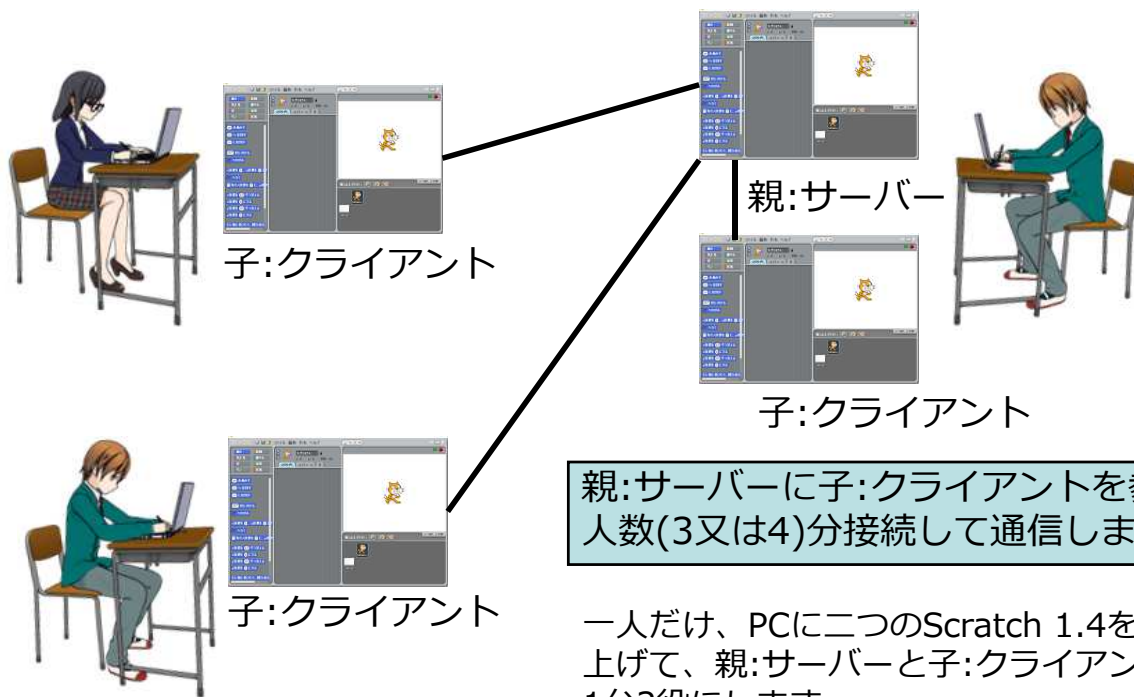


3人又は4人でグループを作って作業します。



親:サーバーに子:クライアントを参加人数(3又は4)分接続して通信します。

一人だけ、PCに二つのScratch 1.4を立ち上げて、親:サーバーと子:クライアントの1台2役にします。

親:サーバーに接続する

手順1: 親:サーバーのIPアドレスを調べる。



親:サーバー



Shiftを押しながら、「共有」をクリックして「Host Mesh」を選択。



IP Addressが表示されるのでメモしておく。



手順2: 親:サーバーのIPアドレスを指定して接続する。



子:クライアント



Shiftを押しながら、「共有」をクリックして「Join Mesh」を選択。



メモした値を指定する。

実験1: SNS(LINEみたいなもの)を動かす



子:クライアント
PC-SNS.sb



親:サーバー
Server-SNS.sb



子:クライアント
PC-SNS.sb

親:サーバーと子:クライアントで対応するプログラムを実行します。
子:クライアントで操作して、両方の画面を見ながらどのように動いているか考えてみましょう。

入力する文字はアルファベット(ローマ字)だけが正しく表示されます。

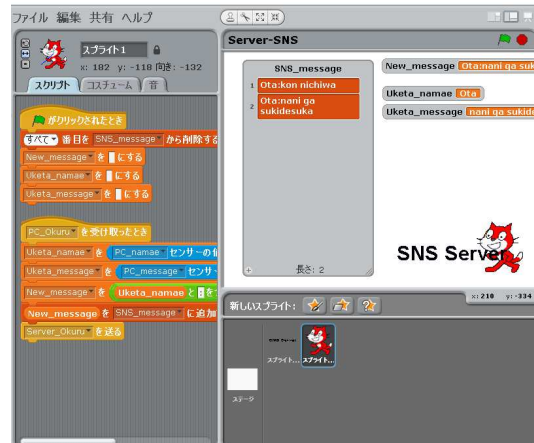
5

実験1: SNS(LINEみたいなもの)のプログラムを解析する

子:クライアント
PC-SNS.sb



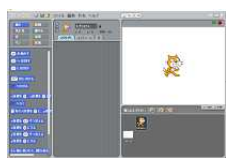
親:サーバー
Server-SNS.sb



親:サーバーと子:クライアントのプログラム(プロジェクト)の中身を見て、どのように動作するプログラムか考えてみましょう。

6

実験2: メール(送信だけ)を動かす



子:クライアント
PC-MailOkuru.sb



親:サーバー
Server-MailOkuru.sb



子:クライアント

PC-MailOkuru.sb

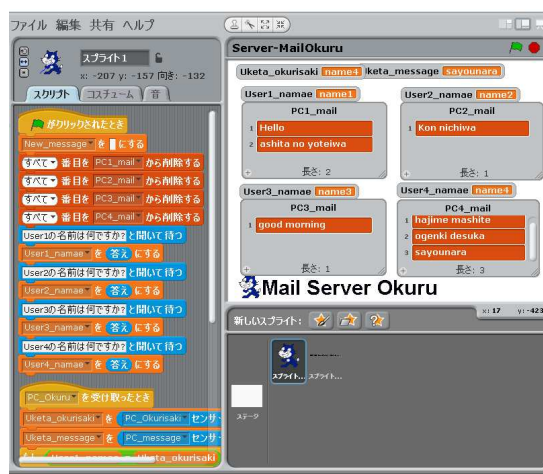
プログラムを新しく開いたら、再度親:サーバーに接続

親:サーバーと子:クライアントで対応するプログラムを実行します。
子:クライアントで操作して、両方の画面を見ながらどのように動いているか考えてみましょう。
特に、親:サーバーにどのようにメッセージが溜まっていくか考えてください。

実験2: MAIL(送信だけ)のプログラムを解析する

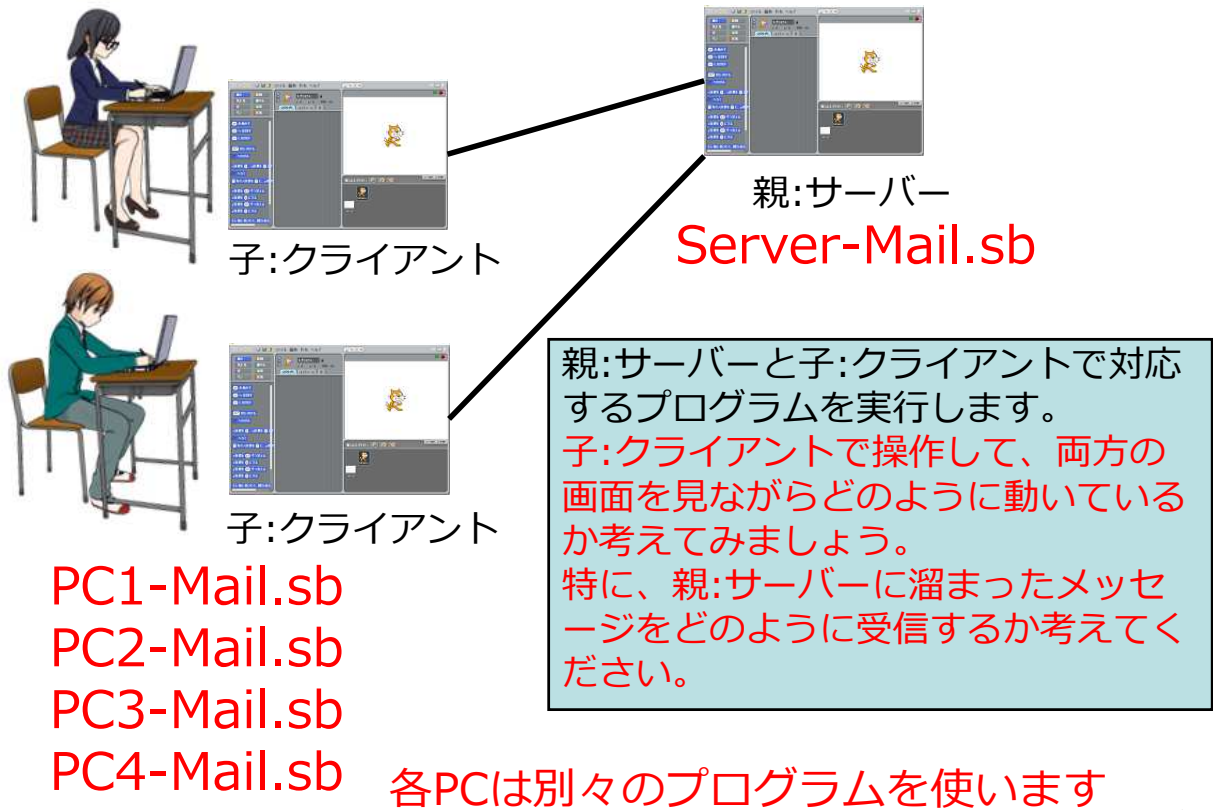
子:クライアント
PC-MailOkuru.sb

親:サーバー
Server-MailOkuru.sb



親:サーバーと子:クライアントのプログラム(プロジェクト)の中身を見て、どのように動作するプログラムか考えてみましょう。

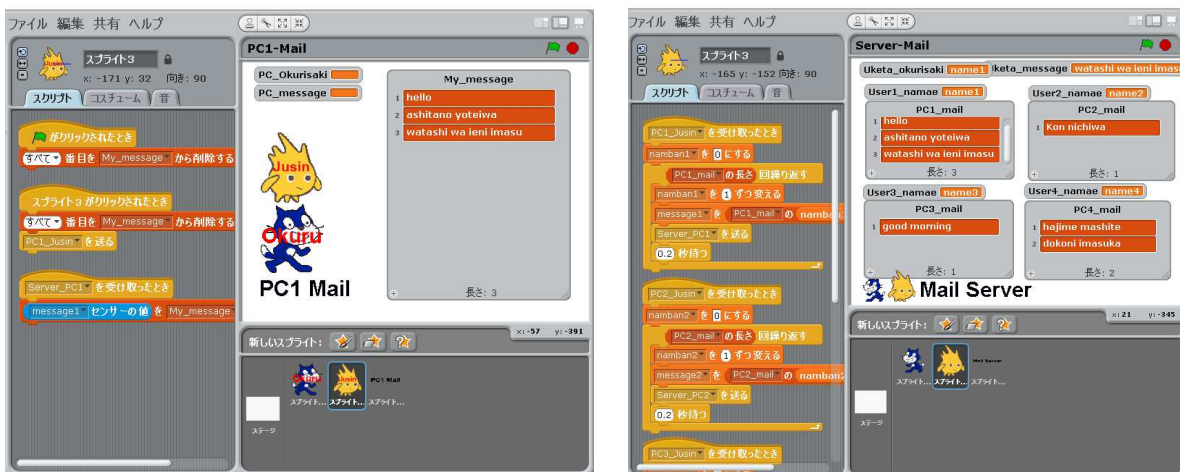
実験3: メール(送信・受信)を動かす



実験3: MAIL(送信・受信)のプログラムを解析する

子:クライアント
PCX-Mail.sb

親:サーバー
Server-Mail.sb



親:サーバーと子:クライアントのプログラム(プロジェクト)の中身を見て、どのように動作するプログラムか考えてみましょう。

キットに送信用、ゴボに受信用のプログラム(スクリプト)が設定されています。

次のことを考えてみよう

- サーバーの役割は?
- SNSとメールの違いは?
- メールで送信と受信の役割の違いは?

