

C. Visual C++ Toolkit 2003 と Platform SDK の使い方

1. Visual C++ Toolkit 2003 と Platform SDK

Visual C++ Toolkit (以下, VCTK) は Microsoft が無償で提供している開発環境で, これと Platform SDK を組み合わせる (以下, VCTK+PSDK) ことで C/C++ による開発環境を構築することができます。ただし, VCTK+PSDK では Visual Studio Ver.6 や Visual Studio .NET のような統合開発環境がありませんので, マウスで感覚的に操作するということはいけません。また, 多少コマンドプロンプト操作の知識を必要とします。

VCTK+PSDK をインストールして開発環境を設定して, list 1.1 のプログラムを作るまでの手順を順に説明していきます。

2. 開発環境を作る

2.1. Visual C++ Toolkit 2003 のインストール

参考資料のリンクから VCTK をダウンロードします。VCTK にはコンパイラ, リンカ, C/C++ のヘッダファイルとライブラリが含まれますので, これだけでも基本的なプログラムを作成できます。インストールはダウンロードした VCToolkitSetup.exe を実行するだけです。

2.2. Platform SDK のインストール

PSDK には Windows プログラム作成に必要なヘッダファイル, ライブラリファイルとリソースコンパイラが含まれています。また, 複数ソースファイルをまとめてコンパイル・リンクするためのメイクも含まれています。参考資料のリンクから PSDK をダウンロードします。ダウンロードできるファイルは, AMD64bit 用, Intel64bit 用, Intel 汎用とフルバージョンの分割ファイルおよび CD イメージ (ISO 形式) がありますので, 自分の環境に合ったものを選択します。インストールはダウンロードした実行ファイルを実行するだけです。

2.3. 環境設定用のバッチファイルを作る

VCTK と PSDK のインストールが済んだら実行ファイルへのパス, インクルードファイルへのパス, ライブラリファイルへのパスを環境変数に設定するために list C.1 のバッチファイルをメモ帳などで作成します。ファイル中の VCTK と PSDK はそれぞれインストールした場所に応じて書き換えて下さい。このファイルをパスの通ったフォルダもしくは windows フォルダに vctk.bat のような名前をつけて保存して下さい。これで開発環境は完成です。VCTK+PSDK では開発にコマンドプロンプトを使用しますが, その際は必ず初めに "vctk" を実行して, 環境変数の設定をするようにして下さい。うまくコンパイルなどが実行できないときはこのバッチファイルを見直して下さい。

```
@echo off
set VCTK=C:\Program Files\Microsoft Visual C++ Toolkit 2003
set PSDK=C:\Program Files\Microsoft SDK
set PATH=%VCTK%\bin;%PSDK%\bin;%PSDK%\bin\Win64;%PATH%
set INCLUDE=%VCTK%\include;%PSDK%\include
set LIB=%VCTK%\lib;%PSDK%\lib
```

list C.1 環境設定用バッチファイル

2.4. コマンドプロンプトの使い方

MS-DOS を使ったことのある人であればコマンドプロンプトは容易に使うことができます。使ったことのない人はエクスプローラと併用すると良いでしょう。その場合, 覚えておいた方が良いコマンドは, ドライブの移動とフォルダの移動, ファイルの一覧表示の3つだけで充分です。

- 1) ドライブの移動 — ドライブレターを入力します。(ex. A:, C:, ...)
- 2) フォルダの移動 — "cd <フォルダ名>" を入力します。スペースの入ったフォルダの場合はフォルダ名を"" で囲みます。また1階層上のフォルダに移動するときは "cd .." と入力します。(ex. cd program, cd "Documents and Settings", ...)
- 3) ファイルの一覧表示 — "dir" と入力すると現在のフォルダのファイル一覧を表示します。

また, 参考資料にある Open Command Window Here というツールをインストールしておくと, 好きなフォルダを右クリ

ックするだけで、そのフォルダでコマンドプロンプトが開くので便利です。

3. ソースファイルを作る

さて、下準備が終わったので、実際にソースを書いてプログラムを作成しましょう。VCTK+PSDK でソースファイルを書くにはエディタを使用します。使うエディタは Windows 標準のメモ帳、秀丸エディタ、などのようなものでもかまいません。自分の使い慣れたものを選ぶようにしましょう。では、エディタを使って list1.1 を書いて LIST1.cpp のような名前前で保存してみましょう。作成中に作業ファイルなどができるので C:\Program\CHAP1 のようにプログラム毎にフォルダを作っておくと便利です。

C/C++言語はスペースや改行を無視します。そのため、これらをうまく使うことで見やすいソースを書くことができます。見やすいソースはバグ(プログラムの問題点)を見つけやすくし、開発の効率を上げるために役立ちます。一般的なスタイルを挙げると、

1) インデント(字下げ)する。

「{」から「}」で囲まれたブロックや関数の範囲を明確にすることで、有効範囲やループの始端・終端をはっきりさせます。インデントには Tab キーを使います。

2) 処理の区切りに改行を入れる。

これもプログラム全体の流れをつかみやすくします。

3) 「,」の後ろ(場合によっては「(,」)の前後)にスペースを入れる。

式が繋がっているとどこからどこまでが変数かわかりづらくなります。適当にスペースを入れることで、それを解消します。

4) 長い行は適当なところで改行する。

ソースを書いている画面の横幅はそれほど大きくありません。横にはみ出してしまうと、スクロールした時にバグを見落としてしまう原因となります。

5) 一つの関数は 20~30 行程度に納め、適度に関数に分ける。

あまり長い関数だと初めの方で行った処理を忘れてたりして、新たなバグを生む原因となります。うまく関数を利用して見通しの良いソースを作るようにしましょう。

6) コメントをうまく利用する。

適度にコメントを入れるようにしましょう。後で見た時の理解の手助けになります。

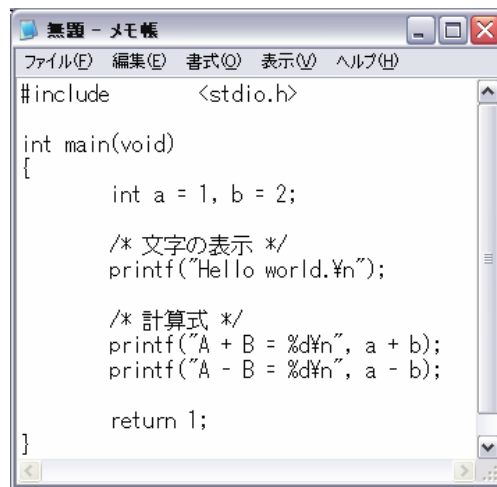


図 C.1 メモ帳でソースを書いたところ

4. ビルドする

作成したプロジェクト内のソースをそれぞれコンパイルし、リンクすることを「ビルド」と言います。ビルドすることで、実行ファイルが作成されます。今回作成した list1.1 もビルドして、きちんと書けているか確認をしてみます。

VCTK+PSDK でビルドするにはコマンドプロンプトを使用します。プログラムメニューからコマンドプロンプトを起動して、ソースファイルを保存したフォルダに移動するか、Open Command Window Here を使って保存したフォルダでコマンドプロンプトを開きます。移動したら、まず環境変数を設定するために準備で作成したバッチファイル(vctk)を実行します。

```
C:\Program\CHAP1> vctk
```

ビルドするには、コマンドラインでコンパイラ cl.exe を実行します。オプションを付けなければ自動的にリンクも呼び出してくれます。

```
C:\Program\CHAP1> cl list1.cpp
```

ビルドに成功すると、

```
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 13.10.3077 for
80x86
Copyright (C) Microsoft Corporation 1984-2002. All rights reserved.

list1.cpp
Microsoft (R) Incremental Linker Version 7.10.3077
Copyright (C) Microsoft Corporation. All rights reserved.

/out:list1.exe
list1.obj
```

のように表示されます。間違いがあった場合はエラーと行番号が表示されますので、それに合わせて修正しましょう。複数のソースファイルに分かれている場合には後に続けてソースファイル名を列記すれば自動的にビルドしてくれます。

5. 実行する

さて、ビルドがうまくいったら実行してみましょう。

```
C:¥Program¥CHAP1> list1
Hello world.
A + B = 3
A - B = -1
```

予想通りの答えになっているでしょうか？ 予想通りの答えになっていない時はソースのどこかに間違いがあるということです。このように、実行段階で間違いを発見するとその原因がどこなのか見つけるのがとても困難になります。また、結果が予想のつかない場合などは、間違いそのものを見つけることが困難になります。できるだけ、見やすいソースを書くことがバグを減らす一番のポイントです。

6. さらに

6.1. Make ファイルを使う

Windows アプリケーションのように複数のソースファイルを使うようになると、コマンドラインで毎回全てのソースを書いてビルドするのは不便になってきます。そのようなときは Make ファイルを使うと便利です。Make ファイルの書式は、list C.2 のように

ターゲット: 依存関係
コマンド

となっており、「依存関係」のファイルから「ターゲット」を作成するには「コマンド」を実行する。「依存関係」のファイルが存在しない場合はそのファイルが「ターゲット」になっている行を探して実行するという形式になっています。このファイルを他のソースファイルと同じフォルダに”Makefile”という名前で保存し、コマンドラインで、

```
C:¥Program¥CHAP1> nmake
```

と実行すると、コマンドラインに直接記述したときと同様の結果が得られます。

```
all: list1.exe

list1.exe: list1.obj
    link list1.obj

list1.obj: list1.cpp
    cl /c list1.cpp
```

list C.2 Makefile

6.2. フリーの開発環境を使う

VCTK+PSDK にフリーの開発環境を追加することでコマンドラインを使わなくても C/C++を使ったプログラミング環境を作ることができます。フリーの開発環境もいくつかありますが、ここでは CPad for Borland C++ Compiler (以下、BCPad) の設定法を設定について説明します。この開発環境はもともとは Borland C++ というコンパイラ用に開発されたものですが、設定を変更することで VCTK+SDK の環境でも使用することが可能です。他の開発環境もほぼ同様なので使いやすい環境を探してみてください。

1) ダウンロードとインストール

参考資料のリンクから BCPad をダウンロードします。ダウンロードしたファイルは圧縮ファイルになっており、適当な

フォルダに Lhasa や LHMelt などを用いて解凍するだけでインストール作業なしに使用することができます。解凍したフォルダの中の BCPad.exe を実行してみましょう。図 C.2 の様な画面が出てくるはずですよ。上がエディタ画面、下がコンパイル時のメッセージウィンドウとなっています。BCPad.exe へのショートカットなどを作っておくと便利でしょう。

2) 環境設定

[実行(R)]-[設定(S)...]から VCTK+SDK に関する設定をしておきましょう。まず、[基本設定]タブのコンパイラのパス(W)にはインストールした VCTK フォルダの bin にある cl.exe を指定します。あと、[高度な設定]タブを開き、[以下の設定を変更する(C)]をチェックして、

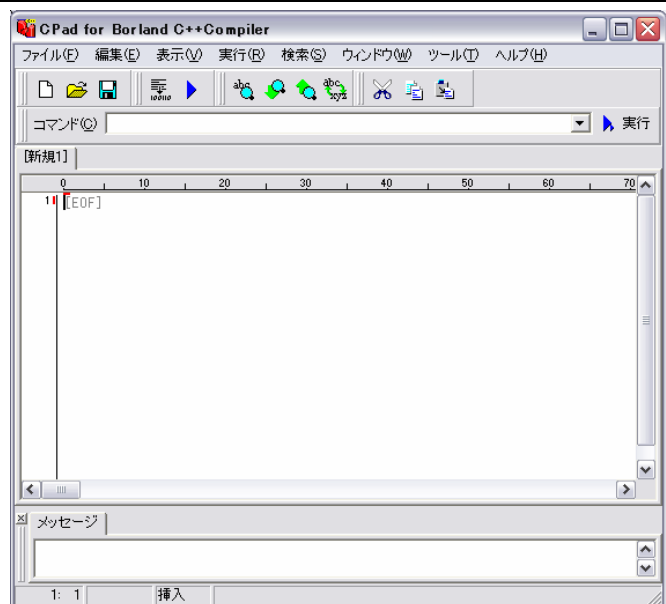


図 C.2 BCPad

[追加する環境変数(C)]

```
VCTK=C:\Program Files\Microsoft Visual C++ Toolkit 2003
PSDK=C:\Program Files\Microsoft SDK
INCLUDE=%VCTK%\include;%PSDK%\Include
LIB=%VCTK%\lib;%PSDK%\Lib
```

[追加するパス(P)]

```
%VCTK%\bin
%PSDK%\Bin
%PSDK%\Bin\Win64
```

のように設定しておきましょう。

3) 使い方

エディタウィンドウにプログラムのソースを書いて、[実行(R)]-[コンパイル(C)]でコンパイル、[実行(R)]-[実行(R)]で実行することができます。エラーなどはメッセージウィンドウに表示されます。