>
<td>モジュール問い合わせ</td>
</tr>
<tr>
<td>省略形</td>
<td>コマンド名</td>
</tr>
<tr>
<td>パラメータ</td>
<td>モジュール ID</td>
</tr>
<tr>
<td>書式</td>
<td>（モジュール ID）</td>
</tr>
<tr>
<td>使用例</td>
<td>モジュール ID を指定して実行</td>
</tr>
<tr>
<td>オフライン運用時</td>
<td>有効</td>
</tr>
</tbody>
</table>

機能概略：
ターゲットのモジュールの状態情報を返す。
引数としてモジュール ID を指定した場合、そのロードモジュールの状態を表示する。

### メッセージ出力

**エラー：**
- 状態取得エラー
- モジュール ID が無効

**問い合わせ成功 状態表示**

<table>
<thead>
<tr>
<th>＾</th>
<th>～</th>
<th>～</th>
<th>～</th>
<th>～</th>
<th>～</th>
<th>～</th>
<th>～</th>
</tr>
</thead>
<tbody>
<tr>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
</tr>
<tr>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
<td>～</td>
</tr>
</tbody>
</table>

.
# モジュールの削除

<table>
<thead>
<tr>
<th>コマンド説明</th>
<th>ターゲットに実装されているロードモジュールのうち、指定したモジュール ID のロードモジュール削除する。</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド名</td>
<td>Unload</td>
</tr>
<tr>
<td>省略形</td>
<td>uld</td>
</tr>
<tr>
<td>パラメータ</td>
<td>モジュール ID</td>
</tr>
<tr>
<td>書式</td>
<td>Unload (モジュール ID)</td>
</tr>
<tr>
<td>使用例</td>
<td>Unload (モジュール ID)</td>
</tr>
<tr>
<td>オフライン運用時</td>
<td>有効</td>
</tr>
</tbody>
</table>

機能概略：
指定した ID のロードモジュールをターゲットから削除する。

## メッセージ出力

### エラー:

- 「アンロードエラー」
- 「モジュール ID が無効」

### 削除成功:

- 「モジュールアンロード」
モジュールビルド（オフライン運用専用コマンド）

<table>
<thead>
<tr>
<th>コマンド説明</th>
<th>ロードモジュールビルド指示情報を『に出力する。</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド名</td>
<td>省略形</td>
</tr>
<tr>
<td>パラメータ</td>
<td>モジュールID</td>
</tr>
<tr>
<td>書式</td>
<td>モジュールバージョン</td>
</tr>
<tr>
<td>使用例</td>
<td></td>
</tr>
<tr>
<td>オフライン運用時</td>
<td>オフライン運用時のみ有効</td>
</tr>
<tr>
<td>備考</td>
<td>事前にCFをターゲットに挿入しておく必要がある</td>
</tr>
</tbody>
</table>

メッセージ出力
エラー：CFカードがセットされていないなど、「ファイルオープンエラー」
ファイル作成成功：「ファイル作成」
ロードモジュール追加、更新

<table>
<thead>
<tr>
<th>コマンド説明</th>
<th>ロードモジュールをターゲットシステムにローディングする。 （オフライン運用時には、コマンドに収められたモジュールをロードする）</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド名</td>
<td>load</td>
</tr>
<tr>
<td>省略形</td>
<td>l</td>
</tr>
<tr>
<td>パラメータ</td>
<td>モジュールID</td>
</tr>
<tr>
<td></td>
<td>バージョン</td>
</tr>
<tr>
<td>書式</td>
<td>load &lt;module_id&gt; &lt;version_no&gt;</td>
</tr>
<tr>
<td>使用例</td>
<td>有効</td>
</tr>
</tbody>
</table>

メッセージ出力
エラー：
「ロードエラー」

ロード成功：
「モジュールロード」
### サーバーの切断（終了）

<table>
<thead>
<tr>
<th>コマンド説明</th>
<th>サーバーを切断します</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド名</td>
<td>disconnect</td>
</tr>
<tr>
<td>省略形</td>
<td>disc</td>
</tr>
<tr>
<td>パラメータ</td>
<td>なし</td>
</tr>
<tr>
<td>使用例</td>
<td></td>
</tr>
<tr>
<td>オフライン運用時</td>
<td>無効（オフライン運用時には、サーバとの通信を行わないので、本コマンドは使用しない）</td>
</tr>
</tbody>
</table>

### メッセージ出力

「サーバーを切断しました」
### ダウンローディングターゲットのセットアップ

#### ターゲットシステムのセットアップ

ダウンローディング機構のターゲットハードウェアとして、デバッグボードが必要となります。ハートウェアボード開発セットに同梱されている取り扱い説明書に従って、ハードウェアを組み立てます。

<table>
<thead>
<tr>
<th>ボード開発セットに含まれるデバッグボードを実装して、デバッグの使用を可能にします。実装したデバッグボードを使用して動作確認をしております。</th>
</tr>
</thead>
</table>

・デバッグボードの設定

#### デバッグボードは実装されています。各ビットの意味を下表に示します。設定位置に合わせて変更します。

<table>
<thead>
<tr>
<th>番号</th>
<th>設定内容</th>
<th>利用機能</th>
<th>出荷時状態</th>
<th>設定位置</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>アプリケーションブート条件</td>
<td></td>
<td>自動</td>
<td>自動</td>
</tr>
<tr>
<td>1</td>
<td>手動</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>通信速度</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>未使用</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>クロック</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>未使用</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ロードモジュール登録初期化</td>
<td></td>
<td>未初期化</td>
<td>未初期化</td>
</tr>
<tr>
<td>8</td>
<td>初期化</td>
<td>未初期化</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>電源条件</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>自動</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>手動</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>端子</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ビッグ</td>
<td>リトル</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

サリアル通信速度はダイナミックダウンローディング機構の出力の通信速度を合せるためです。

---

本セクションはダイナミックダウンローディング機構で使用しています。電源をするとロードモジュール登録情報を初期化します。全て登録されていない状態にフォーマットします。

---

本セクションを設定すると、電源を基準に挿入したことにより電源が全体に供給されます。
れます。この設定では、電源を押下する事により電源コントローラによって電源が全体に供給されます。使用時は設定します。

デバッグボード
デバッグボードを装着することにより使用してのロード実行やフラッシュデバイスへの書き込みが可能となります。以下に使用する際のデバッグボードジャンパ設定を記述します。

<table>
<thead>
<tr>
<th>ジャンパビン</th>
<th>設定内容</th>
<th>設定位置</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ブートアドレスへの配置デバイス</td>
<td>オープン</td>
</tr>
<tr>
<td></td>
<td>オープン/フラッシュ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ショートデバッグボード</td>
<td></td>
</tr>
<tr>
<td></td>
<td>端子レベル</td>
<td>ショート</td>
</tr>
<tr>
<td></td>
<td>オープン通常モード</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ショートモード</td>
<td></td>
</tr>
</tbody>
</table>

拡張ボード（ジャンパビン）設定
拡張ボードもジャンパビンによってハードウェア設定が変更可能です。この設定は出荷時デフォルト設定のままで変更は不要です。

デフォルト設定内容
- ボード割付エリア
- オープン (選択)
- コントローラ割り込み信号
- ショートモード (選択)
ターゲットソフトウェアの構成

配布セットを展開したディレクトリ以下のディレクトリです。以下はターゲットソフトウェアの構成ディレクトリです。完全準拠フルセットカーネルの配布セットから以下のディレクトリをコピーしてマージします。

カーネル本体
システムタスク
ライブラリ
ビルドユーテリティ
システム依存部

ローダ機能のコンフィグレーション説明

ターゲットで動作するローダ機能をネットワーク環境に合わせて以下の変更をする必要があります。

ダウンロード運行モード
ファイル名が以下に示すように変更されます。

<table>
<thead>
<tr>
<th>設定項目</th>
<th>マクロ名</th>
<th>設定例</th>
</tr>
</thead>
<tbody>
<tr>
<td>運用モード設定</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ネットワークサーバデマンド</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ネットワーククライアントデマンド</td>
<td></td>
<td></td>
</tr>
<tr>
<td>オフライン運行</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ネットワークパラメータの設定
ファイル名が以下に示すように変更されます。

<table>
<thead>
<tr>
<th>設定項目</th>
<th>マクロ名</th>
<th>設定例</th>
</tr>
</thead>
<tbody>
<tr>
<td>ターゲットアドレス</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ダウンロード接続ポート</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

本設定はオフライン運用を選択した場合、無効となる情報です。
章 チュートリアル編

3.1 ダウンローディングターゲットシステムの準備

3.1.1 T-Engine CPU のハードウェアセットアップ

ネットワーク操作を行う場合、ターゲット機器とダウンローディングサーバーをネットワークケーブルで接続します。オフライン操作を行う場合、メディアを介して機器とサーバーはインタフェースをします。本節では接続方法の一例を示します。

図 3.1.1.1：T-Engine CPU を単体で使用する場合の接続方法（ネットワーク操作）

図 3.1.1.2：T-Engine CPU を単体で使用する場合の接続方法（オフライン操作）
図 1: 1を使用する場合の接続方法
ベースモジュールのコンフィグレーション及びビルド

ベースモジュールの作成は概ね以下の作業手順になります。以降サンプルベースモジュールを引用し各作業を説明します。

1. ベースモジュールの作成手順
2. システム構成の決定
3. ユーザプログラムのコーディング
4. ベースモジュールの構成編集
5. ベースモジュールのビルド
6. モジュール管理への登録
7. ターゲットへの書き込み・起動
作業1:システム構成の決定
ベースモジュールに含むサブシステムの構成を決定します。
最低限必要なものはカーネル機能、ローダ機能、ネットワークプロトコル機能です。必要に応じてサブシステムを追加して行きます。

作業2:ユーザプログラムのコーディング
必要に応じてベースモジュールに含めるユーザプログラムをコーディングします。

作業3:ベースモジュールの構成編集
ベースモジュールコンフィギュレーションファイルを編集し、ベースモジュールに必要なカーネルオブジェクトを追加していきます。また、今後動的に追加されるカーネル資源の最大数やロードモジュールの装着最大数を見積り、コンフィギュレーションファイルに設定します。サンプルでは本ファイルは`loader_loaderfile`というファイル名で格納されています。

<table>
<thead>
<tr>
<th>カーネルビープ領域サイズ</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>ベースモジュールの属性設定</td>
<td>100</td>
</tr>
<tr>
<td>資源最大化の定義</td>
<td>30</td>
</tr>
<tr>
<td>資源最大数の定義</td>
<td>30</td>
</tr>
</tbody>
</table>

カーネルサービスコールはコンフィギュレータ機能によって自動エクスポートされます。この事により
ユーザはベースモジュールに含まれるサービスコールをロードモジュールサイドから別ロード単位である事を意識することなく呼び出す事が可能です。その他ベースモジュールに含まれるシンボルをエクスポートしたい場合、COPY静的シンボルを使用して明示的に公開する必要があります。
作業：ベースモジュールのビルド
次にベースモジュールのビルドを行います。サンプルベースモジュールのフォルダ
target/modules/baseにて以下のコマンドを実行します。

```
$ make ROM=true
```

下図のように実行によりビルド処理が動作し、最終ロードモジュールモジュールが作成されます。

![ビルドフロー図](image_url)
作業 １: モジュール管理データベースへの登録
ベースモジュール書き込み動作させるターゲット機器のネットワークノード情報をモジュール管理
ノードに追加します。

<table>
<thead>
<tr>
<th>ターゲット機器 ノード登録情報</th>
</tr>
</thead>
<tbody>
<tr>
<td>ターゲット機器の接続情報を登録する。</td>
</tr>
<tr>
<td>書式:</td>
</tr>
<tr>
<td>テーターゲット名</td>
</tr>
<tr>
<td>---------------</td>
</tr>
</tbody>
</table>

作業 ２: ベースモジュールのターゲット機器への書き込み
ターゲットに電源を投入し、最終ロードモジュール ベースモジュール IDを バートリムボードのフラッシュ
モードに書き込みます。フラッシュ モードへのプログラム書き込み方法として、プログラムの利用
する方法とモードを使用する方法があります。書き込み方法の詳細は バートリムガイドのマニュ
アルをご覧下さい。
ロードモジュールの作成

1. ロードモジュールの作成手順
2. ロードモジュール構成の決定
3. ロードモジュール
ユーザプログラムのコーディング
4. ロードモジュールの構成編集
5. ベースモジュール構成へ
ロードモジュールを追加
6. ロードモジュールのビルド
7. モジュール管理DBへの登録
8. ダウンロード実行
作業1 ロードモジュール構成の決定
ロードモジュールに含むタスク構成やライブラリ構成を決定します。また、ここでロードモジュールが外部に公開するサービスも決定します。本書では  社製  サーバをロードモジュールとして動作させる方法を例として説明します。

作業2 ユーザプログラムのコーディング
作業1で決定した構成に基づき、ロードモジュールに含めるソースコードのコーディングを行います。ロードモジュール初期化処理、終了処理は必須ですので合せてコーディングを行います。ここでは 社製 サーバの下の  ベースディレクトリ のディレクトリにソースファイルを格納するものとします。

作業3 ロードモジュールの構成編集
ロードモジュールコンフィグレーションファイルを作成し、ロードモジュールに含めるカーネルオブジェクトを静的  で記述します。  には サーバが使用するカーネルリソースを設定しています。

```c
#include "websvr.h"
#include "ebs.h"

VATR_LDM( ldm_websvr, "web server", { TYPE_OPTION, 0x200} );

CRE_TSK( ID_RTIP_TSK_MAINAPP,
{ TA_HLNG|TA_ACT, 1, tsk_websvr, 12, 4096, NULL } );

/* WEBCFG_WEB_MAX_SPAWN */
CRE_TSK(ID_RTIP_TSK_WEBS_0, {TA_HLNG, 0, ks_task_entry, 10, 16384, NULL });
CRE_TSK(ID_RTIP_TSK_WEBS_1, {TA_HLNG, 0, ks_task_entry, 10, 16384, NULL });
CRE_TSK(ID_RTIP_TSK_WEBS_2, {TA_HLNG, 0, ks_task_entry, 10, 16384, NULL });
CRE_TSK(ID_RTIP_TSK_WEBS_3, {TA_HLNG, 0, ks_task_entry, 10, 16384, NULL });
CRE_TSK(ID_RTIP_TSK_WEBS_4, {TA_HLNG, 0, ks_task_entry, 10, 16384, NULL });

/* Signal Semaphore */
CRE_SEM( ID_WEB_SEM0, { TA_TFIFO, 1, 1 } );  /*  */
CRE_SEM( ID_WEB_SEM1, { TA_TFIFO, 1, 1 } );  /*  */
CRE_SEM( ID_WEB_SEM2, { TA_TFIFO, 1, 1 } );  /*  */
CRE_SEM( ID_WEB_SEM3, { TA_TFIFO, 1, 1 } );  /*  */
CRE_SEM( ID_WEB_SEM4, { TA_TFIFO, 1, 1 } );  /*  */
CRE_SEM( ID_WEB_SEM5, { TA_TFIFO, 1, 1 } );  /*  */
CRE_SEM( ID_WEB_SEM6, { TA_TFIFO, 1, 1 } );  /*  */
CRE_SEM( ID_WEB_SEM7, { TA_TFIFO, 1, 1 } );  /*  */
CRE_SEM( ID_WEB_SEM8, { TA_TFIFO, 1, 1 } );  /*  */
CRE_SEM( ID_WEB_SEM9, { TA_TFIFO, 1, 1 } );  /*  */

/* Exported Variables */
EXPORT_INI( web_init_mdl );
EXPORT_TER( web_exit_mdl );
```
作業 4 ベースモジュール構成へのロードモジュール登録
次に 静的 ロードモジュールコンフィグレーションファイルに先程作成したロードモジュールコンフィグレーション情報を登録します。

登録後、ベースモジュールディレクトリで コンフィグレータを動作させ、ベースモジュール情報にロードモジュールを追加します。（ここではベースモジュールのコンパイル・リンク動作は不要です。）この操作を行うことでモジュール が自動採番され、その 値は マクロとしてマクロとして出力されます。ここではモジュール が として採番出力されたと仮定し説明します。

作業 5 ロードモジュールのビルド
次に先程のロードモジュール作業ディレクトリにて ロードモジュールコンパイラーを実行します。この実行でロードモジュール中間ファイル とモジュールビルダへの補助情報ファイル が作成されます。

作業 6
ロードモジュールの準備として最後のステップである、モジュール管理データベースへの作成ロードモジュール登録を行います。ここではターゲット に実装できるロードモジュール として登録します。
ダウンロードの操作

ターゲットの起動

ターゲットボードの電源を入れ、フラッシュに書き込んだベースモジュールを動作させます。正常に起動するとカーネルとダイナミックダウンローディング機能のパナーが出力される事を確認することができます。

ダウンロードサーバの起動

ダウンロードサーバを起動します。システム起動時、デーモンプロセスとして起動する事も可能ですが、ここではrootユーザでフォアグランドプロセスとして起動しています。
ネットワーク サーバデマンド操作によるダウンロード実行

機器自体が設備に隠蔽され、機器ユーザとのインタフェースを持たない場合、サーバより機器に対して指示を行い、プログラムをプラグインしたり、バージョンアップする運用形態が考えられます。本節ではサーバデマンド操作でのダウンローディングの実際の操作を説明します。

まず、ターゲットサーバのシェルプロンプトにより idl_shell と入力し、ダウンロードシェルプログラムを起動します。

ダウンロードシェルから接続可能なターゲットシステム一覧を表示し、ターゲット接続を行います。以下では コマンドという名称が設定されている機器に接続を行っています。接続が正常に終了するとシェルプロンプトに変わります。プロンプトが接続状態に変わると、ターゲットへの操作（状態取得、ロードモジュールの削除、ダウンロード）が可能となります。

なお、接続するターゲットはグループ化が可能です。コマンドでグループ名を指定した場合、複数の機器に同時に接続を行い、ロードモジュールの一斉ダウンロードが行えます。いよいよダウンロードの実行ですが、以下ではダウンロード前に機器のモジュール状態を確認しています。機器の状態表示からベースモジュール以外は何もロードされていないことが確認できます。
以下は接続ターゲットに対してモジュールのロードを実行し、機器のロード状態を確認する操作したところです。新たにサーバロードモジュールが装着された事が確認できます。

<table>
<thead>
<tr>
<th>モジュール</th>
<th>モジュールID</th>
<th>バージョン</th>
<th>アトリビュート</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPA web server (OPTION)</td>
<td>1</td>
<td>2.00</td>
<td>optional</td>
</tr>
<tr>
<td>IPA web server (OPTION)</td>
<td>1</td>
<td>3.00</td>
<td>optional</td>
</tr>
</tbody>
</table>

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得

モジュールのリソースの取得
以下はダウンロードしたサーバーモジュール機能によって返却されるホームページドキュメントをブラウザより閲覧した画面です。
上記にてダウンロードしたロードモジュールはオプション属性のため、アンロードすることが可能です。以下では、アンロード実行後、再度モジュールの状態を確認しアンロードのロードモジュールが削除されたことを確認しています。

シェルコマンドを終了するには『exit』とタイプします。このコマンドでターゲット接続を終了し、ダウンロードサーバとのセッションを終了します。
ネットワーク クライアントデマンド操作によるダウンロード実行

機器自体が機器ユーザとのインタフェース機能を持つ場合、機器ユーザの都合で新たなプログラ
ムのプラグインやバージョンアップ操作を行う運用形態が考えられます。
本節ではクライアント(機器)デマンド操作でのダウンローディングの実際の操作をご説明います。

ベースモジュールをクライアントデマンドコンフィグレーションを行ってビルドした場合、以下のように
システム起動時 ダウンロードシェルが起動されます。
（機器で動作するダウンロードシェルは Ethernetポート シリアルケーブルをパソコン上でポート
Hに接続し、汎用ターミナルソフトを起動することで操作することができます。）
以下では Toppers コマンドにて実装されているコマンド一覧を表示し、タッピングコマンドで機器のアド
レス(IP アドレス)を確認しています。
機器よりダウンロード操作を行う場合は コマンドでダウンロードサーバに接続します。接続が正常終了するとプロンプトが に変わります。

サーバに接続完了後、ロード可能なモジュール一覧を取得し、モジュールロードの操作を行います。以下ではモジュール 1 = 1 バージョン番号 のロード操作を行っています。ロード実行後に コマンドでロードモジュールの装着状態を確認すると、モジュール が追加されたことが確認できます。
ダウンロードシェルのコマンドを使用して、機器自身のロードモジュールを削除することが可能です。以下は先ほどダウンロードしたロードモジュールを削除しています。

```
CON> discon

Disconnect server
```

ダウンロード操作終了時は、```
CON> unload 1
```
をコマンドにてダウンロードサーバとの切断を行います。

```
CON> unload 1
unload module
```
オフライン操作によるダウンロード実行

昨今の組み込み機器はネットワーク機能を搭載したものが増えているが、ネットワーク機能をもたない機器も依然存在しています。このような機器ではフロッピーディスクやCD-ROMのようなリムーベブルストレージ媒体を使用してソフトウェアのバージョンアップする運用が考えられます。本節ではオフライン操作でのダウンローディング操作を説明します。
一部はボードの カードスロット、サーバ側は 表の カードリードライタを使い、レキサメディア社製 写ストレージカードを使ってダウンロード実行を行っています。)

オフライン運用コンフィグレーションした機器は電源を投入すると以下のように機器で動作するダウンローディングシェルが起動され、コマンドブロントが表示されます。

<table>
<thead>
<tr>
<th>コマンド名</th>
<th>効果</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>ここに使用されるヘルプ情報が表示される</td>
</tr>
<tr>
<td>status</td>
<td>[ID]</td>
</tr>
<tr>
<td>unload</td>
<td>[ID]</td>
</tr>
<tr>
<td>build</td>
<td>[ID]</td>
</tr>
<tr>
<td>load</td>
<td></td>
</tr>
<tr>
<td>bdump</td>
<td>[モジュール名] [サイズ]</td>
</tr>
<tr>
<td>hdump</td>
<td>[モジュール名] [サイズ]</td>
</tr>
<tr>
<td>wdump</td>
<td>[モジュール名] [サイズ]</td>
</tr>
</tbody>
</table>

パーソナルボード カードスロットに カードを挿入し、コマンドをタイプすることでロードモジュールビルド指示データをファイル作成します。

<table>
<thead>
<tr>
<th>コマンド</th>
<th>効果</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy</td>
<td>ファイル作成</td>
</tr>
</tbody>
</table>
次に機器より USB カードを抜き、カードリーダに接続されている USB ドライバに先程書き込んだ USB カードを挿入します。
サーバで動作するダウンロードシェルを起動し、build コマンドでロードモジュールの作成を完了させます。

再度、サーバ USB カードリーダから USB カードを抜き、機器の USB カードスロットに挿入し、ダウンロード操作を完了させます。
以下では idl コマンドでビルドしたモジュール のバージョン番号 のロードモジュールを機器にダウンロードし、ロードモジュールの装着状態を確認しています。
第章 補足資料編

第3章 他のシステムへの移植ガイド

第4章 フル機能・カーネルサービス

ダイナミックダウンロード機能はフルセットカーネルの機能に依存しています。現状フルセットカーネル対応のプロセッサはのみですが、プロジェクトでは対応を増やしていく予定です。またユーザ自身でもフルセットカーネルのシステム依存部の構築ガイドを参照することにより対応することが可能です。詳細はフルセットカーネルの移植ガイドを参照してください。

第4.1 ネットワークサービス

ダイナミックローディングをネットワーク上で動作させるためには、ネットワークへのアクセス機能が必要となります。ダイナミックローディングは現状以下のネットワークプロトコルを用いて実装しておりますが、ソケット利用可能な他社のネットワークプロトコルを実装することで容易にダイナミックローディング機能を移植する事が可能です。以下に移植の際見直す項目を示します。

ファイル名：
ネップウォーキャン機能の初期化
ネップウォーキャン機能の終了
トランスポートオープン
トランスポート解放
ダウンロードセッションのオープン
送信処理

int init_comm ( IDL_SAP *server_sap, IDL_SAP *target_sap )
int terminate_comm ( VP param )
ER_ID comm_open_init ( void )
ER_ID comm_open_term ( ID so_id )
ER_ID comm_open_req ( ID conn_so_id )
int comm_send_req ( ID so_id, UB *data, UH len )
受信処理

セッションクローズ

ファイル転送要求

ファイルデータの読み出し

ファイルデータの確認応答

ファイル転送の中止

ファイル転送の終了

受信データ発生確認
フラッシュメモリドライバ

配布ソースは ターゲットボード に搭載されたフラッシュメモリチップ に対する、メモリドライバを実装しています。他のターゲットシステムに移植する際、搭載されているメモリチップに適合したメモリドライバを移植する必要があります。補足説明として他のメモリチップに移植する際、変更が必要なファイル、関数を以下に示します。

修正ファイル名：

修正対象関数名：
・フラッシュ消去機能
・フラッシュ書き込み機能
・フラッシュバッファ書き込み機能
・フラッシュ保護機能
・フラッシュ保護解除機能
・フラッシュ初期化関数
4.2 ICE を使用したダウンロードモジュールのデバッグ方法の紹介

ダウンロードサーバーをオプション付で起動した場合、ナイナミックローディングで作成した最終リンクファイルを削除しません。これ最終リンクファイルに存在するデバッグ情報のみをダウンロード後デバッグに読み込みます。デバッグ情報のみをロードモジュールの言語ソースデバッグが可能となります。

最終ロードモジュールが残っているディレクトリ:
ダウンローディングサーバ設定ファイルで設定した以下のディレクトリ
...ディレクトリ/.../ダウンローディングセッション番号

なお、ダウンローディングセッション番号はロード実行完了メッセージで確認することが可能です。

最終リンクアウトプットファイル名:
ダウンローディングサーバ設定ファイルで設定したファイルが設定ファイル名として設定したファイルデフォルトファイル名は...

補足:
ツールチェーンをサポートするデバッグシステムは、シームレスにオブジェクトファイルを読み込む事が可能なものとデバッグが認識できるファイルフォーマットのコンバータを提供しているものが存在します。
今回使用した社製デバッグシステムは後者の方でツールチェーンというオブジェクトで実行可能なオブジェクトコンバータが付属します。今回の開発ではこのを使用しシンボルデバッグを行いました。