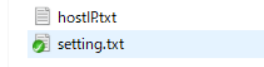


■iniフォルダを用意し、中に設定関連のファイルを入れておこう。



■hostIP.txt

履歴をためて、選択できるようにする。

1行ずつIPアドレスを格納する。

■setting.txt

<byte length="1400">☒

送信バイト長を記述する。

内部的には4で割った値で使用する。

NetWorkクラスのコンストラクタにて、読み込み以下のプライベート変数を初期化する。

unsigned int intSendCount_;

■データ送信パケットの作り方の変更。

ヘッダーを作って渡す方式から、MesTypeのみ、もしくはMesTypeとMesPacketを渡す方法に変更する。

それによって、データ送信時にMesType以外を生成して送信する。

規定のデータ送信サイズを超えている場合、送信時に自動分割する。

それに合わせて、ヘッダーの内容を以下のように調整する。

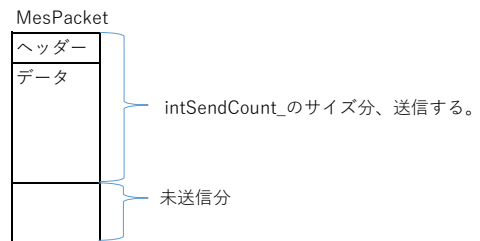
```
26 struct MesHeader
27 {
28     MesType type; // メッセージの種類
29     unsigned char next; // データ送信が分割の場合に、次がある:1、無い:0
30     unsigned short sendID; // 分割送信時のナンバリング
31     unsigned int length; // 分割かどうかに関わらず、単一パケットのデータ長(intの数)
32 }
```

■メソッドの構成図

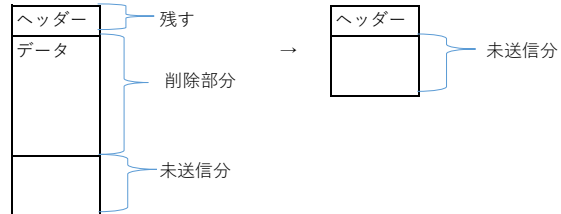


- bool NetWork::SendMes(MesType mesType, MesPacket data)
- ① 受け取ったMesTypeでヘッダーを生成して、MesPacketの先頭に挿入する。
 - ② 送信データ長を求める
intSendCount_、もしくはMesPacketのサイズで、小さいほうを使用する。
 - ③ 求めた送信データ長からヘッダーサイズを除いた分をヘッダーのlengthに入れる。
 - ④ MesPacketのサイズで送信する場合、ヘッダーのnext情報を0にする。
intSendCount_のサイズで送信する場合(分割送信)、ヘッダーのnextを1にする。
 - ⑤ 調整したヘッダー情報をMesPacketに入れる。

- ⑥ MesPacketからヘッダー込みで指定データ長分、データを送信する。



- ⑦ MesPacketからヘッダーを除いて、送信した部分を削除する。



- ⑧ 送信回数のsendIDをカウントアップする。
- ⑨ 未送信部分がまだ残っている場合、②へ戻り、未送信部分がなくなるまで繰り返す。