インテル® OpenMP* 互換ライブラリー

利用ガイド

http://www.intel.com/jp/software/products/

著作権と商標について

本資料に掲載されている情報は、インテル製品の概要説明を目的としたものです。本資料は、明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず、いかなる知的財産権のライセンスを許諾するためのものではありません。製品に付属の売買契約書『Intel's Terms and Conditions of Sales』に規定されている場合を除き、インテルはいかなる責を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商品性に関する保証、第三者の特許権、著作権、その他、知的所有権を侵害していないことへの保証を含む) に関しても一切責任を負わないものとします。

インテルによる書面での同意がない限り、インテル製品は、インテル製品の停止を起因とする人身傷害または死亡を想定して設計されていません。

インテル製品は、予告なく仕様や説明が変更される場合があります。機能または命令の一覧で「留保」または「未定義」と記されているものがありますが、その「機能が存在しない」あるいは「性質が留保付である」という状態を設計の前提にしないでください。これらの項目は、インテルが将来のために留保しているものです。インテルが将来これらの項目を定義したことにより、衝突が生じたり互換性が失われたりしても、インテルは一切責任を負いません。この情報は予告なく変更されることがあります。この情報だけに基づいて設計を最終的なものとしないでください。

本資料で説明されている製品には、エラッタと呼ばれる設計上の不具合が含まれている可能性があり、公表されている仕様とは異なる動作をする場合があります。現在確認済みのエラッタについては、インテルまでお問い合わせください。

最新の仕様をご希望の場合や製品をご注文の場合は、お近くのインテルの営業所または販売代理店にお問い合わせください。

本書で紹介されている注文番号付きのドキュメントや、インテルのその他の資料を入手するには、1-800-548-4725 (アメリカ合衆国) までご連絡いただくか、インテルの Web サイト (http://www.intel.com) を参照してください。

インテル・プロセッサー・ナンバーはパフォーマンスの指標ではありません。プロセッサー・ナンバーは同一プロセッサー・ファミリー内の製品の機能を区別します。異なるプロセッサー・ファミリー間の機能の区別には用いません。詳細については、http://www.intel.co.jp/jp/products/processor_number/ を参照してください。

Intel、インテル、Intel ロゴは、アメリカ合衆国およびその他の国における Intel Corporation の商標です。

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

目次

1 Windows* 互換ライブラリー

Windows 用のインテル® OpenMP* 互換ライブラリーより、Microsoft* C/C++ コンパイラー (cl) でコンパイルされた OpenMP オブジェクト・ファイルと、インテル® C++ コンパイラー (icl) またはインテル® Fortran コンパイラー(ifort) でコンパイルされた OpenMP オブジェクト・ファイルを組み合わせて使用できます。リンクフェーズで、ランタイム・ライブラリーの単一でコヒーレントなコピーが生成されます。

インテル® OpenMP ライブラリーのレガシーバージョンは、インテル以外のコンパイラーとの互換性をサポートしていませんが、インテル® コンパイラーの以前のバージョンとの互換性のために含まれています。

1.1 システム要件

インテル® OpenMP 互換ライブラリーは、インテル® C++ コンパイラー 10.1 およびインテル® Visual Fortran コンパイラー 10.1 に含まれています。使用する Microsoft コンパイラーは OpenMP をサポートしている必要があります。インテル® OpenMP 互換ライブラリーは、Microsoft* Visual Studio* 2005 の C/C++ と互換性があります。Microsoft Visual Studio の新しいバージョンとの互換性については、コンパイラーの新バージョンのリリース後にお知らせします。

1.2 使用モデル

インテル® コンパイラー 10.1 では、新しいオプション `/Qopenmp-lib` をサポートしています。リンクフェーズで `/Qopenmp-lib:compat` オプションを指定すると、インテル® コンパイラーは、Microsoft C/C++ コンパイラーで作成されたオブジェクト・ファイルと互換性のある OpenMP 互換ランタイム・ライブラリーのいずれかを使用します。デフォルト設定の `/Qopenmp-lib:legacy` オプションを指定した場合、インテル® コンパイラー は、Microsoft C/C++ コンパイラーで作成されたオブジェクト・ファイルとは互換性のない OpenMP レガシー・ランタイム・ライブラリーのいずれかを使用します。

このセクションに含まれるほとんどの例およびサンプルコマンドは、インテル® C++ コンパイラーの使用方法を示しています。多くの場合、インテル® Fortran コンパイラーの使用方法も同様です。

1.2.1 アクセス

`libiomp5md.lib` (または `libiomp5mt.lib`) ライブラリーは、リンクフェーズ中にコンパイラーまたはリンカーにアクセスできなければなりません。また、`libiomp5md.dll` ライブラリーは、実行中にアプリケーションにアクセスできなければなりません。アクセスを可能にする最も簡単な方法は、インテル® コンパイラーで提供されるバッチファイルを実行することです。以下のバッチファイルを実行すると、適切なフェーズ中にライブラリーにアクセスできるように環境が設定されます。

- インテル® C++ コンパイラー: `iclvars.bat`
- インテル® Fortran コンパイラー: `ifortvars.bat`

Microsoft Visual C++* コンパイラーのみを使用している場合、ライブラリーへのアクセスを可能にするには、次のように適切な環境変数を手動で設定します。

```
C:¥> set LIB=%LIB%;<インテル・コンパイラーのインストール・フォルダ>¥IA32¥LIB
```

```
C:¥> set PATH=%PATH%;<インテル・コンパイラーのインストール・フォルダ>¥IA32¥BIN
```

Microsoft C/C++ コンパイラーで OpenMP コードを有効にするには、/openmp オプションを使用します。インテル® コンパイラーで OpenMP コードを有効にするには、/Qopenmp オプションを使用します。C/C++ のコンパイルに、Microsoft C/C++ コンパイラーを使用する場合は Microsoft バージョンの omp.h を、インテル® C/C++ コンパイラーを使用する場合はインテル・バージョンの omp.h をそれぞれ使用してください。

1.2.2 Microsoft Visual Studio の使用

Visual C++ 2005 開発環境を使用して必要な設定を行うこともできます。例えば、プロジェクトのプロパティー・ページでインテル® OpenMP ランタイム・ライブラリー (RTL) の場所を設定できます。メインメニューから **[プロジェクト] - [プロパティ]** を選択してプロジェクトのプロパティー・ページを開きます。次に、**[構成プロパティ] - [リンカー] - [全般] - [追加のライブラリー・ディレクトリー]** を選択して、次のように入力します。

```
<インテル・コンパイラーのインストール・フォルダ>¥IA32¥LIB
```

さらに、インテル® OpenMP ダイナミック RTL をランタイム時にアクセス可能にするには、対応するパスを指定しなければなりません。例えば、**[構成プロパティ] - [デバッグ] - [環境]** を選択して、次のように入力します。

```
PATH=%PATH%;<インテル・コンパイラーのインストール・フォルダ>¥IA32¥Bin
```

最後に、インテル® OpenMP RTL の名前をリンカーオプションに追加して、デフォルトの Microsoft OpenMP RTL を除外します。**[構成プロパティ] - [リンカー] - [コマンドライン] - [追加のオプション]** を選択して、次のように入力してください。

```
libiomp5md.lib /nodefaultlib:vcomp
```

1.2.3 簡単な例

すべての cl コマンドの例は、Microsoft コンパイラー用に設定されたほかの環境で実行することができます (必要な場合は「アクセス」の説明に従って設定を行ってください)。icl コマンドは、インテル® コンパイラー用の設定環境で使用してください。
環境の設定に関する詳細は、インテル® C++ コンパイラーのドキュメントまたはインテル® Fortran コンパイラーのドキュメントを参照してください。
http://www.intel.com/jp/software/products/

インテル® コンパイラー用の環境設定で Microsoft コンパイラーを使用してインテル® OpenMP 互換ライブラリーとリンクすることもできますが、名前の重複やほかの潜在的な問題を回避するため、リンクは行わないようにしてください。例えば、両方のコンパイラーに omp.h ヘッダーファイルがある場合、デフォルトではインテル® コンパイラーは自身のヘッダーファイルを使用します。

インテル® コンパイラー用の環境設定を使用している場合、Microsoft コンパイラーからネイティブの Microsoft ヘッダーファイルを使用するには、/I コンパイラー・オプションを追加します。

```
C:¥> cl /MD /Qopenmp /I <VSINSTALLDIR>¥VC¥Include hello.c /link
libiomp5md.lib /nodefaultlib:vcomp
```

インテル® C++ コンパイラーを使用してビルドする

この例は、インテル® OpenMP 互換ライブラリーをインテル® C++ コンパイラーで使用する方法を示しています。

```
C:¥> icl /MD /Qopenmp /Qopenmp-lib:compat hello.c
```

Microsoft コンパイラーを使用してビルドする

Microsoft コンパイラーを使用する場合、インテル® OpenMP 互換ライブラリーをリンクして、Microsoft OpenMP ランタイム・ライブラリー (vcomp) をリンクしないようにします。

```
C:¥> cl /MD /openmp hello.c /link /nodefaultlib:vcomp libiomp5md.lib
```

インテル® C++ コンパイラーとインテル® OpenMP 互換スタティック・ライブラリーを使用してビルドする

インテル® OpenMP 互換スタティック・ライブラリーを使用することもできます (推奨しません)。

```
C:¥> icl /MT /Qopenmp /Qopenmp-lib:compat hello.c
```

Microsoft コンパイラーとインテル® OpenMP 互換スタティック・ライブラリーを使用してビルドする

インテル® OpenMP 互換スタティック・ライブラリーを Microsoft コンパイラーで使用することもできます (推奨しません)。この場合、OpenMP ランタイム・ルーチンがダイナミック・ライブラリーからインポートされるように、Microsoft コンパイラーで使用される OMPIMP マクロを omp.h ヘッダーファイルで再定義する必要があります。次の例は、スタティック・ライブラリーの追加方法を示しています。

```
C:¥> cl /MT /openmp /D_OMPIMP= hello.c /link /nodefaultlib:vcomp
libiomp5mt.lib
```

1.2.4 複雑な例

一部のアプリケーションでは、Microsoft コンパイラー (cl) とインテル® C++ コンパイラー (icl) のように異なるコンパイラーを使用してコンパイルした場合でも、OpenMP オブジェクト・ファイルをリンクすることができます。

インテル® C++ コンパイラーを使用して混在オブジェクト・ファイルをリンクする

この例は、インテル® コンパイラーおよび Microsoft コンパイラーでコンパイルされたオブジェクト・ファイルとインテル® OpenMP 互換ライブラリーを、インテル® C++ コンパイラーでリンクする方法を示しています。

```
C:¥> cl /MD /openmp /c f1.c f2.c f3.c
C:¥> icl /MD /Qopenmp /Qopenmp-lib:compat /c f4.c f5.c f6.c
C:¥> icl /MD /Qopenmp /Qopenmp-lib:compat f1.obj f2.obj f3.obj f4.obj
f5.obj f6.obj /Feapp /link /nodefaultlib:vcomp
```

最初のコマンドは Microsoft コンパイラーで 3 つのオブジェクト・ファイルを生成します。2 番目のコマンドはインテル® C++ コンパイラーで 3 つのオブジェクト・ファイルを生成します。最後のコマンドは 6 つのオブジェクト・ファイルをアプリケーションにリンクします。

Microsoft リンカーを使用して混在オブジェクト・ファイルをリンクする

リンクフェーズでコンパイラーの代わりにリンカーを使用することもできます。

```
C:¥> link f1.obj f2.obj f3.obj f4.obj f5.obj f6.obj /out:app.exe
/nodefaultlib:vcomp libiomp5md.lib
```

Microsoft コンパイラーを使用して混在オブジェクト・ファイルをリンクする

この例は、インテル® コンパイラーおよび Microsoft コンパイラーでコンパイルされたオブジェクト・ファイルとインテル® OpenMP 互換ライブラリーを、Microsoft コンパイラーでリンクする方法を示しています。

```
C:¥> icl /MD /Qopenmp /Qopenmp-lib:compat /c f1.c f2.c f3.c
C:¥> cl /MD /openmp /c f4.c f5.c f6.c
C:¥> cl /MD /openmp f1.obj f2.obj f3.obj f4.obj f5.obj f6.obj /Feapp
/link /nodefaultlib:vcomp libiomp5md.lib
```

最初のコマンドはインテル® C++ コンパイラーで 3 つのオブジェクト・ファイルを生成します。2 番目のコマンドは Microsoft コンパイラーで 3 つのオブジェクト・ファイルを生成します。最後のコマンドはインテル® OpenMP 互換ダイナミック・ライブラリーを使用して 6 つのオブジェクト・ファイルをアプリケーションにリンクします。

Microsoft コンパイラーを使用して混在オブジェクト・ファイルをリンクする (プロシージャー間の最適化を使用)

この例は、インテル® コンパイラーでサポートされているプロシージャー間の最適化 (IPO) を使用して、インテル® コンパイラーおよび Microsoft コンパイラーでコンパイルされたオブジェクト・ファイルとインテル® OpenMP RTL を、Microsoft コンパイラーでリンクする方法を示しています。

```
C:¥> icl /MD /Qopenmp /Qopenmp-lib:compat /O3 /Qipo /Qipo-c f1.c f2.c
f3.c
C:¥> cl /MD /openmp /O2 /c f4.c f5.c f6.c
C:¥> cl /MD /openmp /O2 ipo_out.obj f4.obj f5.obj f6.obj /Feapp /link
/nodefaultlib:vcomp libiomp5md.lib
```

最初のコマンドはインテル® C++ コンパイラーで最適化されたマルチファイル・オブジェクト・ファイルを生成します。(`/Qipo` および `/Qipo-c` コンパイラー・オプションに関する詳細は、*インテル® C++ コンパイラーのドキュメント* または *インテル® Fortran コンパイラーのドキュメント* を参照してください。)
2 番目のコマンドは Microsoft コンパイラーで 3 つのオブジェクト・ファイルを生成します。最後のコマンドはインテル® OpenMP 互換ダイナミック・ライブラリーを使用して 4 つのオブジェクト・ファイルをアプリケーションにリンクします。

Microsoft コンパイラーとインテル® OpenMP スタティック・ランタイム・ライブラリーを使用してリンクする

この例は、インテル® コンパイラーおよび Microsoft コンパイラーでコンパイルされたオブジェクト・ファイルとインテル® OpenMP 互換スタティック・ライブラリーを、Microsoft コンパイラーでリンクする方法を示しています (推奨しません)。

```
C:¥> icl /MT /Qopenmp /Qopenmp-lib:compat /c f1.c f2.c f3.c
C:¥> cl /MT /openmp /D_OMPIMP= /c f4.c f5.c f6.c
C:¥> cl /MT f1.obj f2.obj f3.obj f4.obj f5.obj f6.obj /Feapp /link
/nodefaultlib:vcomp libiomp5mt.lib
```

最初のコマンドはインテル® C++ コンパイラーで 3 つのオブジェクト・ファイルを生成します。2 番目のコマンドは Microsoft コンパイラーで 3 つのオブジェクト・ファイルを生成します。最後のコマンドはインテル® OpenMP 互換スタティック・ライブラリーを使用して 6 つのオブジェクト・ファイルをアプリケーションにリンクします。

1.2.5 使用の制限

1 つのアプリケーションでは、1 つの OpenMP ランタイム・ライブラリーのみを使用してください。Microsoft OpenMP ライブラリー、インテル® OpenMP 互換ダイナミック・ライブラリー、またはインテル® OpenMP 互換スタティック・ライブラリー (推奨しません) のいずれか 1 つを使用できます。複数のライブラリーを組み合わせて使用すると、予期しない結果が生じる可能性があります。例えば、実行形式で Microsoft OpenMP ランタイム・ライブラリーがリンクされた DLL と、インテル® OpenMP 互換ダイナミック・ライブラリーがリンクされた別の DLL を使用すると、予期しない結果が生じることがあります。また、インテル® OpenMP 互換ライブラリーの 2 つのバージョン (スタティックとダイナミックの両方) をアプリケーションにリンクした場合にも、予期しない結果が生じる可能性があります。

インテル® OpenMP 互換ライブラリーは、インテル® Fortran コンパイラー 10.0 よりも前のバージョンで作成されたオブジェクト・ファイルとの互換性はありません。インテル® OpenMP レガシー・ライブラリーは、Microsoft コンパイラーでコンパイルされた OpenMP オブジェクト・ファイルとの互換性はありません。

1.3 インテル拡張ルーチン

インテル® コンパイラーは OpenMP 仕様の一部ではない OpenMP 機能を使用するために多くの拡張ルーチンを提供しています。これらの拡張ルーチンは接頭辞 `kmp_` で始まり、ほかのコンパイラーと共に使用することは想定されていません。

インテル拡張ルーチンを使用するコードはインテル® コンパイラーでコンパイルしてください。このインテル拡張ルーチンをインテル® コンパイラー以外のコンパイラーでコンパイルすると、予期しない結果が生じる可能性があります。拡張の全リストおよび拡張ルーチンの使用に関する詳細は、*インテル® C++ コンパイラーのドキュメント* または *インテル® Fortran コンパイラーのドキュメント* を参照してください。

1.4 インテル® OpenMP ライブラリー

次の OpenMP 互換ライブラリーは、インテル® OpenMP でコンパイルされたオブジェクトと Microsoft OpenMP でコンパイルされたオブジェクトの両方をサポートします。

ライブラリー	関連 DLL	説明	コンパイラー・オプション
libiomp5mt.lib	なし、スタティック・リンク	スタティック・パフォーマンス・ライブラリー	/MT /Qopenmp /Qopenmplib:compat
libiomp5md.lib	libiomp5md.dll	ダイナミック・パフォーマンス・ライブラリーおよび DLL	/MD /Qopenmp /Qopenmplib:compat
libiompprof5mt.lib	なし、スタティック・リンク	スタティック・プロファイリング・ライブラリー	/MT /Qopenmp-profile /Qopenmp-lib:compat
libiompprof5md.lib	libiompprof5md.dll	ダイナミック・プロファイリング・ライブラリーおよび DLL	/MD /Qopenmp-profile /Qopenmp-lib:compat
libiompstubs5mt.lib	なし、スタティック・リンク	スタティック・スタブ・ライブラリー	/MT /Qopenmp-stubs /Qopenmp-lib:compat
libiompstubs5md.lib	libiompstubs5md.dll	ダイナミック・スタブ・ライブラリーおよび DLL	/MD /Qopenmp-stubs /Qopenmp-lib:compat

次の OpenMP レガシー・ライブラリーは、インテル® OpenMP でコンパイルされたオブジェクトのみをサポートします。

ライブラリー	関連 DLL	説明	コンパイラー・オプション
libguide.lib	なし、スタティック・リンク	スタティック・パフォーマンス・ライブラリー	/MT /Qopenmp
libguide40.lib	libguide40.dll	ダイナミック・パフォーマンス・ライブラリーおよび DLL	/MD /Qopenmp
libguide_stats.lib	なし、スタティック・リンク	スタティック・プロファイリング・ライブラリー	/MT /Qopenmp-profile
libguide40_stats.lib	libguide40_stats.dll	ダイナミック・プロファイリング・ライブラリーおよび DLL	/MD /Qopenmp-profile
libompstub.lib	なし、スタティック・リンク	スタティック・スタブ・ライブラリー	/MT /Qopenmp-stubs
libompstub40.lib	libompstub40.dll	ダイナミック・スタブ・ライブラリーおよび DLL	/MD /Qopenmp-stubs

1.5 既知の問題

1.5.1 スレッドプライベートなオブジェクト

Microsoft コンパイラーは、インテル® コンパイラーとは異なるメカニズムを使用してスレッドプライベートなオブジェクトを参照します。変数をスレッドプライベートとして宣言したコードを、インテル® コンパイラーと Microsoft コンパイラーの両方でコンパイルすると、インテル® コンパイラーでコンパイルされたコードと Microsoft コンパイラーでコンパイルされたコードは、同じスレッドで参照された場合でも変数の異なる場所を参照します。

1.5.2 クリティカル構造

Microsoft コンパイラーとインテル® コンパイラーはどちらも、共通シンボルを使用して OpenMP クリティカル構造の実装に使用するロック・オブジェクトを指定します。このメカニズムにより、Microsoft コンパイラーとインテル® コンパイラー 10.1 でコンパイルされた異なるオブジェクト・ファイル間の排他制御が提供されます。Windows では、同じ共通シンボルが異なるコンパイルモジュール (例えば、異なる DLL、または実行形式に含まれる DLL とその実行形式が使用する DLL) の異なるオブジェクトを示します。したがって、like-named (または unnamed) のクリティカル構造は、使用されるコンパイラーに関係なく、異なるコンパイルモジュール間の排他制御を提供しません。これは、Microsoft コンパイラーおよびインテル® コンパイラーの両方における OpenMP クリティカル構造の実装の現在の制限です。

2 Linux* 互換ライブラリー

Linux 用のインテル® OpenMP* 互換ライブラリーにより、GNU* C/C++ コンパイラー (gcc) または GNU Fortran コンパイラー (gfortan) でコンパイルされた OpenMP オブジェクト・ファイルと、インテル® C++ コンパイラー (icc/icpc) またはインテル® Fortran コンパイラー(ifort) でコンパイルされた OpenMP オブジェクト・ファイルを組み合わせて使用できます。リンクフェーズで、ランタイム・ライブラリーの単一でコヒーレントなコピーが生成されます。

インテル® OpenMP ライブラリーのレガシーバージョンは、インテル以外のコンパイラーとの互換性をサポートしていませんが、インテル® コンパイラーの以前のバージョンを使用して作成されたオブジェクト・ファイルとの互換性のために含まれています。

ライブラリーの依存関係のため、インテル® Fortran コンパイラー (ifort) でコンパイルされたオブジェクトと、GNU Fortran コンパイラー (gfortran) でコンパイルされたオブジェクトは、アプリケーションのコンパイル時に -openmp オプションを使用したかどうかにかかわらずリンクはできません。

2.1 システム要件

インテル® OpenMP 互換ライブラリーは、インテル® C++ コンパイラー 10.1 およびインテル® Visual Fortran コンパイラー 10.1 に含まれています。使用する GNU コンパイラーは OpenMP をサポートしている必要があります。インテル® OpenMP 互換ライブラリーは、gcc および gfortan 4.2 と互換性があります。GNU コンパイラーの新しいバージョンとの互換性については、コンパイラーの新バージョンのリリース後にお知らせします。

2.2 使用モデル

インテル® コンパイラー 10.1 では、新しいオプション -openmp-lib をサポートしています。リンクフェーズで -openmp-lib compat オプションを指定すると、インテル® コンパイラーは、GNU コンパイラーで作成されたオブジェクト・ファイルと互換性のある OpenMP 互換ランタイム・ライブラリーのいずれかを使用します。デフォルト設定の -openmp-lib legacy オプションを指定した場合、インテル® コンパイラーは、GNU コンパイラーで作成されたオブジェクト・ファイルとは互換性のない OpenMP レガシー・ランタイム・ライブラリーのいずれかを使用します。

コンパイルフェーズでは、-openmp-lib legacy と -openmp-lib compat は等価です。

2.2.1 アクセス

インテル® C++ コンパイラーまたはインテル® Fortran コンパイラーを使用する前に、シェル環境を適切に設定する必要があります。インテル® コンパイラーには、環境を正しく設定するスクリプトが含まれています。

インテル® C++ コンパイラーでは、次のようなコマンドを入力します。

```
$ . <ifort_path>/ifortvars.sh                    # sh, bash
$ source <ifort_path>/ifortvars.csh              # csh
```

インテル® Fortran コンパイラーでは、次のようなコマンドを入力します。

```
$ . <ifort_path>/ifortvars.sh                    # sh, bash
$ source <ifort_path>/ifortvars.csh              # csh
```

詳細は、インテル® C++ コンパイラーのドキュメント またはインテル® Fortran コンパイラーのドキュメント を参照してください。

GNU コンパイラーで OpenMP コードを有効にするには、`-fopenmp` オプションを使用します。インテル® コンパイラーで OpenMP コードを有効にするには、`-openmp` オプションを使用します。C/C++ のコンパイルに、GNU C/C++ コンパイラーを使用する場合は GNU バージョンの `omp.h` を、インテル® C/C++ コンパイラーを使用する場合はインテル・バージョンの `omp.h` をそれぞれ使用してください。Fortran のコンパイルに、GNU Fortran コンパイラーを使用する場合は GNU バージョンの `omp_lib.h` または `omp_lib.mod` を、インテル® Fortran コンパイラーを使用する場合はインテル・バージョンの `omp_lib.h` または `omp_lib.mod` をそれぞれ使用してください。

GNU コンパイラーのドライバーは、インテル® コンパイラーがインストールされた場所を検索しません。GNU コンパイラーを使用する場合、`-L` オプションを明示的に使用して OpenMP 互換ライブラリーの場所を指定する必要があります (下記の例を参照)。

2.2.2 簡単な例

インテル® コンパイラーを使用してビルドする

この例は、インテル® OpenMP 互換ライブラリーをインテル® C++ コンパイラーで使用する方法を示しています。

```
$ icc -openmp -openmp-lib compat hello.c
```

次の例は、インテル® Fortran コンパイラーの等価なコマンドを示しています。

```
$ ifort -openmp -openmp-lib compat hello.f
```

GNU コンパイラーを使用してコンパイルし、インテル® C++ コンパイラーを使用してリンクする

GNU C コンパイラーで生成された OpenMP オブジェクト・ファイルとインテル® OpenMP 互換ラ

イブラリーをリンクする最も簡単な方法は、インテル® C++ コンパイラーをリンカーとして使用することです。

```
$ gcc -fopenmp -c foo.c
$ icc -openmp -openmp-lib compat foo.o
```

C++ のビルド

C++ ソースをコンパイルして、C++ オブジェクト・ファイルをリンクする場合、`gcc` と `icc` の代わりに `g++` と `icpc` を使用します。

```
$ g++ -fopenmp -c foo.cpp
$ icpc -openmp -openmp-lib compat foo.o
```

2.2.3 複雑な例

GNU を使用してコンパイルとリンクを行う

OpenMP 互換ライブラリー (`libiomp`) および POSIX スレッド・ライブラリーがコマンドラインで明示的に指定されていれば、GNU コンパイラーのドライバーを使用してリンクすることができます。GNU OpenMP ライブラリー (`libgomp`) が誤ってリンクされないようにするため、リンク時に `-fopenmp` オプションを省略します。GNU コンパイラーのドライバーは、インテル® コンパイラーがインストールされた場所を検索しないため、OpenMP 互換ライブラリーのパスを明示的に指定する必要があります。

```
$ gcc -fopenmp -c foo.c bar.c
$ gcc foo.o bar.o -liomp5 -lpthread -L<icc_dir>/lib
```

インテル® C と GNU オブジェクト・ファイル

インテル® C++ コンパイラーを使用してリンクします。

```
$ gcc -fopenmp -c foo.c
$ icc -openmp -c bar.c
$ icc -openmp -openmp compat foo.o bar.o
```

インテル® C++ コンパイラーを使用してリンクしない場合、OpenMP 互換ライブラリー (`libiomp`) と同じディレクトリーにある `libirc` (およびその他のインテル® ライブラリー) を明示的にリンクする必要があります。

```
$ gcc -fopenmp -c foo.c
$ icc -openmp -c bar.c
$ gcc foo.o bar.o -lirc -liomp5 -lpthread -L<icc_dir>/lib
```

C とインテル® Fortran

GNU C++ コンパイラー (gcc) またはインテル® C++ コンパイラーのいずれかでコンパイルされた OpenMP オブジェクト・ファイルは、インテル® Fortran コンパイラーで (インテル® Fortran コンパイラーでコンパイルされたオブジェクトと共に) リンクすることができます。

```
$ ifort -openmp -c foo.f
```

```
$ icc -openmp -c ibar.c
$ gcc -fopenmp -c gbar.c
$ ifort -openmp -openmp-lib compat foo.o ibar.o gbar.o
```

メインプログラムに ifort でコンパイルされた Fortran オブジェクト・ファイルが存在しない場合、リンク時に -nofor-main オプションを指定する必要があります。

C と GNU Fortran

同様に、インテル® C++ コンパイラー (icc) または GNU C++ コンパイラーのいずれかでコンパイルされたオブジェクト・ファイルは、GNU Fortran コンパイラー (gfortran) で (GNU Fortran コンパイラーでコンパイルされたオブジェクトと共に) リンクすることができます。

この場合、C ライブラリー (libirc) をインクルードする必要があります。リンク行で -fopenmp オプションを使用しないでください。

```
$ gfortran -fopenmp -c foo.f
$ icc -openmp -c ibar.c
$ gcc -fopenmp -c gbar.c
$ gfortran foo.o ibar.o gbar.o -lirc -liomp5 -lpthread -lc - L<icc_dir>/lib
```

インテル® C++ コンパイラーを使用して icc でコンパイルされた OpenMP オブジェクト・ファイルと、gfortran でコンパイルされたオブジェクトをリンクすることもできます。この場合、GNU Fortran ライブラリーを明示的に指定する必要があります。

```
$ gfortran -fopenmp -c foo.f
$ icc -openmp -c bar.c
$ icc -openmp -openmp-lib compat foo.o bar.o -lgfortranbegin -lgfortran
```

最後の 2 つのケースでは、メインプログラムに Fortran オブジェクト・ファイルが存在する必要はありません。

2.2.4 使用の制限

1 つのアプリケーションでは、1 つの OpenMP ランタイム・ライブラリーのみを使用してください。GNU OpenMP ライブラリー、インテル® OpenMP 互換ダイナミック・ライブラリー、またはインテル® OpenMP 互換スタティック・ライブラリー (推奨しません) のいずれか 1 つを使用できます。複数のライブラリーを組み合わせて使用すると、予期しない結果が生じる可能性があります。例えば、実行形式で GNU OpenMP ランタイム・ライブラリーがリンクされたダイナミック・ライブラリーと、インテル® OpenMP 互換ダイナミック・ライブラリーがリンクされた別のダイナミック・ライブラリーを使用すると、予期しない結果が生じることがあります。また、インテル® OpenMP 互換ライブラリーの 2 つのバージョン (スタティックとダイナミックの両方) をアプリケーションにリンクした場合にも、予期しない結果が生じる可能性があります。

インテル® OpenMP 互換ライブラリーは、インテル® Fortran コンパイラー 10.0 よりも前のバージョンで作成されたオブジェクト・ファイルとの互換性はありません。インテル® OpenMP レガシー・ライブラリーは、GNU コンパイラーでコンパイルされた OpenMP オブジェクト・ファイルとの互換性はありません。

2.3 インテル拡張ルーチン

インテル® コンパイラーは OpenMP 仕様の一部ではない OpenMP 機能を使用するために多くの拡張ルーチンを提供しています。これらの拡張ルーチンは接頭辞 `kmp_` で始まり、ほかのコンパイラーと共に使用することは想定されていません。

インテル拡張ルーチンを使用するコードはインテル® コンパイラーでコンパイルしてください。このインテル拡張ルーチンをインテル® コンパイラー以外のコンパイラーでコンパイルすると、予期しない結果が生じる可能性があります。拡張の全リストおよび拡張ルーチンの使用に関する詳細は、*インテル® C++ コンパイラーのドキュメント* または *インテル® Fortran コンパイラーのドキュメント* を参照してください。

2.4 インテル® OpenMP ライブラリー

次の OpenMP 互換ライブラリーは、インテル® OpenMP でコンパイルされたオブジェクトと GNU OpenMP でコンパイルされたオブジェクトの両方をサポートします。

ダイナミック・ライブラリー	スタティック・ライブラリー	説明	コンパイラー・オプション
libiomp5.so	libiomp5.a	パフォーマンス・ライブラリー	-openmp -openmp-lib Compat
libiompprof5.so	Libiompprof5.a	プロファイリング・ライブラリー	-openmp-profile -openmp-lib compat
libiompstubs5.so	libiompstubs5.a	スタブ・ライブラリー	-openmp-stubs -openmp-lib compat

次の OpenMP レガシー・ライブラリーは、インテル® OpenMP でコンパイルされたオブジェクトのみをサポートします。

ダイナミック・ライブラリー	スタティック・ライブラリー	説明	コンパイラー・オプション
libguide.so	libguide.a	パフォーマンス・ライブラリー	-openmp
libguide_stats.so	libguide_stats.a	プロファイリング・ライブラリー	-openmp-profile
libompstub.so	Libompstub.a	スタブ・ライブラリー	-openmp-stubs

2.5 既知の問題

2.5.1 スレッドプライベートなオブジェクト

GNU コンパイラーは、インテル® コンパイラーとは異なるメカニズムを使用してスレッドプライベートなオブジェクトを参照します。変数をスレッドプライベートとして宣言したコードを、インテル® コンパイラーと GNU コンパイラーの両方でコンパイルすると、インテル® コンパイラーでコンパイルされたコードと GNU コンパイラーでコンパイルされたコードは、同じスレッドで参照された場合でも変数の異なる場所を参照します。

2.5.2 クリティカル構造

クリティカル構造の名前が変更されたため、like-named のクリティカル構造は、インテル® コンパイラー 10.1 を使用してコンパイルされたオブジェクトと、インテル® コンパイラーの以前のバージョンを使用してコンパイルされたオブジェクト間の排他制御を提供しません。インテル® コンパイラー 9.1 およびそれ以前では、like-named クリティカル構造は異なるオブジェクト間の排他制御を提供しないという制限があります。この制限は、インテル® コンパイラー 10.0 および 10.1 ではなくなりました。そのため、これらのバージョンでは、like-named クリティカル構造はインテル® コンパイラーでコンパイルしたオブジェクト・ファイルと、GNU コンパイラーでコンパイルしたオブジェクト・ファイル間の排他制御を提供します。