

文書番号: ESC-APN-026-02

adviceLUNA Linux デバッグ手順 (MIPS コア編)

はじめに

「adviceLUNA Linux デバッグ手順」(以下、本書)は、adviceLUNA にて下記 Linux 環境をデバッグする手順を説明した文書です。

- Application
- Shared Library
- Loadable Kernel Module

本書の対象となるユーザー

- adviceLUNA と microVIEW-PLUS との組み合わせで MIPS コアにて Linux デバッグを行うすべてのユーザー。

本書の対象となる環境

- HLB600 (MIPS 対応 adviceLUNA/microVIEW-PLUS デバッガソフトウェア)を使用する環境
- HLB600 (MIPS 対応 adviceLUNA/microVIEW-PLUS デバッガソフトウェア) Rev.3.11 以降を使用し、以下のソフトウェアライセンスを使用する環境
 - DLA336 (MIPS Linux 対応ソフトウェアライセンス)

注意事項

- 本書で使用している画面は、Windows 7、Internet Explorer 11 の環境で作成しています。ほかの環境をお使いの場合、表示や操作手順が異なることがあります。
- 本書の操作手順は、adviceLUNA（SLB600）および Linux サポートライブラリ（DLA336）を使用して解説しています。お使いの機種によっては、操作手順内での名称や参照しているマニュアル名が異なる場合があります。
- advice シリーズの製品を安全にお使いいただくために重要な情報は、adviceLUNA ユーザーズマニュアル（固有編）』（HLB600_podm_jpn.pdf）に記載されています。
- 本ガイドに記載されている会社名・製品名は、各社の登録商標または商標です。

アイコンについて

本ガイドで使用しているアイコンには、以下の意味があります。



特に重要な情報を記載しています。操作する際は十分に注意してください。



操作を進める上で役に立つ情報やアドバイスなどの補足事項を記載しています。



本ガイドのほかのページやほかのマニュアルなどの参照情報を記載しています。

目次

はじめに	1
本書の対象となるユーザー	1
注意事項	2
アイコンについて	2
1. Linux 環境デバッグ設定	4
1.1. プロジェクトの新規作成	4
1.2. デバッグ設定手順	5
1.3. パス変換設定	6
2. カーネルのデバッグ方法	8
2.1. カーネルのデバッグ情報ファイルのシンボルエントリ	8
3. ローダブルカーネルモジュールのデバッグ方法	9
3.1. デバッグ手順	9
3.2. 制限事項	12
3.2.1. _init 属性	12
3.2.2. アプリケーションの起動	12
4. アプリケーションのデバッグ方法	13
4.1. デバッグ手順	13
4.2. PID 選択機能	16
4.3. 仮想モードの使用方法	18
4.4. 制限事項	20
4.4.1. アプリケーションの main 関数からデバッグ	20
5. 共有ライブラリのデバッグ方法	21
5.1. デバッグ手順	21
5.2. 特定プロセスからコールされた時のみブレークさせる	24
5.3. 制限事項	24
5.3.1. 関数の先頭からのデバッグ	24
6. 参考資料	25
改訂履歴	26

1. Linux 環境デバッグ設定

1.1. プロジェクトの新規作成

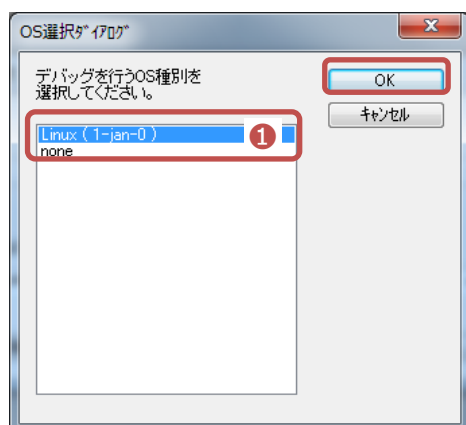
本作業は新規プロジェクト作成時に行います。

Linux デバッグ設定済みの既存プロジェクトファイルを利用する場合は不要です。

Linux デバッグを行う為には、新規プロジェクトの作成時に OS の選択が必要です。

DLA336 ライセンスを登録の場合、ユーザーシステム接続後に「OS 設定ダイアログ」が表示されます。

- ① 「OS 設定ダイアログ」画面では、OS の選択を行います。

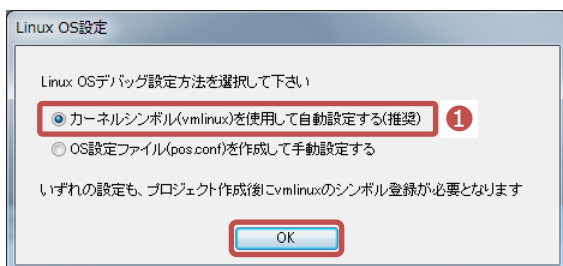


- ① OS 種別を選択します。

[Linux] を選択して [OK] をクリックしてください。

▶ 「Linux OS 設定」ダイアログが表示されます。

- ② 「Linux OS 設定」画面では、Linux OS のデバッグ設定方法を選択します。



- ① Linux OS デバッグ設定方法を選択します。

[カーネルシンボル(vmlinux)を使用して自動設定する(推奨)] を選択して、[OK] をクリックしてください。

Linux 環境のデバッグには、カーネルが起動した状態で Attach をおこない、カーネルシンボルファイル vmlinux のシンボルエントリが必要となります。

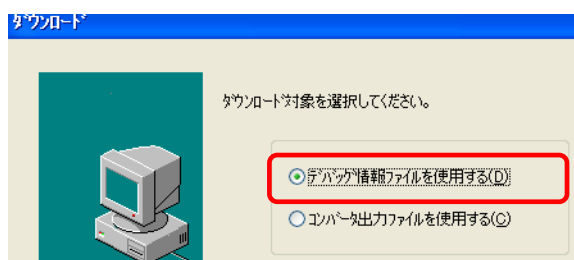
以下の手順にて Linux デバッグの設定を行ってください。

1.2. デバッグ設定手順

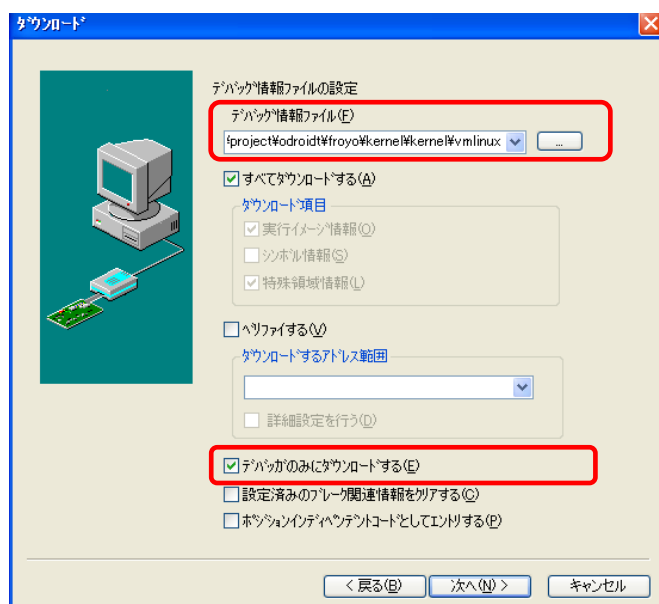
- ① プロジェクト起動後にカーネルシンボルファイル vmlinux をシンボルエントリします。
microVIEW-PLUS メニュー[環境]-[ダウンロード]を選択し、ダウンロードウィザードを開いてください。
ダウンロードアイコンをクリックしてもウィザードを開くことができます。



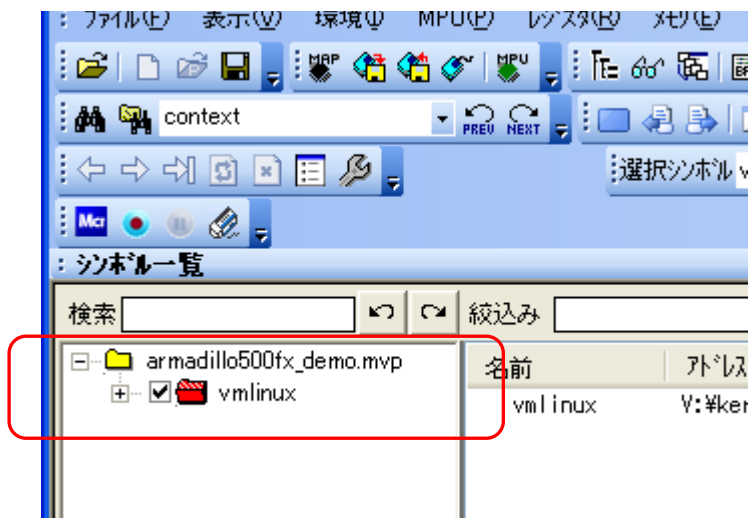
- ② 「デバッグ情報ファイルを使用する」を選択して「次へ」ボタンを押してください。



- ③ カーネルシンボルファイル vmlinux を選択、デバッガのみにダウンロードするにチェックをおこない「次へ」ボタンを押してください。



- ④ エントリ実行後に以下の vmlinux のシンボル情報が登録されれば正常に設定完了です。



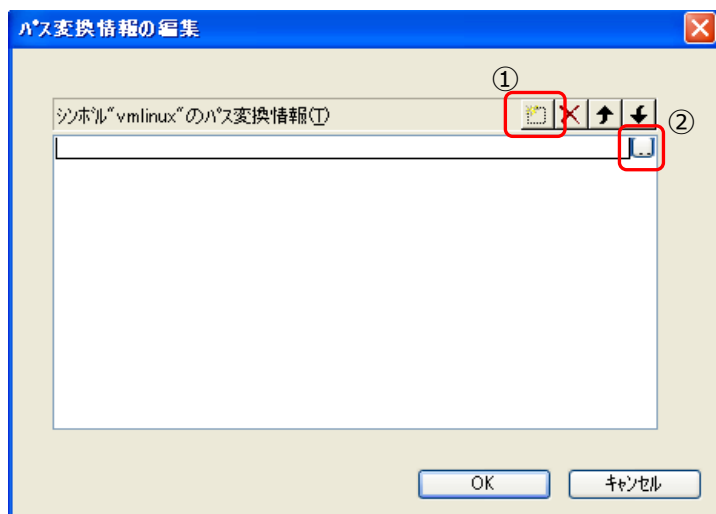
1.3. パス変換設定

Linux 環境の開発環境は Linux にてビルドされております。Windows にてパス情報を参照する際には、パス整合の設定が必要となります。

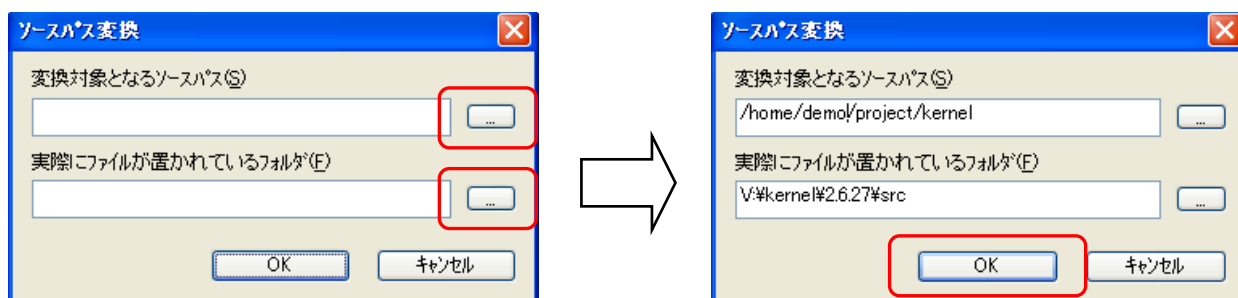
microVIEW-PLUS メニュー[表示]-[シンボラー一覧]よりシンボラー一覧を開いてください。

シンボラー一覧より、デバッグ対象の vmlinux シンボルを右クリックして「パス変換情報の編集(T)」を選択し、設定ウィンドウを開きます。



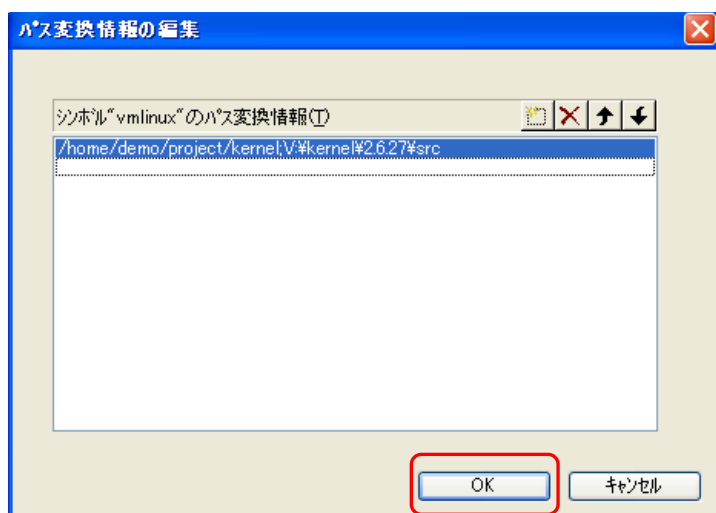


Linux 環境でのパスと Windows 環境からのパスを設定します。



変換設定が追加されます。

「OK を選択してください。」



以上で設定は完了です。

2. カーネルのデバッグ方法

Linux のデバッグにはカーネルのシンボルファイルが必要となります。カーネルシンボルファイルの情報をもち、Linux デバッグ機能が動作します。

2.1. カーネルのデバッグ情報ファイルのシンボルエントリ

1.1 章を参照し、カーネルデバッグ情報ファイル (vmlinux) をシンボルエントリしてください。エントリ後カーネルのシンボリックデバッグが可能となります。

3. ロードブルカーネルモジュールのデバッグ方法

ロードブルカーネルモジュール（Loadable Kernel Module(以下、LKM)）のデバッグ手順、制限事項を説明します。

3.1. デバッグ手順

デバッグ対象の LKM が既にロードされている場合、手順②～⑤はスキップしてください。

LKM がロードされているかの確認は、以下のいずれかでできます。

- ・ linux コンソールより「lsmod」コマンドを実行する
- ・ LinuxResourceViewer LKM タブの「Reload」ボタンを押す

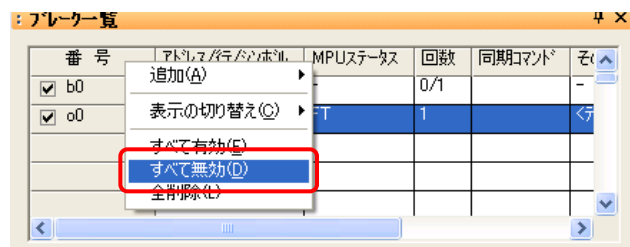
① LinuxResourceViewer の起動

microVIEW-PLUS メニュー[Linux]-[LinuxResourceViewer]を選択し、LinuxResourceViewer を起動してください。

② ブレーク設定の無効化

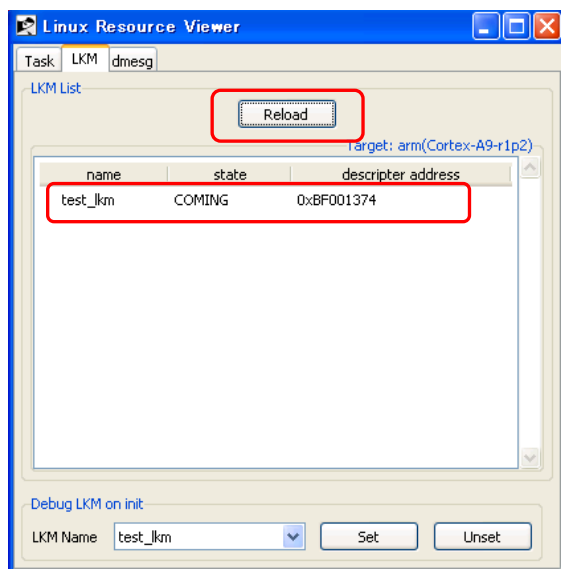
ブレーク設定をしている場合、一度全て無効化してください。

microVIEW-PLUS メニュー[表示]-[ブレーク一覧]よりブレーク一覧画面を開き、右クリックして「すべて無効」を選択してください。



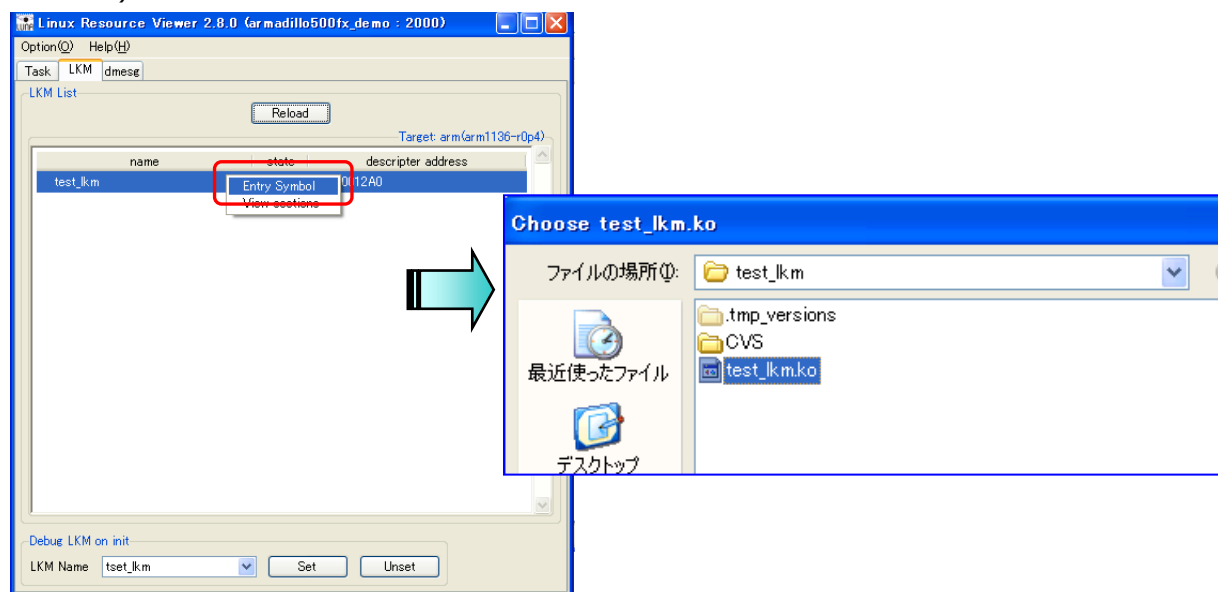
③ LKM List の更新

「Reload」ボタンを押すと、現在 Linux kernel でロードされている LKM 一覧が更新されます。



④ シンボルエントリ

デバッグ対象の LKM を右クリックし、「Entry symbol」ボタンを押し、LKM のシンボルファイル(*.ko ファイル)を選択してください。



⑤ パス変換設定

ビルドしたフォルダと、実際にソースファイルがあるフォルダが異なる場合、パスの変換設定を行ってください。

⑥ ソースファイル表示

microVIEW-PLUS で「home」コマンドを実行すると、ソースファイルが表示されます。

3.2. 制限事項

3.2.1. __init 属性

__init 属性が指定された関数のソースデバッグには対応していません。

__init 属性を削除し、リビルドしてください。

例)

```
static int __init my_init_module(void)
{
    :
}
```

3.2.2. アプリケーションの起動

init 関数からデバッグを開始する「Debug LKM on init」機能は、対応していません。

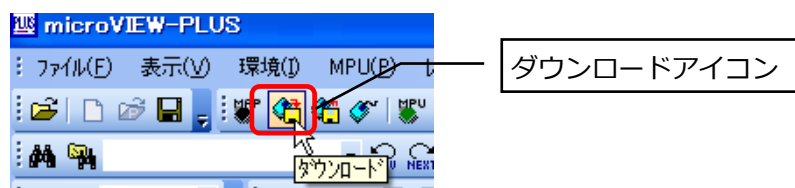
4. アプリケーションのデバッグ方法

アプリケーションのデバッグ手順を説明します。

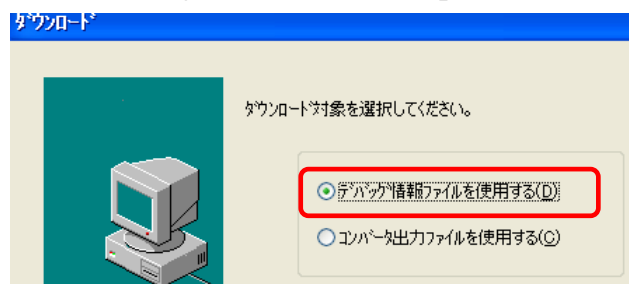
4.1. デバッグ手順

① シンボル登録

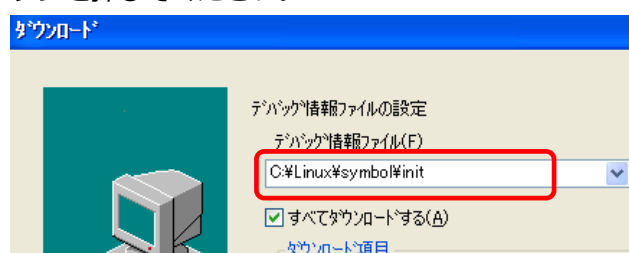
microVIEW-PLUS メニュー[環境]-[ダウンロード]を選択し、ダウンロードウィザードを開いてください。
ダウンロードアイコンをクリックしてもウィザードを開くことができます。



「デバッグ情報ファイルを使用する」を選択して「次へ」ボタンを押してください。



デバッグ対象とするアプリケーションのシンボルファイル(デバッグ情報ファイル)を選択して「次へ」ボタンを押してください。



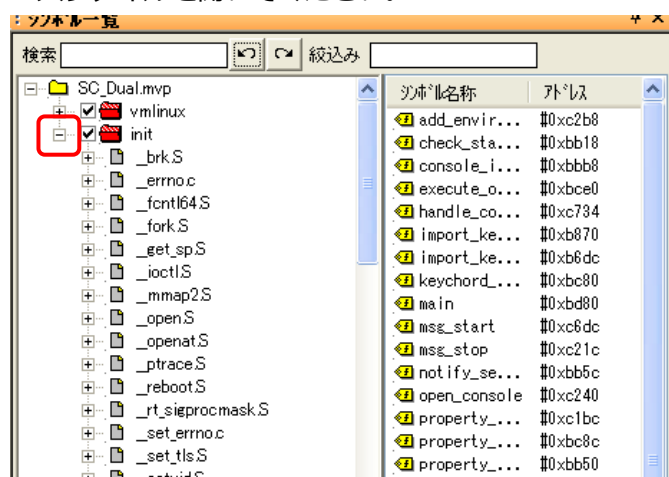
確認画面が表示されますので、内容を確認して「ダウンロード開始」ボタンを押してください。
シンボル登録が完了します。

② パス変換設定

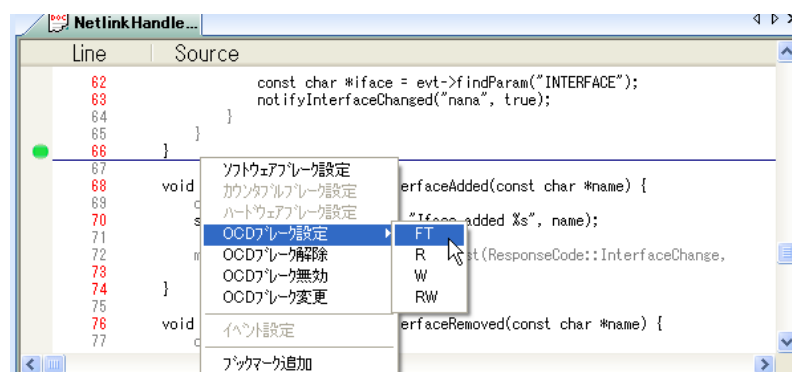
ビルドしたフォルダと、実際にソースファイルがあるフォルダが異なる場合、パスの変換設定を行ってください。

③ ブレーク設定

シンボルー覧よりシンボルをツリー展開し、ソースファイル名、もしくは関数名をダブルクリックしてソースファイルを開いてください。

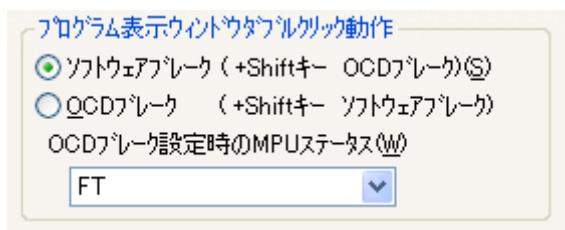


実行行にカーソルをあわせ、[右クリック]-[OCD ブレーク設定]-[FT]を選択して OCD ブレークを設定してください。



※ デフォルト設定の場合、OCD ブレークはソース上で「Shift キー」を押しながらダブルクリックすることでも設定できます。

設定は、microVIEW-PLUS メニュー[ツール]-[オプション]の「プログラム表示ウィンドウダブルクリック動作」設定を確認してください。



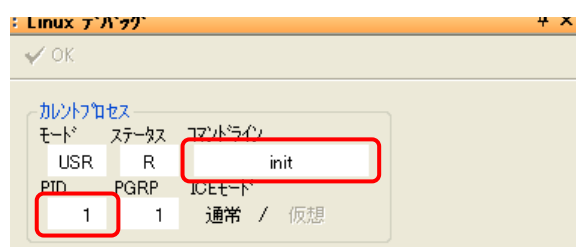
④ ユーザーシステムの実行

ユーザーシステムがブレーク中の場合、go コマンドで実行してください。

⑤ デバッグ対象プロセスでのブレーク

手順③で設定した OCD ブレーク行が実行されると、ブレークが発生します。

OCD ブレークは、同一の論理アドレスが実行された場合、デバッグ対象以外のプロセスでもブレークが発生します。microVIEW-PLUS メニュー[Linux]-[デバッグ]より Linux デバッグ画面を開き、カレントプロセス欄がデバッグ対象のプロセスであるかを確認してください。



デバッグ対象プロセスではない場合、該当のプロセスでブレークするまで再実行してください。

また、複数回再実行してもデバッグ対象プロセスでブレークしない場合、以下のいずれかの方法をご使用ください。

- ・ PID 選択機能によって、デバッグ対象プロセスを指定することが可能です。
使用方法については、「PID 選択機能」を参照してください。
- ・ デバッグ対象のプロセスが実行されていない場合、PID 選択機能はご使用になれません。

⑥ デバッグ開始

デバッグ対象プロセスのデバッグを開始してください。

カレントコンテキストに対しては、ソフトウェアブレークの設定が可能です。ソフトウェアブレークは、ブレーク設定したプロセスのみがブレーク対象となります。

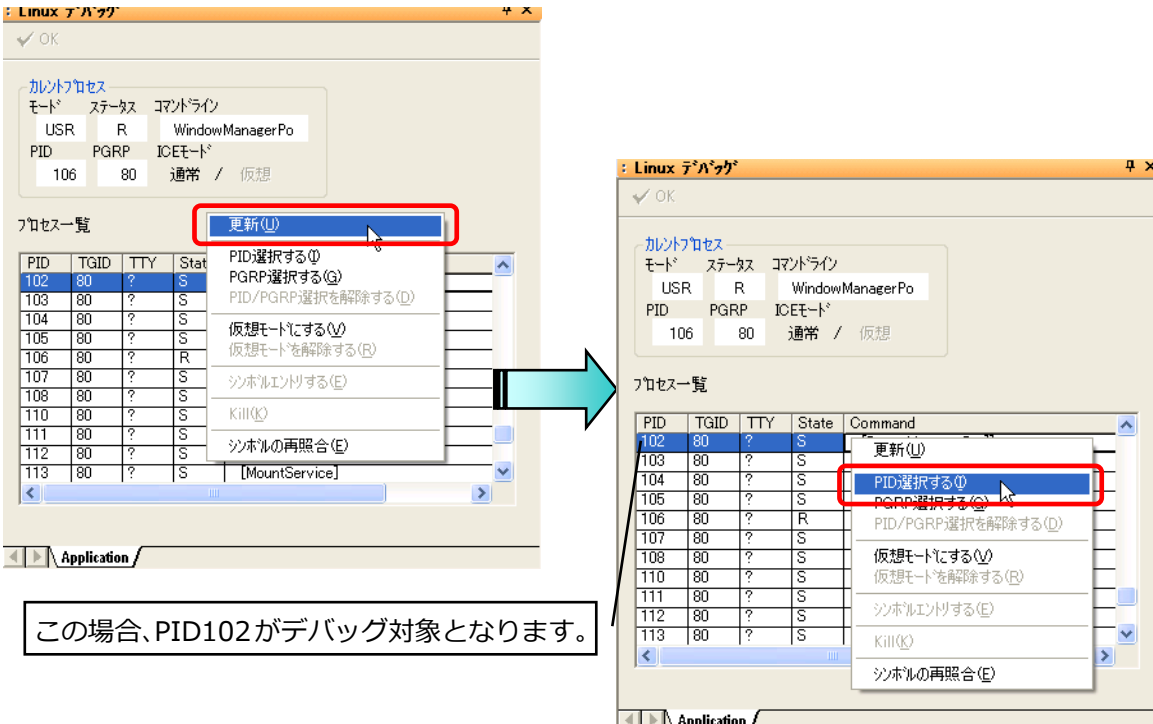
4.2. PID 選択機能

OCD ブレークは、同一の論理アドレスが実行された場合、デバッグ対象以外のプロセスでもブレークが発生しますが、PID 選択機能を使用することで、指定したプロセスのみでブレークを発生させることが可能です。

microVIEW-PLUS メニュー[Linux]-[デバッグ]より Linux デバッグ画面を開き、画面上で右クリックして「更新」を選択してください。

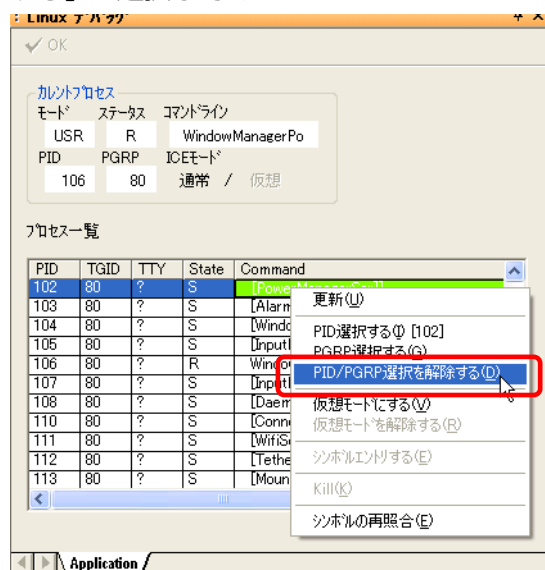
プロセス一覧よりデバッグ対象のプロセス(PID)を選択し、右クリックして「PID 選択する」を選択してください。

選択したプロセスがデバッグ対象となります。



この場合、PID102がデバッグ対象となります。

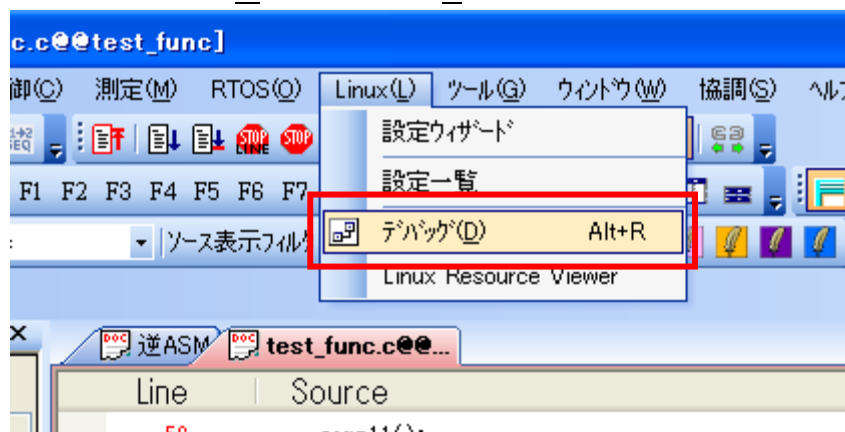
PID 選択を解除する場合は、PID 選択されているプロセスを選択し、右クリックして「PID/PGRP を解除する」を選択してください。



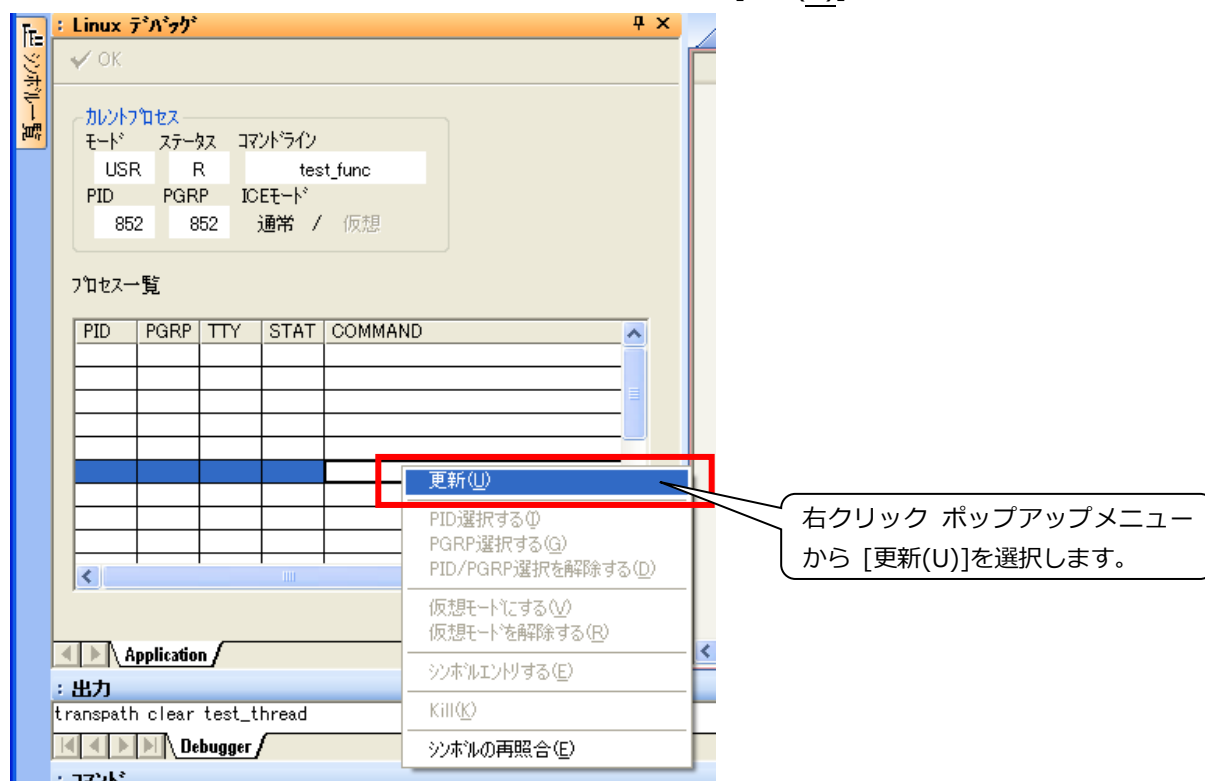
4.3. 仮想モードの使用法

仮想モードを使用することにより、カレントコンテキスト以外の任意のプロセス/スレッドにデバガを切り替えてデバッグすることができます。

- ① ツールバー [Linux(L)] → [デバッグ(D)] を選択します。

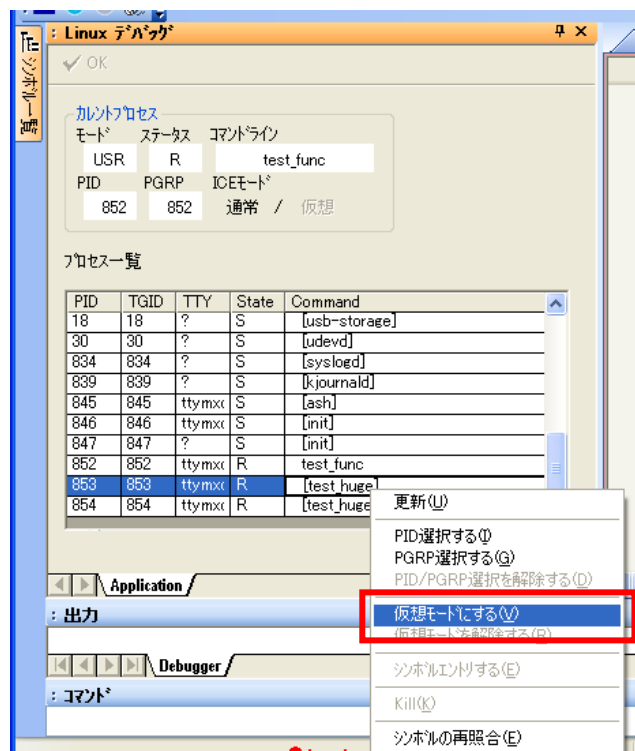


- ② 「Linux デバッグ」画面表示後、プロセス一覧の右クリック → [更新(U)]を選択します。

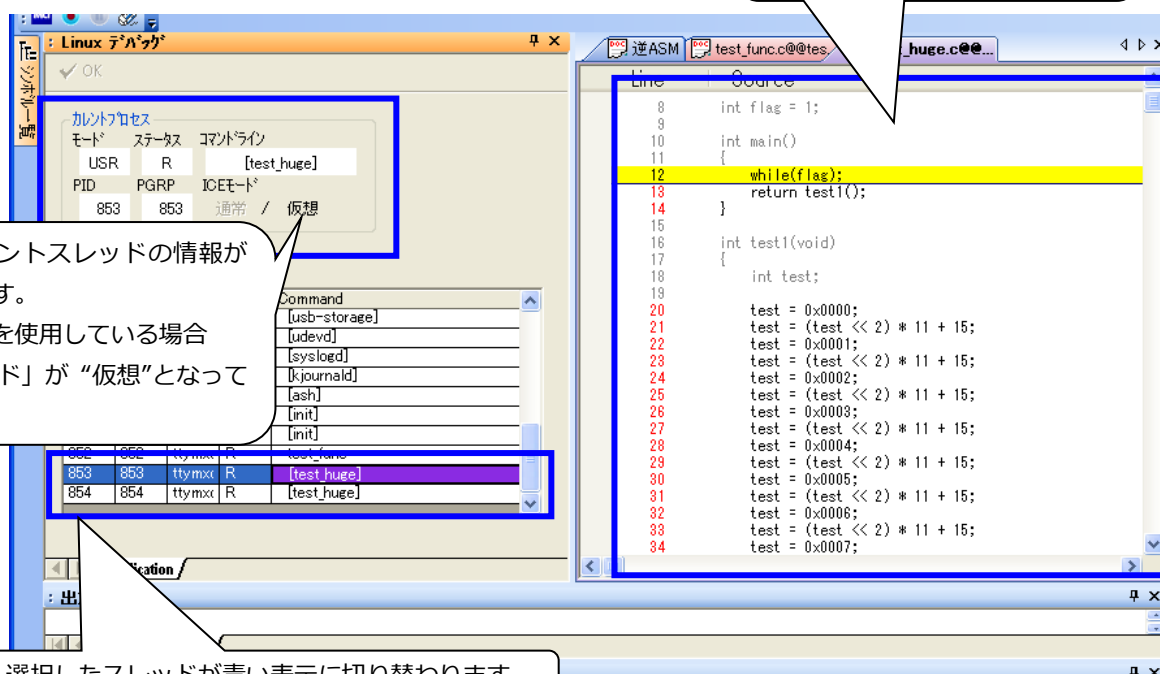


スレッド一覧表示が表示されます。

- ③ デバッグしたいスレッドを選択して右クリック ポップアップメニュー → [仮想モードにする(V)]を選択、またはデバッグしたいスレッドをダブルクリックします。



切り替えたアプリケーションのシンボルを登録している場合、ソースが表示されます。



4.4. 制限事項

4.4.1. アプリケーションの main 関数からデバッグ

「アプリケーションの main 関数からデバッグ」機能は、対応しておりません。

5. 共有ライブラリのデバッグ方法

共有ライブラリ(Shared Library)のデバッグ手順を説明します。

5.1. デバッグ手順

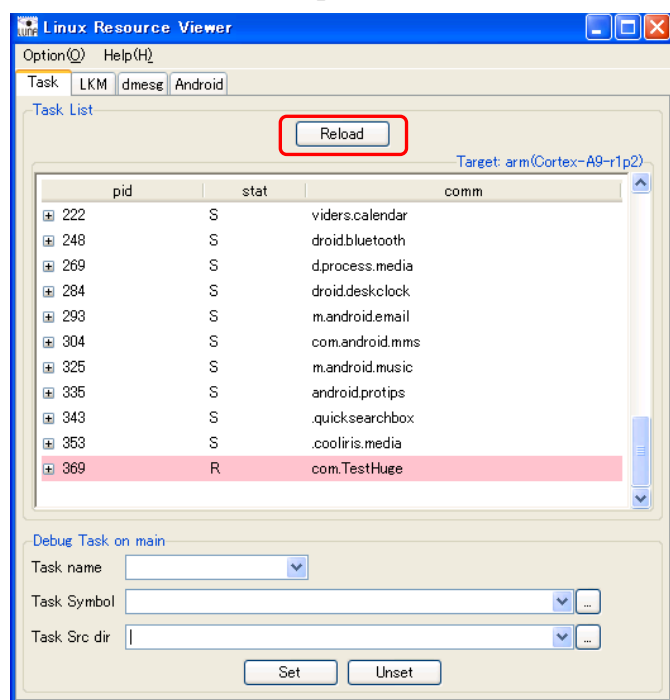
ロード済みの共有ライブラリのデバッグ手順です。

① LinuxResourceViewer の起動

microVIEW-PLUS メニュー[Linux]-[LinuxResourceViewer]を選択し、LinuxResourceViewer を起動してください。

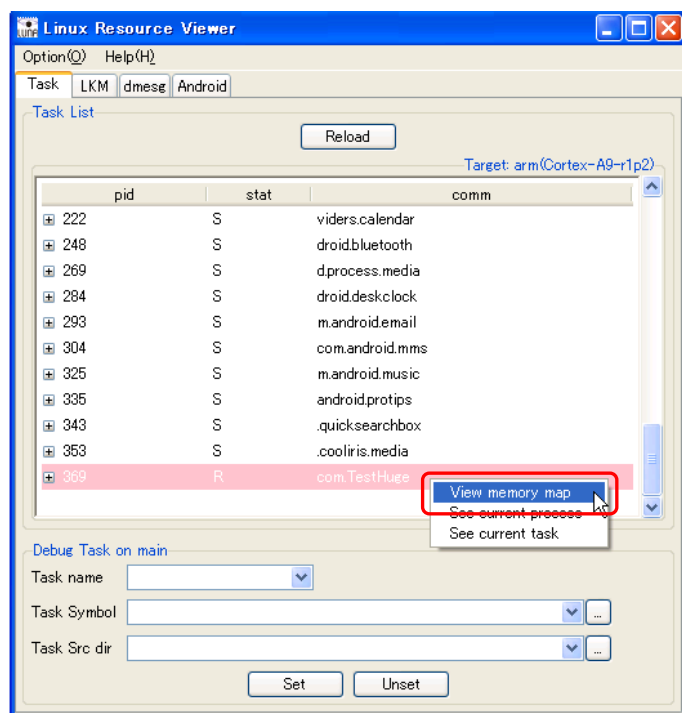
② Task list 更新

Task list タブの「Reload」ボタンを押してタスク一覧表示を更新してください。



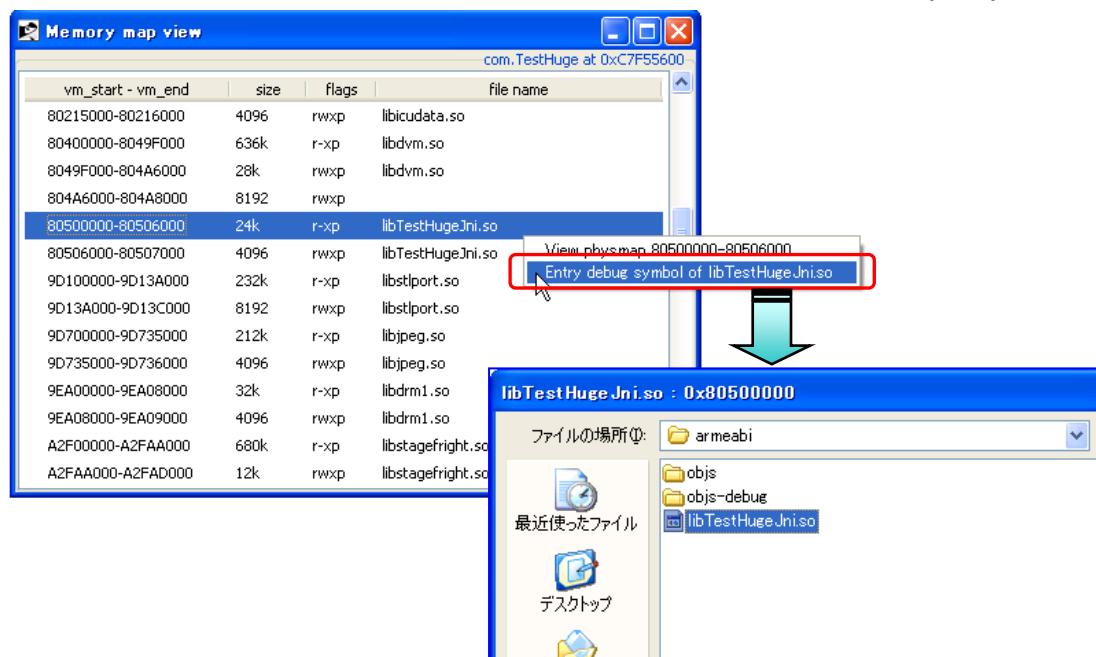
③ メモリマップ表示

デバッグ対象の SharedLibrary をロードしているタスクを右クリックし、「View memory map」を選択してください。



④ シンボルエントリ

Memory Map View にてデバッグ対象の SharedLibrary を右クリックして「Entry debug symbol of libxxx.so」を選択し、デバッグ対象とする共有ライブラリのシンボルファイル(*.so)を指定してください。



⑤ パス変換設定

ビルドしたフォルダと、実際にソースファイルがあるフォルダが異なる場合、パスの変換設定を行ってください。

5.2. 特定プロセスからコールされた時のみブレークさせる

ブレーク対象プロセスを指定する場合は、「4.2 PID 選択機能」をご使用ください。

5.3. 制限事項

5.3.1. 関数の先頭からのデバッグ

「関数の先頭からのデバッグ」機能は、対応していません。

6. 参考資料

- 「microVIEW-PLUS インストールガイド」 (mvp_installguide.pdf)
- 「microVIEW-PLUS ユーザーズマニュアル(共通編)」 (mvwPLUSj.pdf)
- 「microVIEW-PLUS Linux サポートライブラリユーザーシステムセットアップマニュアル TX49, MIPS32[DA336/DLA336]」 (DA336_DLA336_jpn.pdf)
- 「microVIEW-PLUS Linux サポートライブラリユーザーズマニュアル」 (mvwPLUS_Linuxj.pdf)
- LinuxResourceViewer ヘルプファイル (LinuxResourceViewerHelp.chm)
- 「adviceLUNA ユーザーズマニュアル (固有編)」 (HLB600_podm_jpn.pdf)

改訂履歴

版	発行日付	変更内容
第 1 版	2015.02.10	新規発行
第 2 版	2017.04.01	新商号対応