

コンピュータグラフィックスS 演習資料

サンプルプログラムのコンパイル方法

九州工業大学 情報工学部 システム創成情報工学科

講義担当：尾下真樹

1. 開発環境

本講義の演習は、AV 講義室で Linux を使用して行う。本資料では、主に、Linux 環境で gcc を使用してサンプルプログラムのコンパイルを行う方法について説明する。

また、講義以外の時間に自分で演習を行いたい人は、CL 端末の Windows 環境や、自宅のパソコンを使用することもできる。本資料では、Windows 環境で Visual Studio 2010 を使用してコンパイルを行う方法についても、簡単に説明する。ただし、講義中は、Linux 環境のみで演習を行う。(自力のできる人のみ、講義後に Windows で演習を行っても良い。)

2. Linux 環境 (AV 講義室) でのコンパイル

ターミナル上で、ソースファイルが置かれているディレクトリに移動し、以下のコマンドを入力することで、GNU C コンパイラ (gcc) を使用して C プログラムをコンパイルし、実行形式のファイルを生成することができる。

```
gcc opengl_sample.c -L/usr/X11R6/lib -lglut -lGLU -lGL -lXmu -lm -o opengl_sample
```

コマンドの引数にある、`opengl_sample.c` と `opengl_sample` は、入力のソースファイル名と、出力の実行形式のファイル名を表している。必要に応じて適切なものに変更する。(出力ファイル名を省略すると、`a.out` というファイル名で出力される。)

また、`-L/usr/X11R6/lib` は、ヘッダファイルを検索する追加ディレクトリを指定している。`-lglut -lGLU -lGL -lXmu -lm` は、それぞれ、`glut`, `glu`, `OpenGL`, `X` ウィンドウシステムのユーティリティライブラリ、算術計算ライブラリをリンクすることを指定している。これらのオプションについては、本演習で扱うプログラムでは全て共通なので、変更する必要はない。

コンパイルに失敗すると、エラーメッセージが表示される。このとき、コンパイルできない原因や、ソースファイルの何行目でそのエラーが発生したか、といった情報が表示されるので、それらを参考に、ソースファイルを修正する。

コンパイルに成功すると、出力ファイル名で指定したファイルが生成されるので、以下のようにターミナルからコマンドを入力して、コンパイル結果のプログラムを実行する。

```
opengl_sample
```

3. Windows 環境 (CL 端末室) でのコンパイル

ここでは、学科の CL 端末にインストールされている、Visual Studio 2010 (C/C++のプログラムを開発するための統合環境) を使って、サンプルプログラムをコンパイルする方法を説明する。
講義時間外に、CL 端末室で Windows を使って演習を行いたい人は、参考になると良い。

3.1. 文字コード・改行コードの変換

Linux と Windows では、テキストファイルの 2 バイト文字コードや改行コードが異なるため、普通にファイルをコピーしても、コンパイルすることができない。日本語の文字コードとして、Windows ではシフト JIS、Linux では EUC が使用されている。また、改行コードとして、Windows では CR+LF、Linux では LF が使用されている。

Linux にインストールされている `nkf` コマンドなどを使うことで、文字コードや改行コードを変換できる。(Windows にも、文字コード変換のためのフリーソフトなどが多くあるが、標準では付属していないので、自分でインストールする必要がある)

Linux 用の文字コード (EUC、LF) から、Windows 用の文字コード (S-JIS、CR+LF) へ変換する方法。

```
$ nkf -s -Lw opengl_euc.c > opengl_sjis.c
```

Windows 用の文字コード (S-JIS、CR+LF) から、Linux 用の文字コード (EUC、LF) へ変換する方法。

```
$ nkf -e -Lu opengl_sjis.c > opengl_euc.c
```

3.2. コンパイル&実行の方法

Visual Studio などの統合環境では、プログラムを構成するソースファイルやライブラリ、コンパイルオプションなどの情報を「プロジェクト」として作成する必要がある。

以下、サンプルソースをコンパイルするためのプロジェクトの作成手順を示す。

1. Visual Studio 2010 を起動
2. メニューから **ファイル**→**新規作成** を選ぶ
3. 新規作成ダイアログ
 - (1) 上のタブから "プロジェクト" を選択
 - (2) プロジェクトの種類を Visual C++ → Win32 の中の "Win32 Console Application" にする
 - (3) プロジェクトの名前を適当につける (opengl_smaple など)
 - (4) プロジェクトを保存する場所を自分の MyDocuments の下に適当に設定
 - (5) "ソリューションのディレクトリを作成" のチェックは、外した方がよい
 - (6) "OK" をクリックして次に進む
4. アプリケーションの設定
 - (1) プロジェクトの種類は、"コンソールアプリケーション" とする
 - (2) 空のプロジェクトを選択して、"完了" をクリック
5. ここまでの手順で、空のプロジェクトが作成された状態で Visual Studio に戻ってくる
6. 手順 3 で作成されたプロジェクトのディレクトリにソースファイルをコピーする

※ ここで、ファイルの拡張子は、`c` ではなく、`cpp` に変更した方がよい (拡張子が `c` のままだと、C++ で拡張された変数定義の方法などが有効にならず、コンパイルエラーが出るため)
7. プロジェクトにソースファイルを追加する

メニューから **プロジェクト**→**既存項目の追加** で、ソースファイルを選ぶ

8. コンパイルしてみる
メニューから **ビルド→ソリューションのビルド** を選択 (または、F7 キーを押す)
9. コンパイルが成功したら、実行してみる
メニューから **デバッガー→デバッグの開始** を選択 (または、F5 キーを押す)

4. Windows 環境 (自宅) でのコンパイル

自宅で Visual Studio を使用している人は、GLUT をインストールすれば、サンプルプログラムのコンパイルを行うことができる。

(CL 端末室のパソコンには、既にインストールされているので、自分でインストールする必要はない。)

4.1. GLUT のインストール (Visual Studio を使用する場合)

OpenGL のソースファイルやライブラリは、Visual Studio と同時にインストールされる。

しかし、GLUT はコンパイラや OpenGL の標準機能ではなく、一緒にインストールされないので、基本的には自分でインストールする必要がある。

まず、GLUT をダウンロードする。講義のページにもコピーを置いている。

<http://www.cg.ces.kyutech.ac.jp/lecture/cg/opengl/glut-3.7.6-bin.zip>

上のファイルは、下記の GLUT のページからもダウンロードできる。

<http://www.opengl.org/developers/documentation/glut.html>

<http://www.xmission.com/~nate/glut.html>

GLUT をダウンロードしたら、解凍して、以下の 3 つのファイルをコピーする。

- ヘッダファイル (glut.h) を C:\Program Files\Microsoft Visual Studio 8\PlatformSDK\Include\GL にコピー
- ライブラリファイル (glut32.lib) を C:\Program Files\Microsoft Visual Studio 8\PlatformSDK\Lib にコピー
- ダイナミックライブラリ (glut32.dll) を C:\Windows\System にコピー

上のフォルダは、Visual Studio 2005 を標準の設定でインストールした場合、Visual Studio のインストール場所を変更した場合は、それに応じて上のフォルダも変更する必要がある。

また、Visual Studio 6 を使っている場合は、ヘッダファイルとライブラリファイルは、C:\Program Files\Microsoft Visual Studio\VC98 の下の各フォルダに、コピーする。同じく、Visual Studio .NET を使っている場合は、C:\Program Files\Microsoft Visual Studio .NET\vc7\PlatformSDK の下の各フォルダに、コピーする。

(使用している Visual Studio のバージョンによって、細かいフォルダ名は異なるので、適宜読み替えること。)