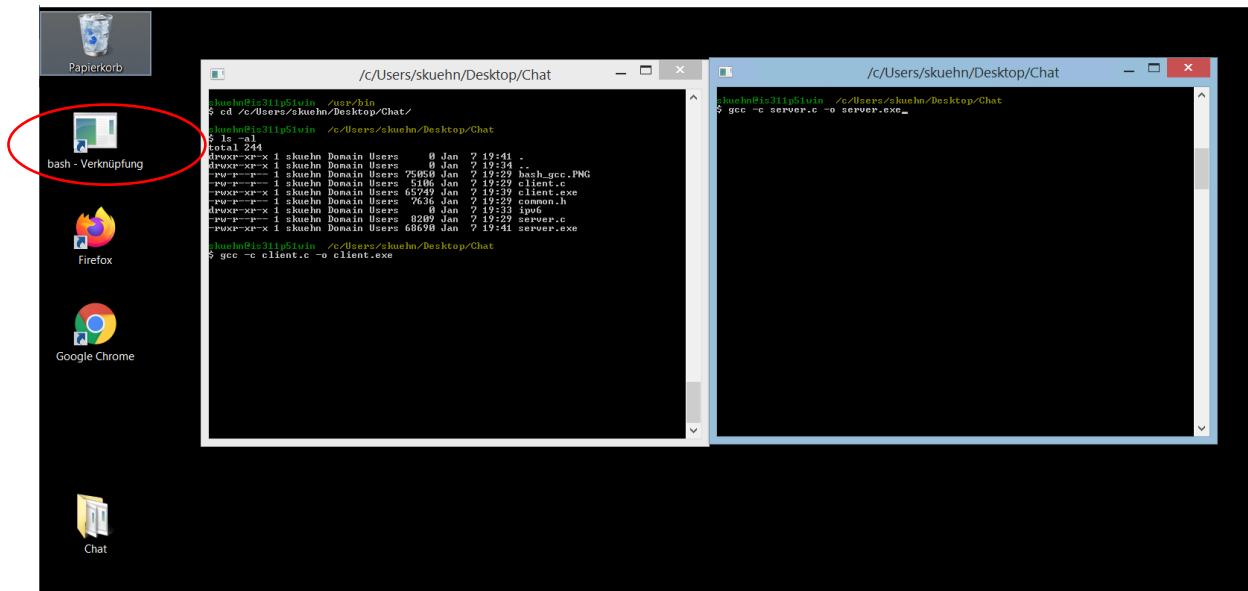


## Testen der Client – Server Anwendung im Labor S 311 am Beispiel einer Chat-Anwendung für Windows/ Linux

- I. Übersetzen und Aufrufen in Linux-Umgebung unter Windows:  
Starten der bash.exe (MSYS2 Umgebung d.h. **Linux**-Umgebung auf Windows) auf dem Desktop des Rechners (für den lokalen Test auf Client und Server können Sie auch gleich 2 x die bash aufrufen wie im folgenden Screenshot gezeigt):



Die Quellen kann man entweder auf dem Laufwerk c in seinem Verzeichnis (sNummer) ablegen oder unter dem Laufwerk u. Letzteres hätte den Vorteil, dass die Quellen etc. rechnerübergreifend zur Verfügung stehen.

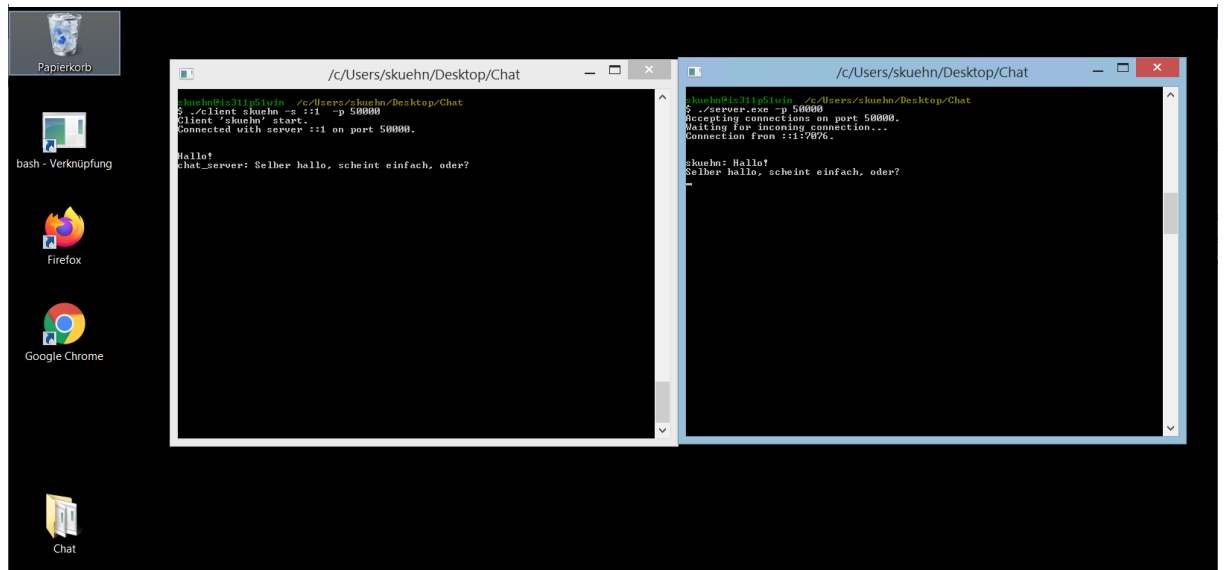
Aus der bash heraus hat man Zugriff auf die Dateien, indem man sich mittels cd in das Verzeichnis /c/Users/sNummer... bzw. /u/... bewegt. Hier:  
cd /c/Users/skuehn/Desktop/Chat.

Dort kann nun mittels des Gnu C – Compiler (gcc) die Übersetzung erfolgen. Fertig.

- II. Übersetzen und testen der Anwendung unter Ubuntu:  
Funktioniert zumindest host lokal (::1) wie oben gezeigt.
- III. Übersetzen unter Windows z.B. unter Verwendung der Visual Studio Entwicklungsumgebung. Bitte beachten Sie unbedingt, dass Sie die Bibliothek ws2\_32.lib mit an das Anwendungsprogramm linken. Entweder in der Konfiguration angeben oder ganz einfach oben im Quelltext mittels:  
#pragma comment(lib,"Ws2\_32.lib") //Winsock Library

**Beim Aufruf der Programme immer den PORT 50000 angeben. Nur dieser ist in der Firewall für Windows und Linux freigeschaltet.**

(Das heißt, der Server bindet sich an/hört auf den Port 50000 !!!) Hier gezeigt am lokalen Aufruf von Client und Server.



#### IV. Testen von Client und Server auf unterschiedlichen Rechnern...

Dazu notwendigerweise eine 2. Shell auf einem weiteren Rechner öffnen. Sollte der Server dort gestartet werden, ist vorab noch dessen link local IPv6 Adresse (fe80::....) über `ipconfig -all` (windows und bash-Shell) bzw. `ip a` (Ubuntu) abzufragen. Diese dann auf Clientrechner bei Starten der Anwendung als Programmargument angeben, zusammen mit dem Port. Die Serveradresse lässt sich wiederum günstigerweise über Laufwerk u teilen.

Achtung: die link local IPv6 Adresse beinhaltet unter Windows am Ende einen scope id (%5 in der Regel für alle Interfaces im Labor). Falls Sie mit der Adressfunktion `getaddrinfo` arbeiten, können Sie die gesamte Adresse inklusive scope id an die Funktion übergeben und alles funktioniert. Falls Sie andere Funktionen verwenden, müssen Sie scope id und Adresse getrennt behandeln, siehe unteres Beispiel.

Tipp: lesen Sie sich die Hinweise im Chat unten durch!!!

