

## 5.1 開発環境の準備とプログラム開発の流れ

プログラムを書く、というと…まずなにをすれば良いのか、ピンとこない方も多いでしょう。実は、通常の文章を書くのと同じように、テキストファイルにプログラムを書いていくことになります。Windows、Mac、スマートフォン、どんな端末にもメモ帳のようなものが入っていますね。Raspberry Pi でも、テキストエディタが用意されていますので、それを利用しましょう。開発の流れは以下のようになります。

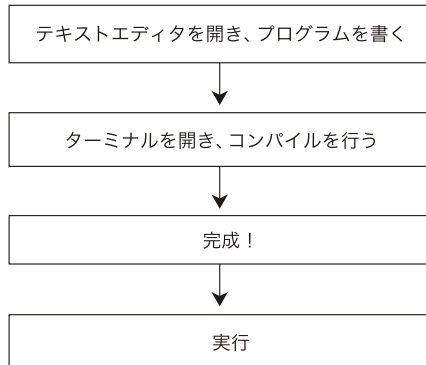


表 5.1.1

## 5.2 "Hello World!" に挑戦!

まず、デスクトップ画面左上の [Menu]→[ アクセサリ ]→[LX Terminal] を選び、ターミナルを起動します。基本的に、C 言語プログラミングの作業はこのターミナル上で行います。コマンドを一行ずつ打ちこみ、[Enter] を押して実行して下さい。

コマンドリスト 5.2.1

```
pi@raspberrypi ~ $ cd /home/pi  
/home/pi ディレクトリに移動します。
```

コマンドリスト 5.2.2

```
pi@raspberrypi ~ $ mkdir programs  
programs というディレクトリを作成します。
```



### コマンドリスト 5.2.3

```
pi@raspberrypi ~ $ cd programs
今作った programs ディレクトリの中へ移動します。
```

#### ワンポイント

「cd」「mkdir」とは？

「cd」は change directory の略で、ディレクトリの移動に利用します。「mkdir」は make directory の略で、新規にディレクトリ作成する場合に使います。ディレクトリとは、フォルダのことです。表 5.2.1 にディレクトリ、ファイル操作の基本コマンドを示します。

表 5.2.1 ディレクトリ操作の基本コマンド

cd	ディレクトリの変更
mkdir	ディレクトリの作成
pwd	カレントディレクトリ【現在作業を行っているフォルダ】の確認
ls	ディレクトリ内のファイル一覧

では、今の位置を「pwd」で確認してみましょう。

### コマンドリスト 5.2.3

```
pi@raspberrypi ~/programs $ pwd
```

```
pi@raspberrypi:~/programs $ pwd
/home/pi/programs
```

となっていたでしょうか。このディレクトリ内に、プログラムファイルを作っていきます。

### コマンドリスト 5.2.4

```
pi@raspberrypi ~/programs $ sudo nano test.c
```

ここでは、操作性がよい nano というテキストエディタを使用します。

test.c というファイルが存在すればそれを開き、なければ作成する、というコマンドになっています。nano が実行され、テキストが入力できる状態になりますので、実際にプログラムを打ち込んでいきましょう。



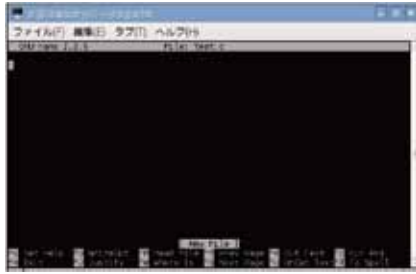


図 5.2.1 nano を起動する

## ソースリスト 5.1

```
//*****  
// 作成者、作成日付、ver など  
// このプログラムの概要を記入する  
//*****  
#include <stdio.h>  
int main () {  
    printf(" Hello World ! \n" );  
}  
/*****  
*  
*  
***** /
```

ターミナル上に「Hello World！」と表示させるプログラムです。

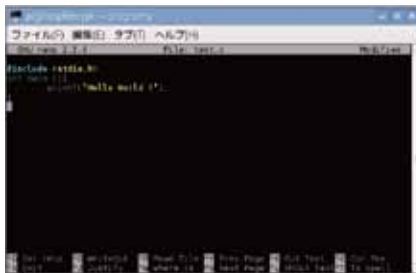


図 5.2.2 入力後イメージ



上記を打ち込んだら、ファイルをセーブしましょう。

[Ctrl] + [O] キーを押し、保存先を確認して（今回は変更しなくて大丈夫です） [Enter] を押します。

これで test.c ファイルに書き込まれましたので、

いよいよコンパイルを行っていきましょう。

[Ctrl] + [X] を押し、nano を終了します。

続いて、以下のコマンドを入力して [Enter] を押下します。

コマンドリスト 5.2.5

```
pi@raspberrypi ~/programs $ sudo gcc test.c -o test
```

test という名前の実行ファイルを作成せよ、というコマンドです。

成功すると、以下のようにエラーメッセージが出ずに終了します。

```
pi@raspberrypi ~/programs $ sudo gcc test.c -o test  
pi@raspberrypi ~/programs $
```

図 5.2.3

入力したプログラムに書き間違いなどがある場合、以下のようにエラーが出ます。

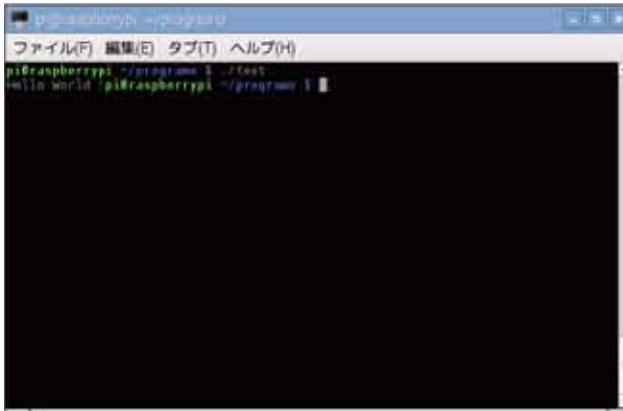
```
pi@raspberrypi ~/programs $ sudo gcc test.c -o test  
/tmp/ccPdtDDs.o: 関数 `main' 内 :  
test.c:(.text+0xc): `print' に対する定義されていない参照です  
collect2: ld returned 1 exit status
```

図 5.2.4



いかがでしょうか？

ターミナル上に「Hello World！」と表示されれば成功です。



```
pi@raspberrypi ~$ cd /home/pi
pi@raspberrypi ~$
pi@raspberrypi ~$ cd programs
pi@raspberrypi ~/programs$
pi@raspberrypi ~/programs$ echo Hello World!
Hello World!
```

図 5.2.5 実行イメージ

もしエラーなどが出た場合、入力したプログラムの内容やコマンドを間違えていないか、ディレクトリを間違えていないか確認しましょう。

### 5.3 「if 文」を使ってゲームを作ってみよう

次に、実際になにか遊べるプログラムを作ってみましょう。

ここでは、簡単な足し算ゲームを作成します。

if 文というのは、「もし〜であった場合、特定の動作をする」という条件分岐を行うためのプログラムです。

前回同様、デスクトップ画面左上の [Menu]→[アクセサリ ]→[LX Terminal] を選び、ターミナルを起動します。

#### コマンドリスト 5.3.1

```
pi@raspberrypi ~ $ cd /home/pi
/home/pi ディレクトリに移動します。
```

#### コマンドリスト 5.3.2

```
pi@raspberrypi ~ $ cd programs
programs ディレクトリの中へ移動します。
```

