# So you want to chat on Xinu By B. Haag

The purpose of this assignment is to illustrate how to use hardware UARTs via the TTY drivers, as such, our “chat” program will utilize these to communicate back and forth. Conceptually, the flow of data is as follows. Remember you connect to Brylow(or whichever) with mips-console, and to Brylow1 with xinu-console.

To start with, make sure to include standard libraries for dealing with text programs on Xinu.

kernel.h

stdlib.h

shell.h

stdio.h

You’ll want some kind of structure to hold messages coming it from the console. Typically this will be a character array. For ease you can define a constant for its length.

#define message\_length 16

char messages[message\_length]

Let’s say TTY1 has just sent us a message(us being TTY0), we can read that simply with the read command. You do this by reading from your own local TTY.

read(TTY0, messages, message\_length)

This checks TTY0, for the char array messages, up to the length we defined earlier.

Similarly you can write to TTYs quite simply.

write(TTY1, messages, message\_length)

Or

fprintf(TTY1, “text”)

So TTY0 reads from itself, gets any new messages, user types a message, sends it to TTY1, TTY1 reads from itself, checks for new messages. This cycle repeats until the program is exited.

One consideration is clearing the buffer of each TTY, when you send text to the TTY driver, it’s not deleted. So your second message may appear concatenated to the first one if you leave it there! You’ll need a method to reset it to empty after each message.

That’s really all there is to chat on Xinu.