

インテル® スレッドチェッカー

Windows* 版

入門ガイド

インテル® スレッドチェッカーは、競合状態、デッドロック、ストールなど、スレッド化問題を特定します。解析中にエラーが発生しない場合でも、潜在的なエラーを特定することができます。スレッドチェッカーを使用して、特定の診断結果をフィルターしたり、クリティカル・ソースの場所を識別したり、並列ソフトウェアをより強固にするためのヒントを得ることができます。

概要

本ガイドでは、マルチスレッド・コード例を用いて、インテル・スレッドチェッカーでスレッド化問題を特定し、対処する方法について示します。本ガイドは、スレッドチェッカーを使用してコードを解析、修正することを目的としています。

本ガイドを印刷して、提供される例を参考に進めてください。

目次

著作権/法律に基づく表示	2
1. サンプルコードのビルド	3
2. データ収集	5
3. 結果の解析およびコードの修正	7
4. 次のステップ	10



著作権/法律に基づく表示

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 Sale』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）にも一切応じないものとします。インテル製品は、医療、救命、延命措置、重要な制御または安全システム、核施設などの目的に使用することを前提としたものではありません。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。

本資料で説明されているソフトウェアには、不具合が含まれている可能性があり、公開されている仕様とは異なる動作をする場合があります。現在までに判明している不具合の情報については、インテルのサポートサイトをご覧ください。

本資料およびこれに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、その使用および複製はライセンス契約で定められた条件下でのみ許可されます。本資料で提供される情報は、情報供与のみを目的としたものであり、予告なく変更されることがあります。また、本資料で提供される情報は、インテルによる確約と解釈されるべきものではありません。インテルは本資料の内容およびこれに関連して提供されるソフトウェアにエラー、誤り、不正確な点が含まれていたとしても一切責任を負わないものとします。

ライセンス契約で許可されている場合を除き、インテルからの文書による承諾なく、本資料のいかなる部分も複製したり、検索システムに保持したり、他の形式や媒体によって転送したりすることは禁じられています。

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。

Intel、インテル、Intel ロゴ、Intel Core、Itanium、Pentium、VTune、Xeon 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。

©2007 Intel Corporation.

改訂履歴

資料番号	リビジョン番号	説明	改訂日付
	001	初版	2005 年
313031	030	3.0 リリースのために更新。新しいテンプレートを適用。	2006 年 4 月
313031	3.0	マイナー・アップデート・バージョン	2006 年 7 月

1. サンプルコードのビルド

サンプルコード Primes は、1 ~ 1,000 の範囲の素数を割り出し、集計します。Windows スレッド化 API を使用して、マルチスレッドで実行しますが、スレッドが同時に同じメモリー・ロケーションにアクセスするため、潜在的な競合状態を引き起こします。その結果、不正確な結果を生成することがあります。

1. Microsoft* Visual Studio* でプロジェクト・ワークスペース・ファイル Primes.dsw を開きます。このプロジェクトはインテル・スレッドチェッカーとともにインストールされ、デフォルトでは次の場所に配置されます。

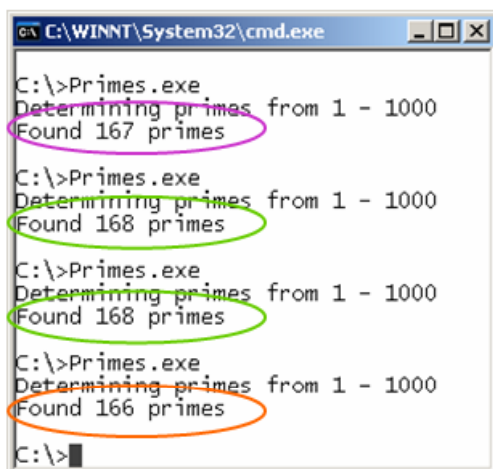
C:\Program Files\Intel\VTune\tcheck\Samples\Primes

2. Primes.dsw プロジェクトをビルドします。

このサンプル・プロジェクトは、シンボルをインクルードするための “/zi” オプション、最適化を無効にするための “/od” オプション、コードを移動可能にするためのリンクされた “/fixed:no” オプション、およびスレッドセーフなランタイム・ライブラリーを使用してビルドするための “MDd” オプションを使用するように設定されています。これらのオプションは、インテル・スレッドチェッカーが最も多くの情報 (エラーが発生した変数名や行番号など) を収集できるようにするために必要です。

ヒント オプションおよびアプリケーションのビルドに関するヒントは、スレッドチェッカーのヘルプを参照してください。

コマンドプロンプトからプログラム Primes.exe を何度か実行すると、次のような出力が表示されます。



```
C:\WINNT\System32\cmd.exe
C:\>Primes.exe
Determining primes from 1 - 1000
Found 167 primes
C:\>Primes.exe
Determining primes from 1 - 1000
Found 168 primes
C:\>Primes.exe
Determining primes from 1 - 1000
Found 168 primes
C:\>Primes.exe
Determining primes from 1 - 1000
Found 166 primes
C:\>
```






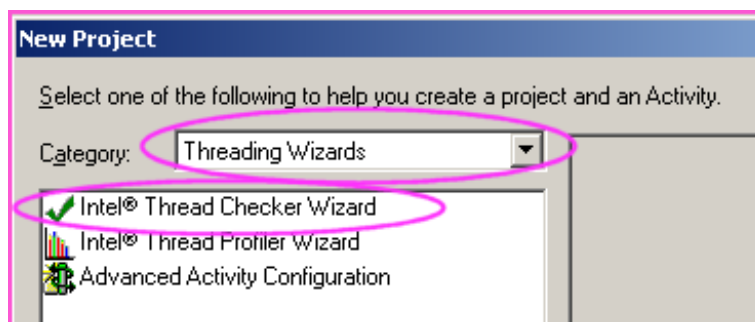
この出力結果をよく見ると、同じプログラムであるにもかかわらず、実行結果が異なっていることがわかります。正しい結果は “Found 168 primes (168 個の素数が見つかりました)” です。ここでは、スレッド化による不整合を比較的簡単に見つけることができましたが、大規模なプログラムでは、スレッド化による不整合を見つけることはより困難になります。

スレッドチェッカーは、スレッド化による不整合を特定します。

2. データ収集

インテル・スレッドチェッカー・ウィザードを使用してプロジェクトとアクティビティーを作成し、Primes.exe のデータを収集します。

1. デスクトップにあるインテル® VTune™ パフォーマンス・アナライザーのショートカット・アイコン  をダブルクリックして、インテル・スレッドチェッカー・プラグイン対応 VTune アナライザーを起動します。
2. 表示される [Easy Start (イージースタート)] ダイアログボックスまたはメインツールバーの [Easy Start (イージースタート)] ボタン  をクリックします。
3. [Category (カテゴリー)] ドロップダウン・リストで [Threading Wizards (スレッド化ウィザード)] を選択します。
4. 次に示すように、 [Intel(R) Thread Checker Wizard (インテル(R) スレッドチェッカー・ウィザード)] を選択します。



5. [Project Name (プロジェクト名)] を入力します (例: PrimesProject)。スレッドチェッカーによって、[Project Location (プロジェクトの場所)] にデフォルトのディレクトリーが設定されます。これは必要に応じて変更できます。
6. [OK] をクリックします。[Intel(R) Thread Checker Wizard (インテル(R) スレッドチェッカー・ウィザード)] が表示されます。
7. [Launch an application (アプリケーションを起動する)] で [...] をクリックし、ビルド済みの Primes.exe を指定します。デフォルトでは、Primes\Debug フォルダにあります。その他のオプションは、デフォルトの値でかまいません。



8. **[Finish (完了)]** をクリックして、ウィザードを終了します。






スレッドチェッカーは、アプリケーションのインストールメンターションを行い、インストール済み済みのアプリケーションを実行してデータを収集し、結果を **[Diagnostics (診断)]** リストに表示します。

デフォルトでは、**[Update diagnostics while the Activity is running (アクティビティー実行中に診断を更新する)]** オプションが選択されています。スレッドチェッカーは、アクティビティー実行中に新しい診断が利用可能になると通知します。この機能は、実行時間の長いアプリケーションで役立ちます。

これで、スレッドチェッカーを使用してスレッド化問題を特定することができます。

3. 結果の解析およびコードの修正

ウィザードが終了すると、次のような結果が表示されます。


Relation Sets	ID	Short Description	Severity	Description	Count	Filtered
1	1	Write -> Read data-race		Memory read at "Primes.cpp":43 conflicts with a prior memory write at...	42	False
1	2	Write -> Read data-race		Memory read at "Primes.cpp":44 conflicts with a prior memory write at "Primes.cpp":44 (flow...	42	False
1	3	Write -> Write data-race		Memory write at "Primes.cpp":44 conflicts with a prior memory write at "Primes.cpp":44 (output dependence)	42	False
1	4	Write -> Write data-race		Memory write at "Primes.cpp":43 conflicts with a prior memory write at "Primes.cpp":43 (output dependence)	1	False
2	5	Thread termination		Thread termination at "Primes.cpp":60 - includes stack allocation of 1048576 and use of 4096 bytes	1	False

Diagnostics Stack Traces Source View

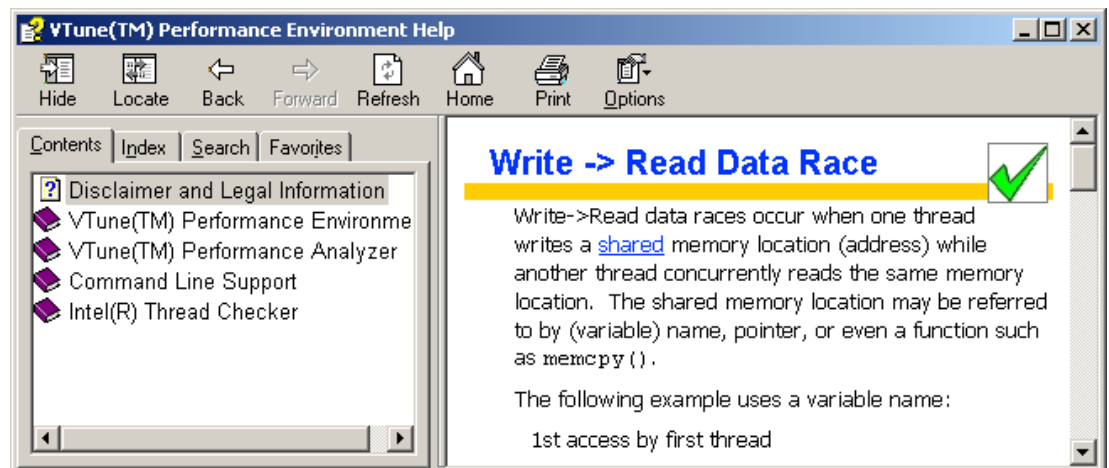
図 1: インテル・スレッドチェッカーの診断ビュー

これで、診断結果を解析し、アプリケーションにおけるスレッド化による不整合を修正する準備が完了しました。

メモ: 結果が表示されない場合、[Help (ヘルプ)] - [Search (検索)] を選択して、“Troubleshooting Thread Checker (スレッドチェッカーのトラブルシューティング)”を検索し、原因と解決方法を参照してください。

リストで赤くハイライトされ、ID 1 と示されている最初の診断データを見てみましょう。ここでは、**Write -> Read data-race (書き込み -> 読み取りの競合状態)** エラーが発見されました。赤いストップサイン・アイコン  によって示されるこのエラーは、適切に同期化されていない変数による共有メモリーへのアクセスが原因で引き起こされます。

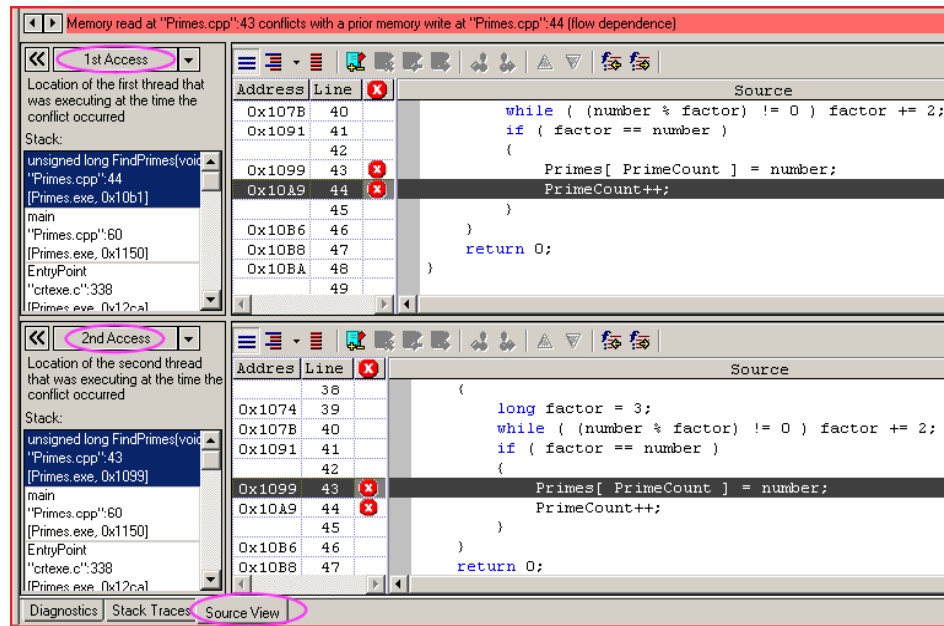
診断リストの最初の行を右クリックして、ポップアップ・メニューから [Diagnostic Help (診断ヘルプ)] を選択します。[Help (ヘルプ)] が表示され、診断の説明、原因、および解決方法が提供されます。



Write->Read data races (書き込み -> 読み取りの競合状態) は、1 つのスレッドが共有メモリーアドレスへの書き込みを行い、同時に別のスレッドが同じメモリーアドレスからの読み取りを行うことによって競合が発生し、引き起こされます。

メモ: コンテキストが似ている診断は、同じ **[Relation Set (関係セット)]** にグループ化されます。使用されたコンテキストを表示するには、グリッド上で右クリックして **[Show Column (列の表示)] - [Context (コンテキスト)] > [Context[Best] (コンテキスト [最高])]** を選択します。

ソースで競合箇所を確認するには、**[Source View (ソースビュー)]** タブをクリックします。上部にある **[1st Access (1 回目のアクセス)]** ペインでは、スレッドによって共有メモリーアドレスに書き込みが行われたソース箇所が表示されます。この場合、**44 行目**です。下部にある **[2nd Access (2 回目のアクセス)]** ペインでは、同期化されていない共有メモリーの読み取りが行われたソース箇所が表示されます。この場合、**43 行目**です。



Memory read at "Primes.cpp":43 conflicts with a prior memory write at "Primes.cpp":44 (flow dependence)

1st Access

Location of the first thread that was executing at the time the conflict occurred

Stack:

- unsigned long FindPrimes(void)
- "Primes.cpp":44
- [Primes.exe, 0x10b1]
- main
- "Primes.cpp":60
- [Primes.exe, 0x1150]
- EntryPoint
- "crtexe.c":338
- [Primes.exe, 0x12ca]

Address	Line	Source
0x107B	40	while ((number % factor) != 0) factor += 2;
0x1091	41	if (factor == number)
	42	{
0x1099	43	Primes[PrimeCount] = number;
0x10A9	44	PrimesCount++;
	45	}
0x10B6	46	}
0x10B8	47	return 0;
0x10BA	48	}
	49	}

2nd Access

Location of the second thread that was executing at the time the conflict occurred

Stack:

- unsigned long FindPrimes(void)
- "Primes.cpp":43
- [Primes.exe, 0x1099]
- main
- "Primes.cpp":60
- [Primes.exe, 0x1150]
- EntryPoint
- "crtexe.c":338
- [Primes.exe, 0x12ca]


Address	Line	Source
	38	{
0x1074	39	long factor = 3;
0x107B	40	while ((number % factor) != 0) factor += 2;
0x1091	41	if (factor == number)
	42	{
0x1099	43	Primes[PrimeCount] = number;
0x10A9	44	PrimesCount++;
	45	}
0x10B6	46	}
0x10B8	47	return 0;

Diagnostics Stack Traces Source View

競合の解決およびテスト結果

`CRITICAL_SECTION` を追加してソースコードを修正します。同期オブジェクトを追加して `Primes []` (43 行目) と `PrimeCount` (44 行目) をシリアル化することにより、時間分割するマルチスレッドにおける不整合や不正確さを除去することができます。

スレッドチェッカーによって特定された **Write -> Read data-race (書き込み -> 読み取りの競合状態)** が修正されていることを確認するには、次のように以前行った手順を繰り返します。

1. 修正済みコードをビルドします。
2. **F5** キーを押すか、または **[Run Activity (アクティビティーの実行)]** ボタン  をクリックして、修正済み `Primes.exe` のデータを収集します。
3. 結果を解析します。

どのような解析結果になりましたか? すべての競合状態が解決されていれば、青いアイコン



で示される診断情報のみ表示されます。

メモ: 競合状態の解決方法については、`PrimesFixed.cpp` のコードをご覧ください。また、`PrimesFixed.dsw` プロジェクトを使用してこのコードをビルドし、スレッドチェッカーで `PrimesFixed.exe` を解析して、競合状態が解決されていることを確認できます。

4. 次のステップ

スレッドチェッカーを最大限に活用するために、次のリソースも参考にしてください。

- **オンラインヘルプ:** 製品の詳細なユーザーズガイドです。オンラインヘルプでは、リモートデータ収集やその他の詳細設定オプションなど、追加機能についての情報を得ることができます。

F1 キーを押すとヘルプが表示されます。

- **サンプル:** 追加のコード例を利用することができます。本ガイドで取り上げていない他のスレッド化問題を特定し、対処する方法について理解するのに役立ててください。

サンプルは `tcheck\Samples` フォルダーにあります。また、`tcheck\Doc` フォルダーにある関連ドキュメントの **サンプルコード・ガイド** (`CodeExamplesGuide.pdf`) もあわせてご覧ください。

- **リリースノート:** 製品の簡単な概要が含まれています。動作環境、テクニカルサポート、および既知の制限事項に関する更新情報は、リリースノートを参照してください。

リリースノートを表示するには、**[スタート] - [すべてのプログラム] - [Intel(R)**

Software Development Tools (インテル(R) ソフトウェア開発ツール)] - [Intel(R)

Thread Checker (インテル(R) スレッドチェッカー)] - [Intel(R) Thread Checker

Release Notes (インテル(R) スレッドチェッカーのリリースノート)] を選択してください。

- **インテル® スレッド・プロファイラー:** インテル・スレッドチェッカーでコードを解析した後に、インテル・スレッド・プロファイラーを使用してパフォーマンスを向上させることができます。

スレッド・プロファイラーおよびその他のインテル® ソフトウェア開発製品に関する情報は、<http://www.intel.co.jp/jp/software/products/> を参照してください。