

Welcome to FLS!!

FLSとはWindowsとUbuntu22.04を対象としたプロセス間通信ライブラリです。

環境構築とコーディングの容易さを両立することを目的として開発しました。使用言語はRustで「Rustを先日初めて知ったよ！」みたいなレベルでも開発可能であると自負しております。

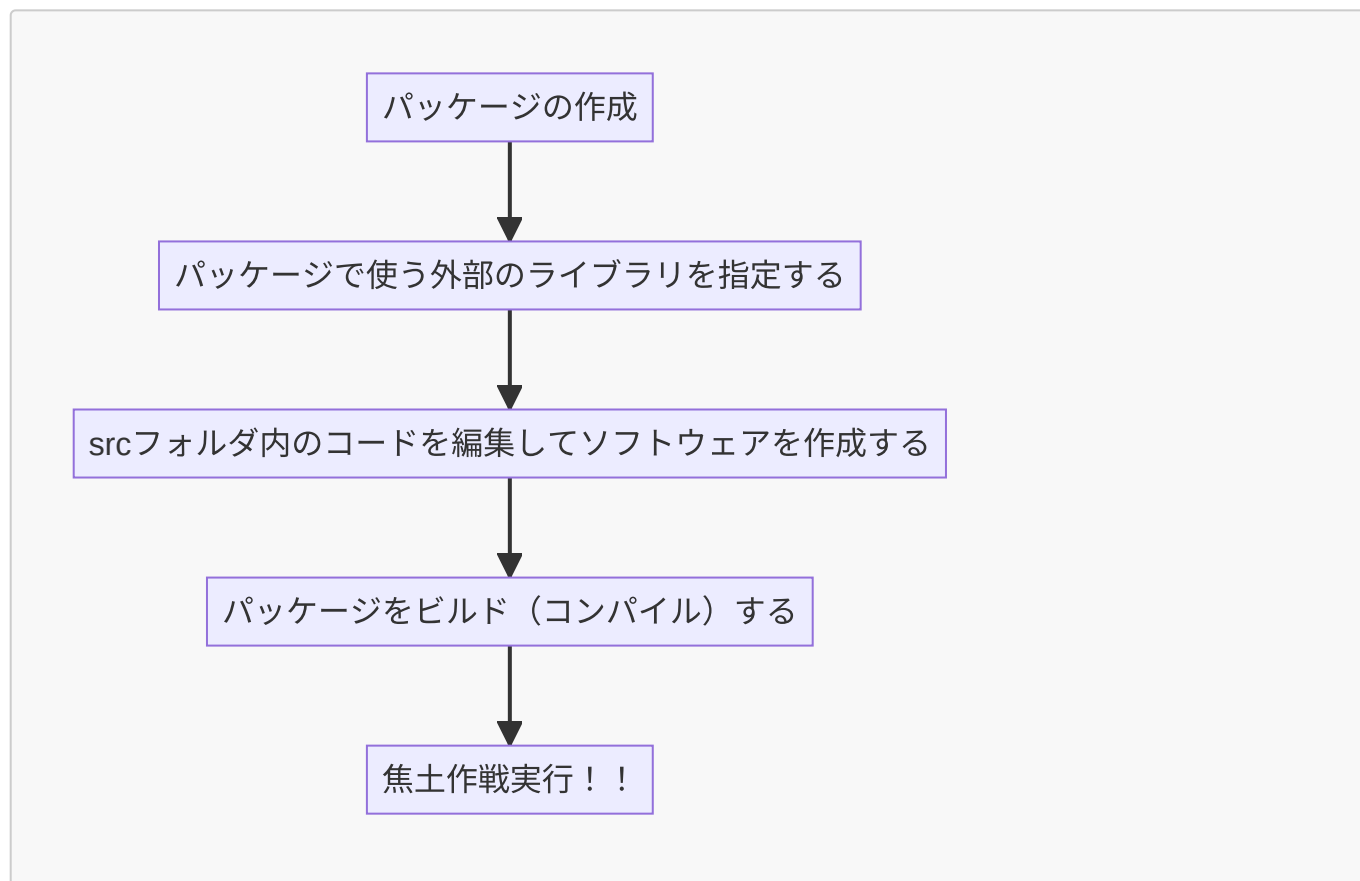
この資料を読んで**FlexLinkSystem**、通称**FLS**への理解を深め開発に参加してくれることを願っています。

このライブラリの意味

ずばり、このライブラリがどのような機能を持つのかということについての解析です。

プログラミングをするときには色々な開発方法があると思いますが、このライブラリでは共同開発に適した開発方法であるパッケージ開発をサポートしております。

パッケージ開発ってなんぞや



上記にパッケージ開発の核となるような部分について示しました。

ここでいう**パッケージ**とは**パッケージ**で使いたいライブラリを指定しているファイルとソフトウェアを構築するソースコードを含んだフォルダのことを示します。

また**ライブラリ**とは外部のソースコードを示します。ソフトウェア開発の際には0から100まですべて自分で書いているというわけではありません。開発を高速に行うためにも**誰かが書いたことのあるコードは引用したいものです**。そんなときにソフトウェアにおける**ライブラリ機能**が役立ちます。**人が書いたコードを引用するときにライブラリを用いるということです**。

最後にビルドについてです。通常、授業などでプログラムを機械語に翻訳することをコンパイルと呼ぶと思います。ですがパッケージで開発しているときには先述したライブラリをインストールしたりする操作も含まれます。よってそれらの操作をまとめてビルドと呼んでいるのだなと私は理解しています。

プロセスとスレッド

プロセスとスレッドについても軽く解説しておきます。

プロセスとはソフトウェアの単位のようなものでC言語なら以下のような感じで1 プロセスが実行されます。

```
#include <stdio.h>

int main(void)
{
    int index = 0;
    while(index > 100)
    {
        printf("Hello, FlexLinkSystem!!\n");

        index++;
    }

    return 0;
}
```

こちらのコードでは「Hello, FlexLinkSystem!!」が連呼されることでしょう。これは1 プロセスが動くパターンのプログラムです。

次にスレッドが複数あるときのプログラムです。

```
#include <thread>

void thread1()
{
    int index = 0;

    while(index > 99)
    {
        printf("Hello, MultiThread\n");

        index++;
    }
}

void thread2()
{
    int index = 0;
    while(index > 198)
    {
        printf("Don't Move!!\n");
    }
}
```

```
        index++;  
    }  
}  
  
int main(void)  
{  
    std::thread th1(thread1)  
    std::thread th2(thread2)  
  
    th1.join()  
    th2.join()  
}
```

このプログラムでは「Hello, MultiThread」と「Don't Move!!」が交互（ほぼ同時）に出力されるはずです。ここで注目してほしいのが***index***という変数です。なんと同じ名前の変数ではあるもののスレッドが分かれているため全く別の変数です。また、1プロセスの中に2つのスレッドがあるので2つのループが同時に動いているんですね。