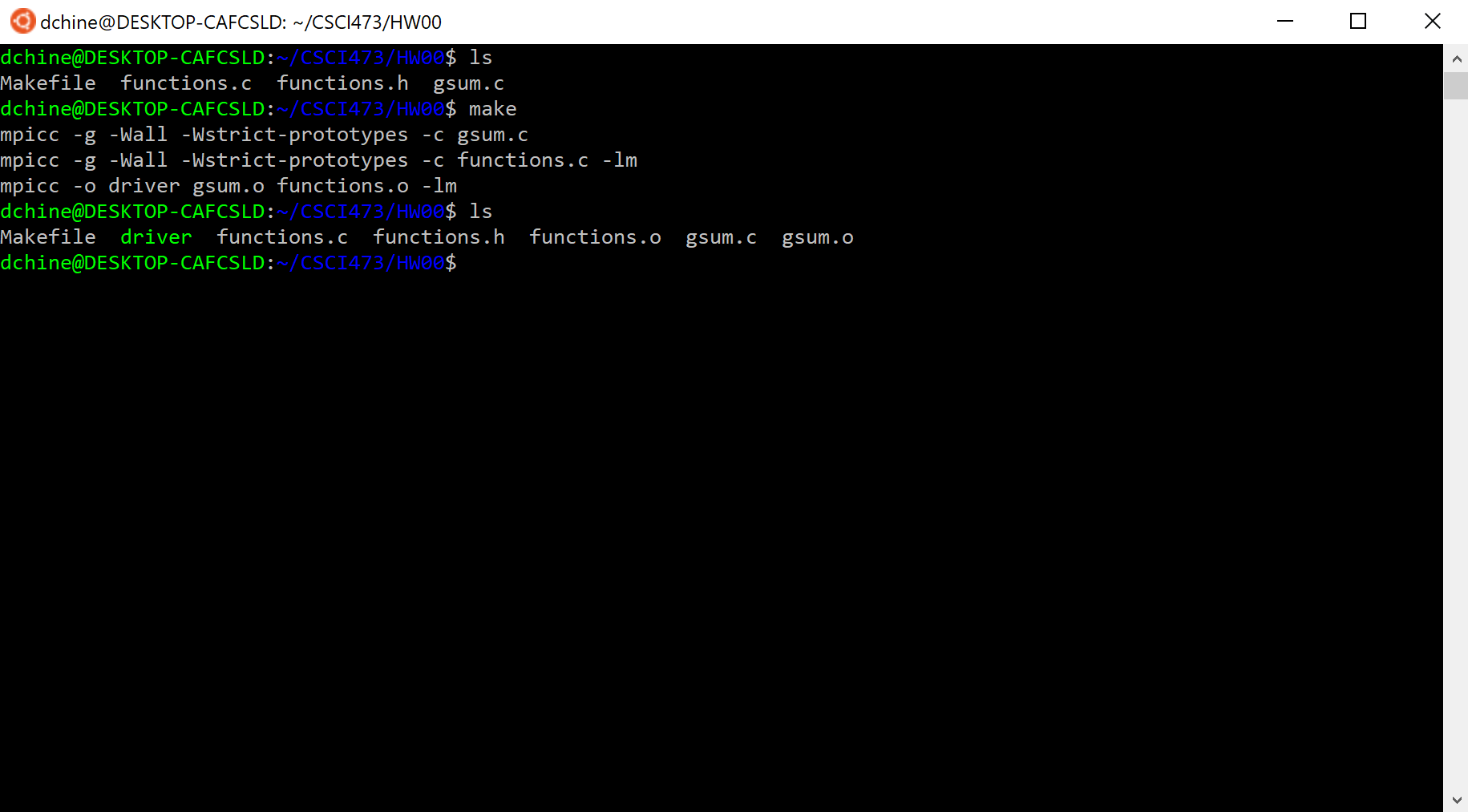
Dan Hine

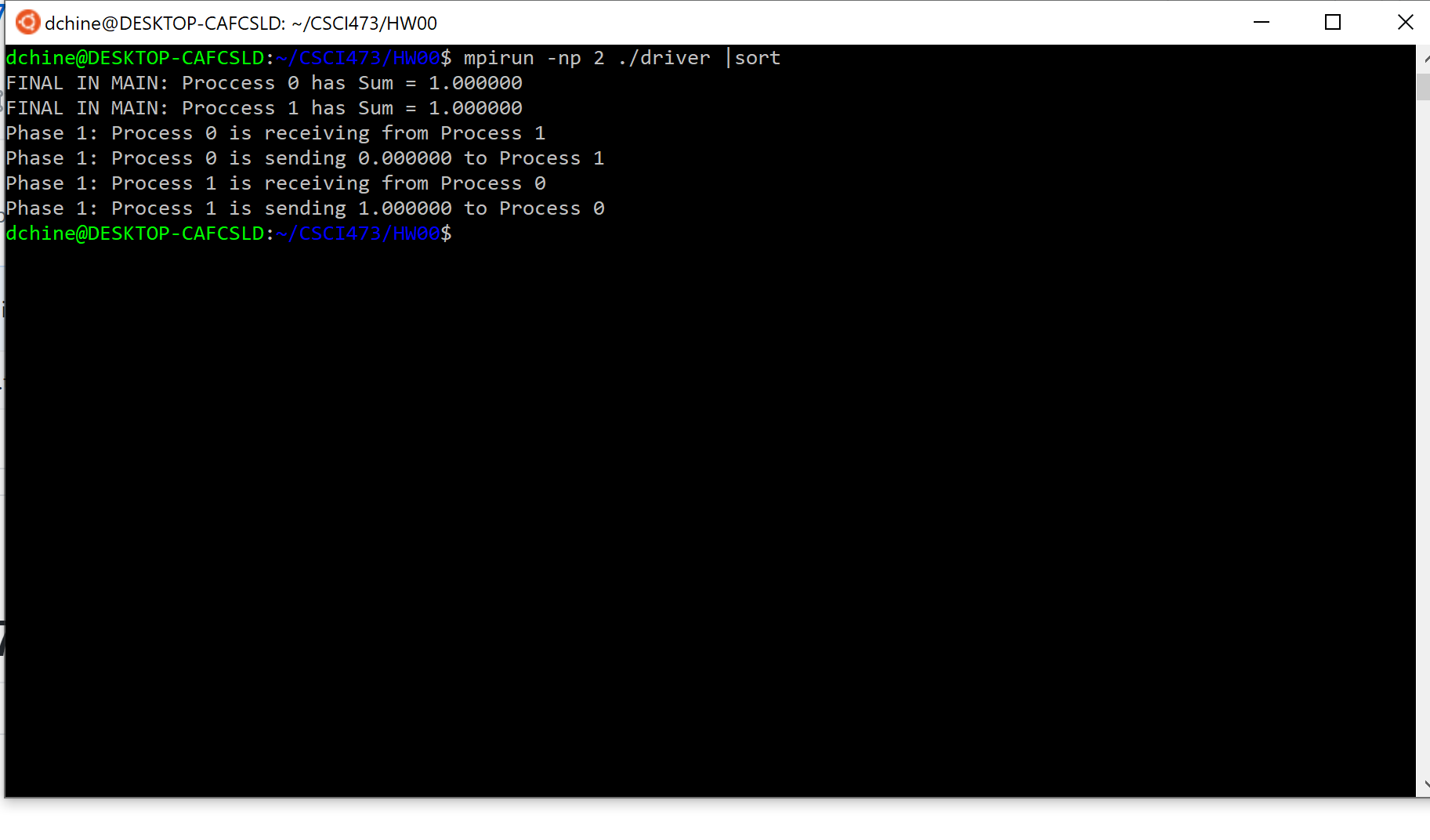
CSCI 473

HW00

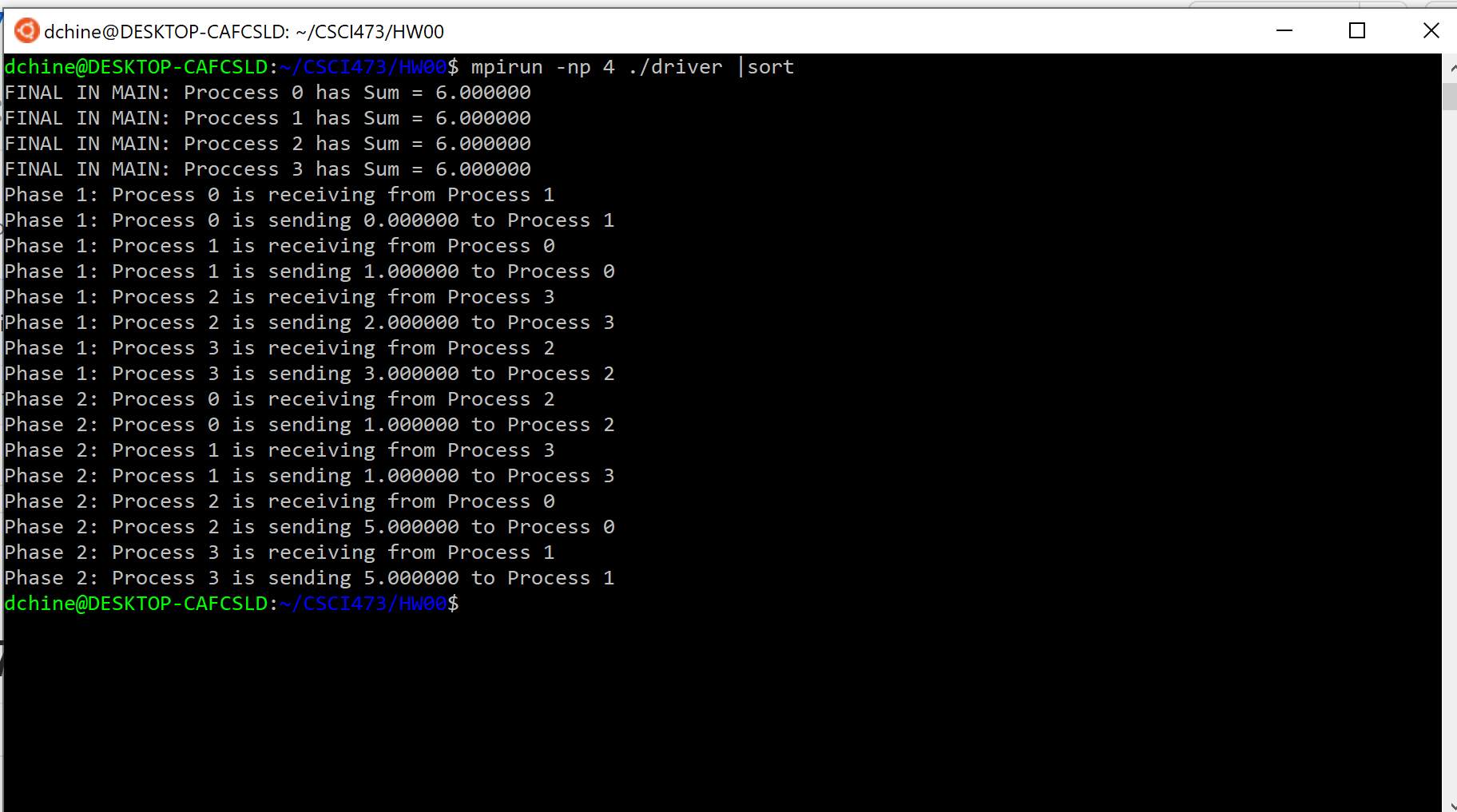
Here is the contents of the HW00 file which are compiled by the makefile. After which the .o files and driver program can be seen. The gsum program contains the main method and calls the global\_sum function that is prototyped in functions.h, and defined in functions.c. In my makefile I had to move $(LDFLAGS) to the end of the line defining the compilation of driver, in order to use methods from the math.h library in my global\_sum method. This included log2 to find the number of steps and pow to create my mask. The number of steps was equal to log2(np), (number of processes), and my mask was equal to 2^(step - 1) + 0.5. or (int)(pow(2, (step-1)))+0.5.



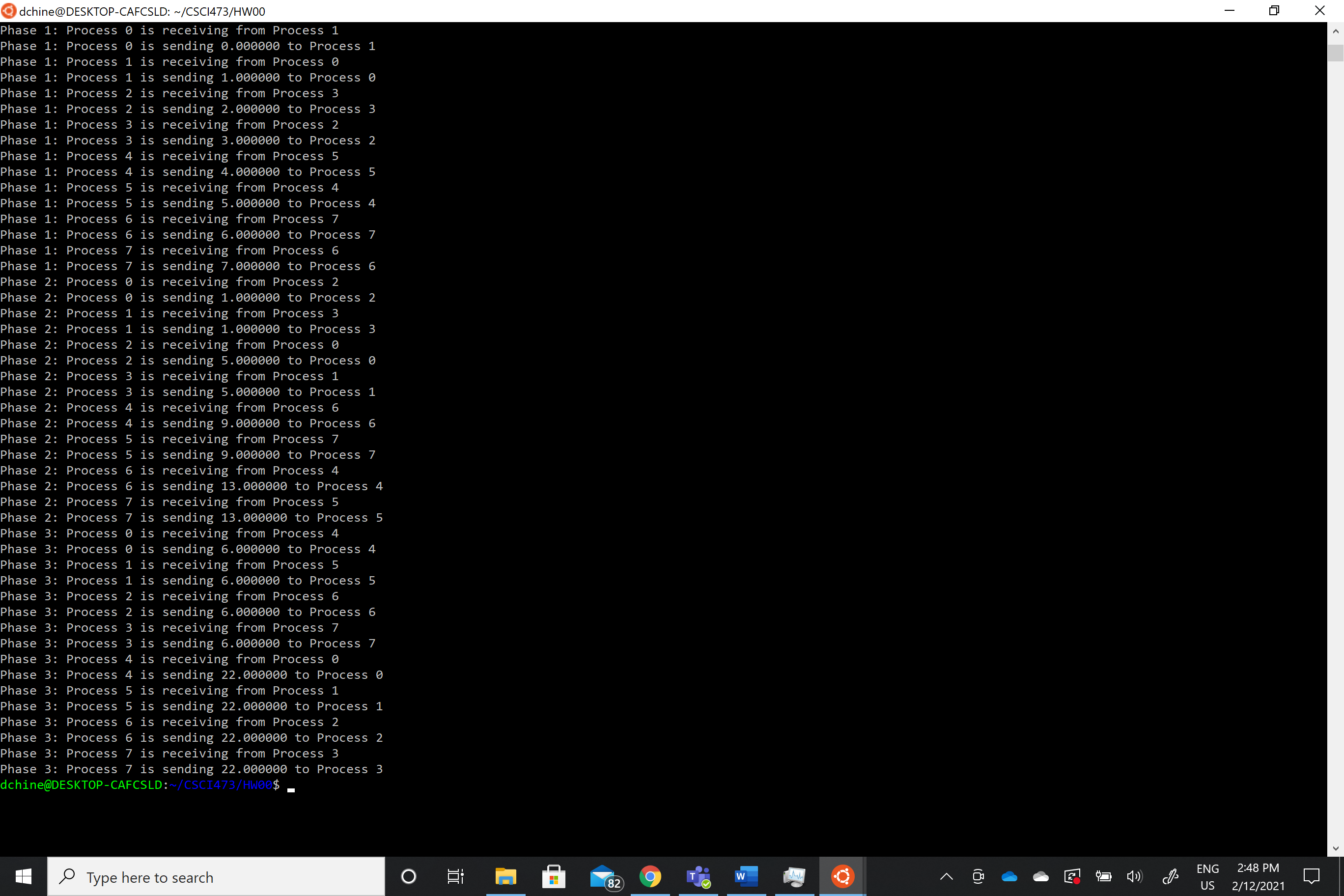
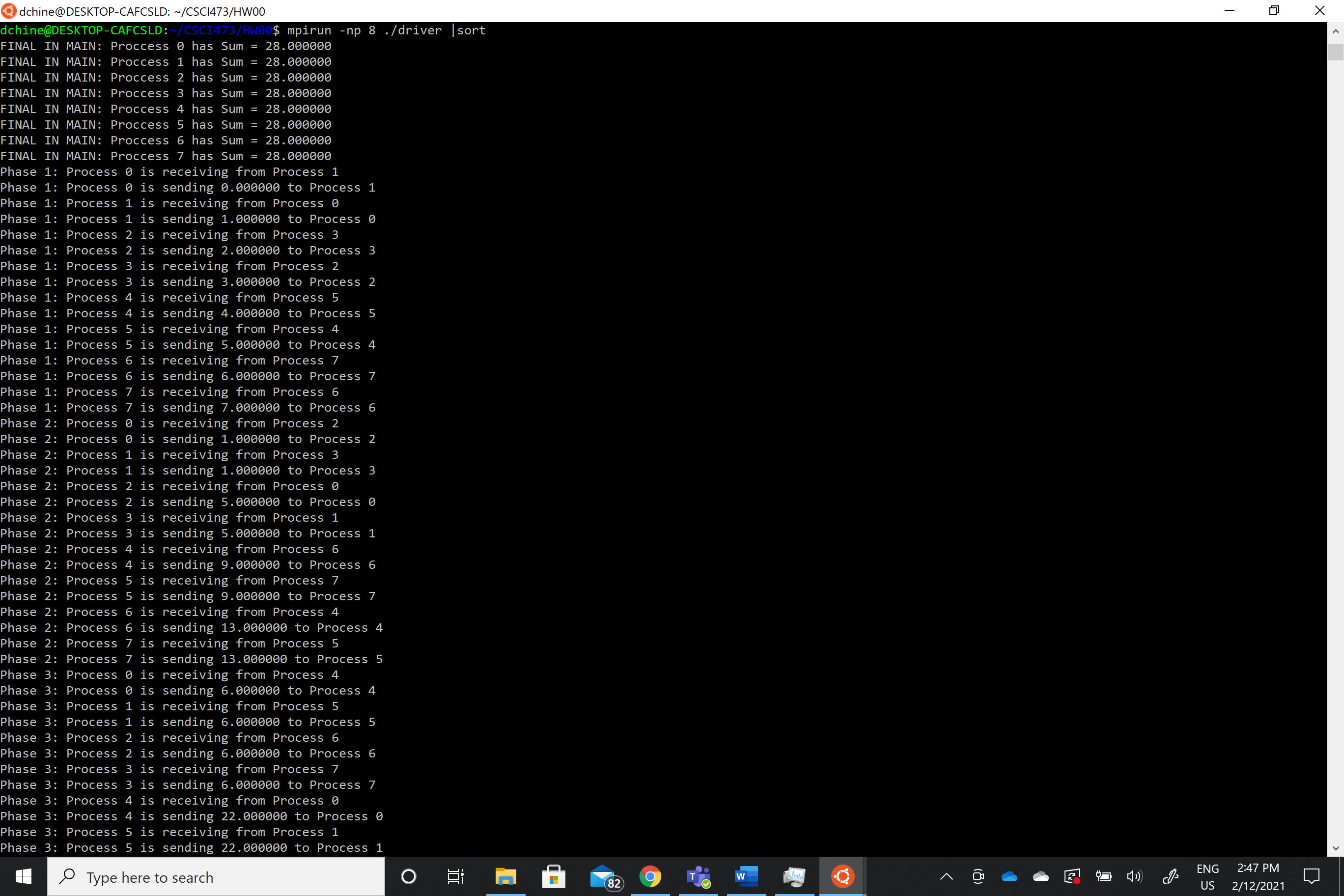
Running the driver program with np = 2. Global sum equals 1 as the processes are set to have an initial value equal to their rank.



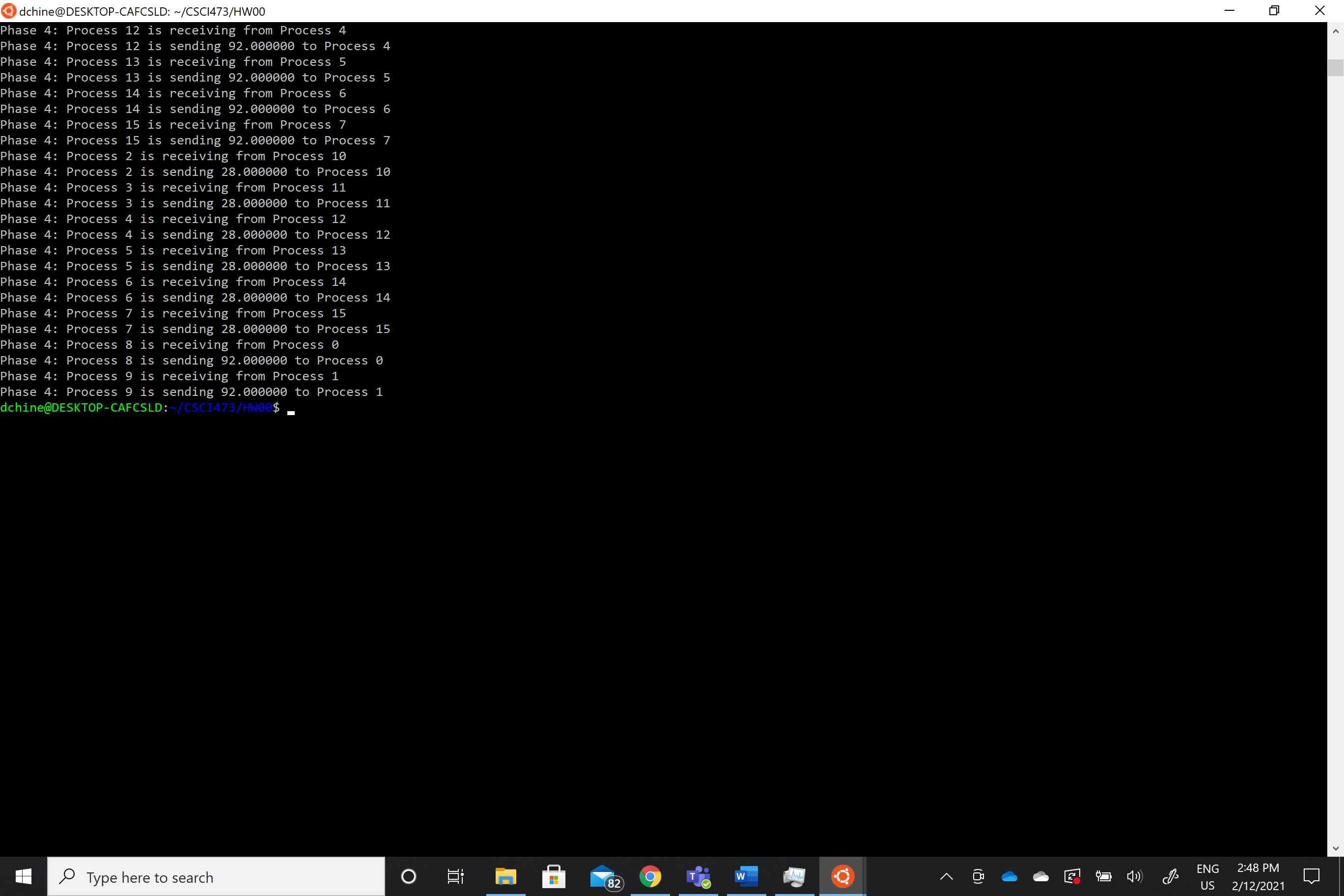
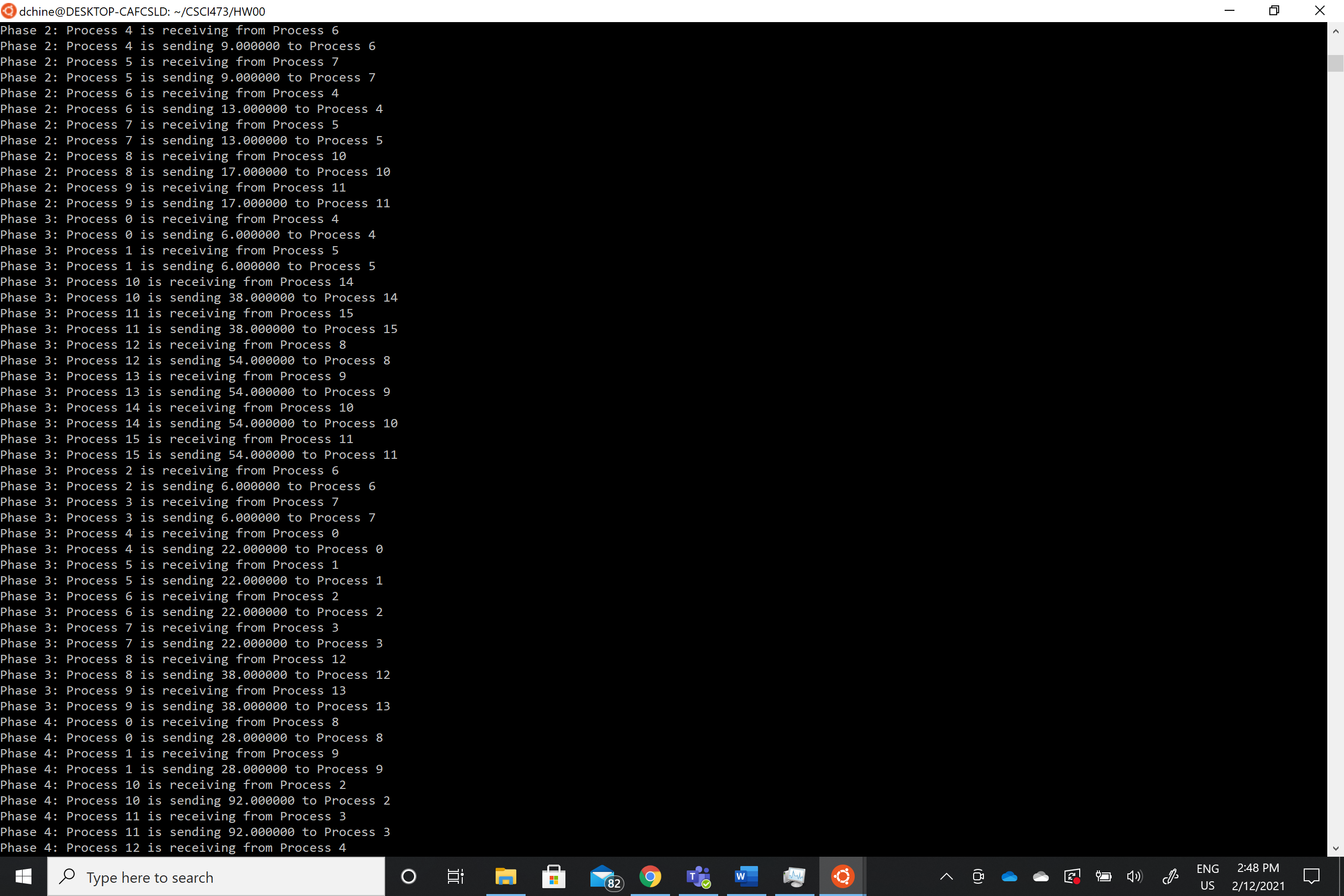
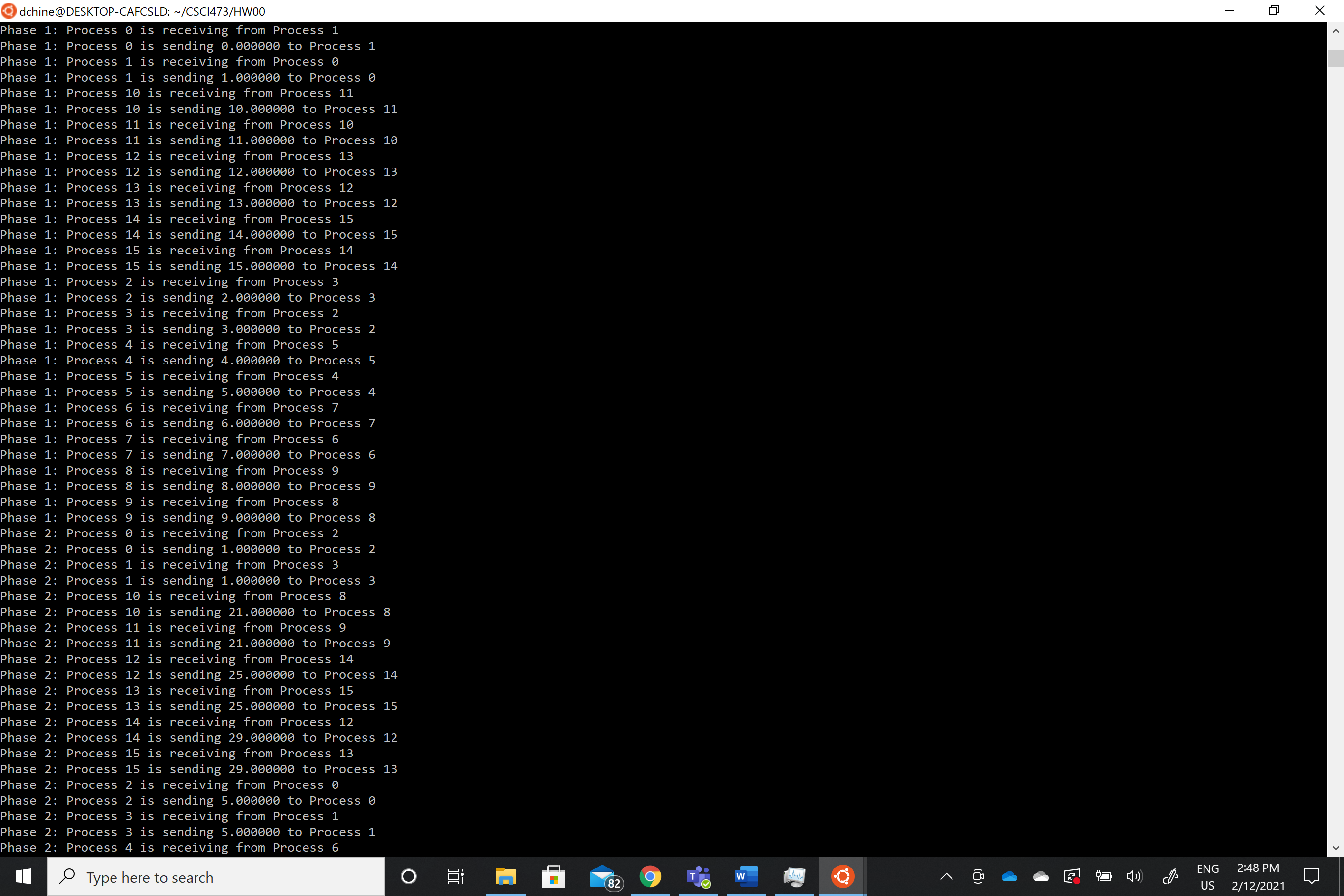
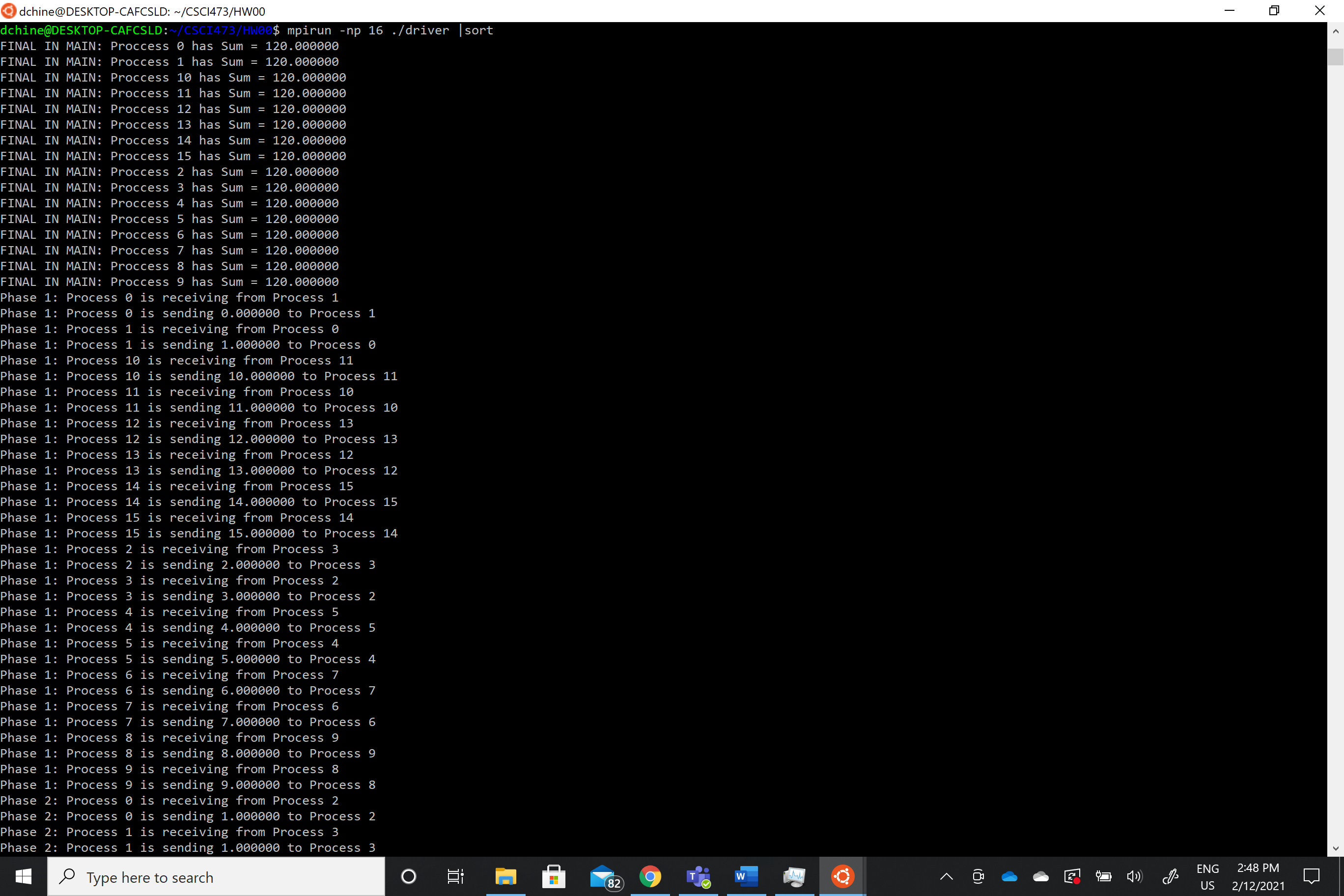
Running the program with np = 4 yields a global sum of 6. The steps of sending and receiving within each phase can also be seen. The printed statements show which processes is sending what value to which processes, as well as the processes that are being received from.



Running the program with np = 8 yields a global sum of 28.



And finally running the program with np = 16 yields a global sum of 120.

The program is only able to run when np = power of 2, and if np does not equal a power of two the program informs the user that it must be, and the program exits.

