



インテル® oneAPI ポーティング・ガイド ifx へ移行する ifort ユーザー向け

IntelCorporation

www.intel.com(英語)

著作権と商標について

注意事項:

この日本語マニュアルは、インテルコーポレーションのウェブサイト <https://software.intel.com/content/www/us/en/develop/articles/porting-guide-for-ifort-to-ifx.html> (英語) で公開されている『Porting Guide for ifort Users to ifx』(更新日 2021/08/25、2021.2 対応版)の参考訳です。

インテル社の許可を得て iSUS (IA Software User Society) が翻訳版を作成した iSUS の著作物です。

原文は Intel Corporation の Copyright であり、日本語参考訳版にも適用されます。

目次

著作権と商標について

はじめに

用語

IFX に対する指針

コンパイラーのデフォルトに関する大きな変更

コンパイラー・バージョン

オブジェクト・ファイルと .MOD ファイル

インテル固有のディレクティブのサポート

PGO、IPO、リンクに関する変更

IFX でサポートされない IFORT の機能

付録

著作権と商標について

インテルのテクノロジーを使用するには、対応したハードウェア、ソフトウェア、またはサービスの有効化が必要となる場合があります。

絶対的なセキュリティを提供できるコンピューター・システムはありません。

実際の費用と結果は異なる場合があります。

© Intel Corporation. Intel、インテル、Intel ロゴ、その他のインテルの名称やロゴは、Intel Corporation またはその子会社の商標です。

* 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。

最適化に関する注意事項

インテル® コンパイラーでは、インテル® マイクロプロセッサに限定されない最適化に関して、他社製マイクロプロセッサ用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサ依存の最適化は、インテル® マイクロプロセッサでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサ用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。

注意事項の改訂 #20110804

はじめに

このポーティング・ガイドは、新しい LLVM ベースのインテル® Fortran コンパイラー(`ifx`)に移行するインテル® Fortran コンパイラー・クラシック(`ifort`)ユーザー向けに情報と提案を提供します。DPCPP または ICX へ移行する ICC ユーザー向けポーティング・ガイドは[こちら](#)をご覧ください。

このポーティング・ガイドは、今後も更新されます。今後数カ月は頻繁に更新される予定ですのでご確認ください。

ベータ版インテル® Fortran コンパイラー(「本マテリアル」)は、同梱の開発ツール向けのインテル® エンドユーザー使用許諾契約書(2020 年 10 月版)(「EULA」)で定義される無料ユーザータイプでライセンスが供与されます。本マテリアルは、EULA のセクション 4.2 の「プレリリース・マテリアル」です。

EULA のセクション 4.2 にあるプレリリース・マテリアルに関する次の制限は、免除されます。(i) プレリリース・マテリアルの変更またはお客様の製品への組込み。(ii) 製品版リリース後のプレリリース・マテリアルの継続使用。(iii) プレリリース・マテリアルに関するベンチマーク、パフォーマンスの評価結果、またはその他の情報の第三者への開示。

お客様はこれらのプレリリース・マテリアルを使用する場合、(a)プレリリース・マテリアルはベータ品質レベルでのみ検証されており、エラーが含まれている可能性があり、より頻繁な更新が必要になることを認め、同意するものとします。(b)現在のパフォーマンスは、最終的な製品バージョンのパフォーマンスを示すものではありません。(c)これらのプレリリース・マテリアルの使用に起因する問題に関しては、お客様がすべての責任を負います。

用語

長い歴史を持つインテル® Fortran コンパイラーは、非公式に `ifort` と呼ばれます。2021.1 リリース以降では、正式にはインテル® Fortran コンパイラー・クラシックと呼ばれ、インテル® oneAPI HPC ツールキットに含まれます。

`ifx` と呼ばれる新しいコンパイラーは、2021.1 リリース以降のインテル® oneAPI HPC ツールキットに含まれます。

ifx に対する指針

ベータ版インテル® Fortran コンパイラー (ifx) は、LLVM バックエンド・テクノロジーを採用したインテル® Fortran コンパイラー・クラシック (ifort) フロントエンド、およびランタイム・ライブラリーをベースとする新しいコンパイラーです。ifx は、ifort がサポートしていない OpenMP* TARGET ディレクティブを使用するインテル GPU へのオフロードを実装することに関心のあるユーザー向けのベータ版としてリリースされています。

ifort から ifx への移行を容易にする努力はしていますが、お客様がアプリケーションをチューニングする労力が必要になる場合もあります。

ifx が ifort よりも優れたパフォーマンスと機能を提供できるまで、インテルが推奨するコンパイラーは ifort のままです。今後、ifx の機能とパフォーマンスが ifort と同等以上になった場合、インテルはパフォーマンスが重要なアプリケーションのコンパイルに ifort の代わりに ifx を推奨する可能性があります。その時期は未定です。それまで、インテル® oneAPI ツールキット・パッケージでは ifort と ifx の両方が提供されます。したがって、今後数年間は、パフォーマンスと安定性を求める場合は ifort を、OpenMP* を使用したサポートするインテル® GPU へのオフロードには ifx を使用できます。

コンパイラーのデフォルトに関する大きな変更

- インテル® Fortran コンパイラー・クラシックのユーザーは、引き続き ifort ドライバーを使用します。
- ベータ版インテル® Fortran コンパイラーのユーザーは、OpenMP* TARGET ディレクティブによりインテル® GPU へオフロードする追加のコンパイル機能を利用するため、ifx ドライバーを使用します。。

動作

- 浮動小数点数の比較で ifort と同じ動作になるようにするには、ifx ユーザーは、`assume [no]ieee_compares` や `assume nan_compares` を指定する必要があります。

- Fortran 2018 標準で要求される、浮動小数点比較向けの IEEE compareSignaling 操作を生成するようにコンパイラーに指示します。デフォルトでは、ifx は NaN が浮動小数点比較のオペランドにならないことを前提としており、NaN チェックを行いません。
 - Linux*
 - assume [no]ieee_compares
 - Windows*
 - /assume:[no]ieee_compares
- fp-model | fp および fast
 - fp-model | fp fast=1 または 2 オプションの動作は、浮動小数点比較では ifort と ifx は異なります。ifx で fp-model または fp fast オプションを使用する際に ifort と同じ動作を期待するには、ifx コマンドラインに assume nan_compares を指定する必要があります。
 - Linux*
 - fp-model fast=1 または -fp-model fast=2 -assume nan_compares
 - Windows*
 - /fp:fast 1 または /fp:fast 2 /assume:nan_compares

パフォーマンス

- ifx を使用する際に、ifort -O2 で行われたすべてのループ最適化を有効にするには、次のコンパイルオプションを追加します。
 - -Ofast -funroll-loops -nostandard-realloc-lhs -align array32byte

ifx OpenMP* オプション

- OpenMP* 4.5/4.5/5.0 の TARGET ディレクティブは、インテル® oneAPI HPC ツールキットに含まれる ifx のみが認識できます。ifx はまた、OpenMP* PARALLEL と SIMD ディレクティブを認識します。
- -fiopenmp または -qopenmp
 - OpenMP* parallel と SIMD ディレクティブと節を認識してコンパイルし、インテルの OpenMP* ランタイム・ライブラリーを使用します。

コンパイラーのデフォルトに関する大きな変更

- `-fopenmp`(非推奨)
 - OpenMP* `parallel` と SIMD ディレクティブと節を認識してコンパイルし、オープンソースの OpenMP* ランタイムを使用します。
 - パフォーマンスを目的としない互換性テストのみに使用してください。
 - パフォーマンスを向上し、インテル® コンパイラーの機能を使用するには、`-fiopenmp` または `-qopenmp` を使用します。
- `-fopenmp-targets=spir64`
 - OpenMP* 4.5/5.0/5.1 の TARGET ディレクティブを使用してインテル® GPU で実行する場合、このオプションを指定する必要があります。
 - `spir64` は、64 ビットの標準ポータブル中間表現です。
- 上記のコンパイラー・オプションを併用して、CPU とインテル® GPU 両方のファットバイナリーを生成します。
 - `ifx -fiopenmp -fopenmp-targets=spir64 code.F90`
 - `ifx -qopenmp -fopenmp-targets=spir64 code.F90`

重要なコンパイラー・オプションの対応付け

- `ifx` では多くの `ifort` コンパイラー・オプションを使用できます。
- すべての `ifort` オプションが `ifx` で許可/実装されるわけではないことに注意してください。
- `ifort` コンパイラーのドキュメントに明記されていないオプションは実装されません。また、今後も実装される予定はありません。`ifx` コンパイラーと `ifort` コンパイラーは大きく異なります。古い内部オプションやドキュメント化されていないオプションは作用せず、対応する `ifx` オプションはありません。ドキュメント化されていない内部オプションに必要なと思われる機能がある場合は、想定される動作と `ifx` でそれがサポートされていないことを[オンライン・サービス・センター\(OSC\)](#) (英語)に報告してください。「Makefile でこのオプションを使用しており、`ifort` ではこのオプションが使用できた」、ということは正当な理由とは認められません。`ifx` コンパイラーの最適化と動作は、`ifort` コンパイラー・クラシックとは異なります。最初はオプションを指定せずに `ifx` を試してみてください。
- 現時点で `ifx` で実装される予定がない `ifort` コンパイラー・オプションに対して、診断の警告が出力されます。以下に例を示します。

```
command line warning #10148: option '-ip' not supported.
```

- `ifx` オプション `-qnextgen-diag` を指定すると、`ifx` で使用できない `ifort` コンパイラー・オプションのリストを表示します。

- サポートされていないオプションがアプリケーションにとって重要であり、そのオプションを `ifx` のビルドオプションから除くとエラーになる場合は、[オンライン・サービス・センター \(OSC\)](#) (英語) に問題として報告してください。`ifx` に実装する必要があるオプションについて、皆様のご意見をお聞かせください。
- 実装済みまたは間もなく実装される予定の `ifort` コンパイラー・オプションに対してメッセージは表示されません。
- GNU* および Microsoft* 互換オプションは `ifort` と `ifx` で認識されます。

コンパイラー・バージョン

- `ifx` では新しいバージョンマクロがサポートされます。

- `__INTEL_LLVM_COMPILER`

- バージョン文字列

2つのコンパイラー `ifort` と `ifx` のバージョン文字列が新しくなりました。インテル® oneAPI のバージョン管理セマンティクスが使用されます。詳細は、[こちら](#) (英語) を参照してください。

例:

```
ifx -version
ifx (IFORT) 2021.2.0 Beta 20210317
```

形式は次のとおりです。

MAJOR.MINOR.PATCH (ビルド番号)

説明:

- MAJOR は、製品バージョン(年)です。必ずしも暦年と一致するとは限りません。
- MINOR は、"1" から始まる 1 桁のマイナーバージョン番号です。"1" は最初のリリースで、新しいマイナーリリースが提供されるたびに 1 ずつ増えます。
- PATCH は "0" から始まり、特定のバグとセキュリティーの問題が解決されるごとに数値が増加します。
- ビルド番号は YYYYMMDD 形式です。

オブジェクト・ファイルと .mod ファイル

ifx のバイナリー(.o/.obj)およびモジュール(.mod)ファイルは ifort と互換性があります。ifort で生成されたバイナリーとライブラリーは、ifx で生成されたバイナリーやライブラリーとリンクが可能であり、ほかのコンパイラーで生成された .mod ファイルを使用できます(64 ビット・ターゲットのみ)。icc、icl、icpc、icx、icpx、dpcpp で作成されたオブジェクト・ファイルはバイナリー互換であり、相互にリンクできます。

ただし、-ipo コンパイラー・オプションを使用してコンパイルする場合、ifort と ifx で生成されたオブジェクト・ファイルには互換性はありません。

インテル固有のディレクティブのサポート

一般的な非 OpenMP* コンパイラー・ディレクティブ(!DIR\$)は、リリースごとに追加されています。

PGO、IPO、リンクに関する変更

ifx コンパイラーは、プロシージャー間の最適化(IPO)とプロファイルに基づく最適化(PGO)に関して、コンパイラー・クラシックとは異なるアプローチを採用しています。

- PGO
 - 詳細は、[マルチステージ PGO](#) (英語)を参照してください。
- IPO
 - LLVM は、リンク時の最適化(英語)テクノロジーを採用しています。ifort では、「プロシージャー間の最適化(IPO)」と呼ばれます。
 - LLVM LTO の詳細については、[LLVM リンク時の最適化](#) (英語)の[設計と実装](#) (英語)を参照してください。
 - -ipo は、ifx ドライバーによって -flto に自動変換されます。
 - -ipo コンパイラー・オプションのバリエーションは削除されました。
 - ただし、-ipo コンパイラー・オプションを使用してコンパイルすると、ifort と ifx で生成されたオブジェクト・ファイル間には互換性はありません。

- インテルの `xi*` リンカーツール(`xilink`)は削除されました。
 - インテルのツール `xilink`、`xild`、および `xiar` は、インテル® コンパイラーが使用する独自のオブジェクト・ファイル形式向けであるため、`ifx` からは削除されました。
 - Makefile やプロジェクトの設定で `xilink` や `xild` を使用している場合、同等のネイティブリンカーに置き換えてください。同様に、`xiar` は `ar` などのアーカイバーに置き換えます。

ifx でサポートされない ifort の機能

- ガイド付き自動並列化
 - `ifort` で非推奨
- プロファイルに基づく最適化(PGO)は再設計されています。
- 投機実行の副作用の可能性があるコードを識別

付録

参考資料

次のリファレンスをご覧ください。

- [インテル® Fortran コンパイラー・クラシックおよびベータ版インテル® Fortran コンパイラー・デベロッパー・ガイドおよびリファレンス\(英語\)](#)
- [インテル® oneAPI プログラミング・ガイド:環境設定からアプリケーションのコンパイルと実行まで、プログラミングを始めるのに役立つ一般的なガイド](#)
- [ifx の Fortran 言語と OpenMP* 機能:コンパイラーのリリースごとのアップデート](#)
- [oneAPI 向けインテル® Fortran コンパイラー・リリースノート\(英語\)](#)
- [インテル® Fortran コンパイラー・クラシックの Fortran 言語標準のサポート](#)
- [インテル® Fortran コンパイラーの情報とよくある問い合わせ\(英語\)](#)
- [DPCPP または ICX へ移行する ICC ユーザー向けポーティング・ガイド](#)

製品とパフォーマンス情報

¹性能は、使用状況、構成、その他の要因によって異なります。詳細については、www.Intel.com/PerformanceIndex (英語)をご覧ください。