

Fine Kernel ToolKit システム (Visual Studio CLI 版) セットアップマニュアル

FineKernel Project

(2019/04/11 版)

1 Visual Studio のバージョンについて

Fine Kernel ToolKit システム (以下 FK) は C# や F# などの .NET Framework 対応言語から利用可能な「CLI 版」を提供している。現時点では Visual Studio(以下 VS) 2017 をサポートしている。本マニュアルでは、VS2017 Community を前提に記述していく。

2 前準備

インストールの前に、まず Windows Update をかけて OS を最新の状態にしておくこと。また、インストールに 10G を必要とするので、最低でも 12G 程度はドライブに空きを作っておく必要がある。

3 Visual Studio 2017 Community のインストール

Visual Studio 2017 Community (以下「VS」) のインストール方法は以下の通り。

1. まず、下記 URL にアクセスする。

<https://visualstudio.microsoft.com/jp/downloads/>

2. ページ内にある Visual Studio 2017 欄内の「コミュニティ」を選択する。
3. ダウンロードされるインストーラを起動する。
4. インストールする項目のうち、最低以下の 2 項目を選択しておく。
 - ユニバーサル Windows プラットフォーム開発
 - .NET デスクトップ開発その他の項目については、適宜自身が必要と思う項目を追加しておくといよい。
5. 本体のインストールが終わるまで待つ。数十分を要する。
6. 再起動が必要というダイアログが出た場合は、一度再起動する。
7. マイクロソフトアカウントを取得していない場合、事前に取得しておく。アカウントの設定を行わなくても使用はできるが、30 日で使用できなくなるので必ず取得して Visual Studio 内で登録しておくこと。
8. Visual Studio 2017 を起動する。初回の起動は数分かかることがあるので、しばらく待つ。
9. 起動を確認したら終了する。

4 マイクロソフトアカウントの設定

VS は、利用の際にマイクロソフトアカウントでサインインをしておく必要がある。既に Skype や OneDrive、あるいは Windows ストア等を利用して、マイクロソフトアカウントを取得済みの場合、それをそのまま用いてもよい。

マイクロソフトアカウントをまだ持っていない人は新規に取得する。以下のページにアクセスする。

<https://login.live.com/>

ここで「アカウントをお持ちでない場合、作成できます。」の「作成」の部分をクリックし、あとは指示に従う。大学のメールアドレスを利用することを推奨するが、その他のアドレスをアカウントとして利用しても、新規にメールアドレスを作成してもよい。

なお、マイクロソフトではアカウント作成処理が同一 IP アドレスから行われた場合、作成数を制限している。この処理のため、学内ネットワークから大勢が同時に作成処理を行うと、エラーとなってしまう。そのため、この作成は自宅で行うか、スマートフォンを用いて行うことを推奨する。

5 FK CLI 版のインストール

C# で FK のプログラミングを行うには、FK CLI 版をインストールする必要がある。これは以下の URL から「FK_CLI」で始まるインストーラ (exe ファイル) をダウンロードする。

<http://osdn.jp/projects/fktoolkit/releases>

ダウンロードの際、「ダウンロードしたユーザー数が少ないため、PC に問題を起す可能性があります。」というメッセージが表示される場合があるが、問題はない。

このインストーラを実行しようとする、「Windows によって PC が保護されました」というメッセージが表示される場合があるが、その場合は「詳細情報」をクリックして「実行」を選択する。あとは基本的に指示通りにインストールを進めればよい。

インストールが終わった後、一度サインアウトか再起動を行っておくこと。

6 開発の手順

VS 上で FK を用いた C# の開発は、まず「プロジェクト」を作成するところから始める。プロジェクトの作成方法は以下の手順で行う。

1. VS を起動する。
2. 「ソリューションエクスプローラー」が表示されていない場合は、メニューの「表示」→「ソリューションエクスプローラー」を選択して表示させておく。
3. 「ファイル」→「新規作成」→「プロジェクト」を選択する。
4. 「新しいプロジェクト」ダイアログの左側で「テンプレート」→「Visual C#」(または「テンプレート」→「その他の言語」→「Visual C#」) を選択する。
5. 同ダイアログの右側で「FK_CLI_Template」を選択する。
6. 名前を適当に入力し、場所の欄にプログラムソース等を置くフォルダを指定する。
7. 「OK」ボタンを押す。

これで準備は完了である。

7 ビルドと実行

上記手順で作成したプロジェクトには、最初から「Program.cs」にサンプルコードが記述されている。その状態でメニューの「ビルド」→「ソリューションのビルド」を選択することでビルド (コンパイル) ができる。その後、メニューの「デバッグ」→「デバッグの開始」で実行できる。ショートカットキーは、ビルドが F6、実行は F5 が割り当てられることが多いが、その他のショートカットが割り当てられることもあるので、メニューで確認しておくこと。

ビルドと実行の方法が確認できたら、Program.cs の内容を書き換えて開発を行っていく。

8 開発の中断と再開

開発を一旦中断する際は、「ファイル」→「全て保存」を選択した後、VS を修了すればよい。

開発を再開する際には、以下の 2 つの方法がある。

- VS を起動し、「ファイル」→「開く」→「プロジェクト/ソリューション」を選択する。その後、プロジェクトが置いてあるフォルダの「sln」ファイルを選択する。
- エクスプローラーにてプロジェクトを保存したフォルダを開き、そのフォルダの中にある「sln」ファイルを選択する。

9 開発に関する諸注意

9.1 ファイル読み込み処理を行うプログラムを実行する場合

プログラム上から画像や形状データなどを読み込むプログラムを作成した場合、カレントフォルダはビルドされた実行ファイルの存在しているフォルダになる。

9.2 ビルドしたファイルを他の環境で実行する場合

作成してビルドできたプログラムを他者の PC 環境で実行する場合は、bin フォルダ中の実行ファイルに加えて、FK_CLI.dll も実行ファイルと同じフォルダに配置する必要がある。Windows フォームや WPF を利用している場合は、FK_FormHelper.dll も必要である。更に、実行する PC に次の 3 つがインストールされている必要がある。

1. .NET Framework 4.5 以降
2. OpenAL(本パッケージインストール先の redist フォルダに収録)
3. Visual C++ 2017 再頒布可能パッケージ (本パッケージインストール先の redist フォルダに収録)

.NET Framework 4.5 は Windows Update か Microsoft のサイトから、OpenAL と Visual C++ 2017 再頒布可能パッケージは、本パッケージインストール先の redist フォルダに収録されている、oalinst.exe と vc_redist_x86.exe を実行することでインストール可能である。成果物を配布する際にはこのことを周知しておく必要がある。

9.3 デバッグ用ライブラリの利用

FK CLI 版が内部で利用している「C++/CLI」という技術はやや不安定な面があり、時折途中で処理が止まってしまうことがある。そのような場合、以下の手順で「デバッグ版」を利用することで、動作が正常となる場合がある。

1. VS で「プロジェクトエクスプローラー」を表示する。
2. 該当プロジェクトの「参照」から「FK_CLI」を削除する。
3. メニューの「プロジェクト」→「参照の追加」を選択する。
4. 表示される「参照マネージャー」というダイアログで、下側の「参照」というボタンを押す。
5. FK_CLI フォルダ (通常は C ドライブ直下) の下の「bin」→「x86」→「FK_CLI.d.dll」というファイルを選択する。

以上の手順で、実行時に通常版ではなくデバッグ版を参照するようになる。