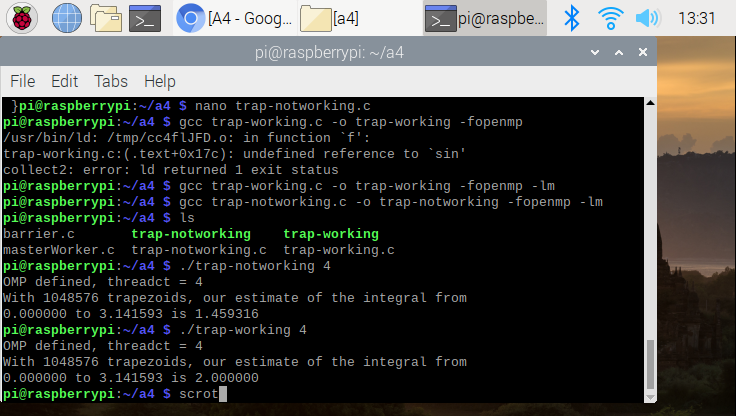
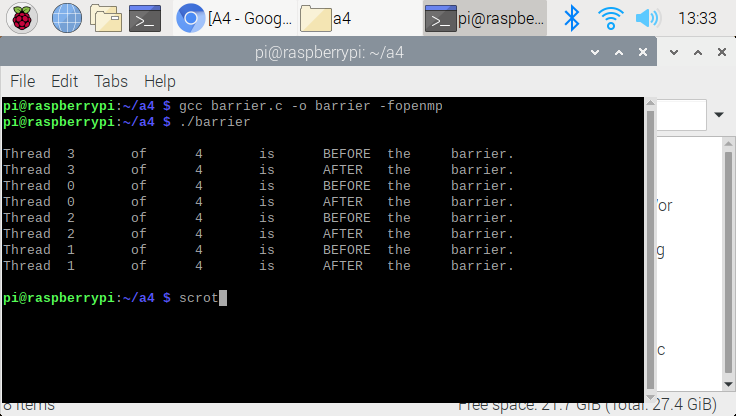
TrapWork & TrapNotWork ScreenShot



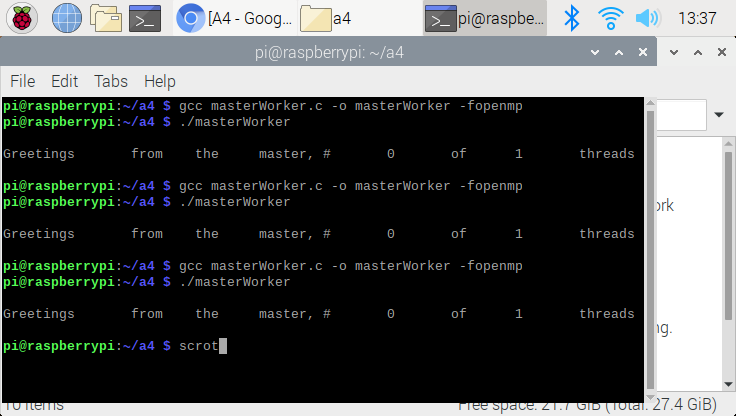
The photos are included both “TrapWork” and “TrapNotWork” result. This program use a 4 threads, since raspberry pi contains 4 core processor. Both programs are designed to compute the integral of sin of x from 0 to pi. Both programs are nearly equal program except line 37 and 38. The difference for those program is “private(i) shared (a, n, h) reduction(+: integral) “(in trapWork), this statement would allow the processor to perform the reduction parallel algorithm. This is the main reason why the program for Work and NotWork shows the different result. Overall, the correct answer for integral equation is shown in “Trap-Work” result(2.000). The answer for trap-notWork is incorrect.

Barrier Screenshot



Barrier program is to ensure all designated parallel program is completed before moving to another section. From the photo above, the threads (4 threads from 0 to 3) are being checked by the barrier. All threads are moved from “BEFORE” to “AFTER” then completed the program, which means the each threads are completed (after) and ready to moved to another section.

masterwork ScreenShot with pragma and without pragma.



The very first output is with the pragma. Next two result is without pragma. Unfortunately, we defined on difference between with and without pragma. The main purpose of this code is to define the common pattern and check whether the which threads are working on the program. When all workers are done with the processor, the master thread (in this case thread 1) will compile all the code and finished the program by giving us a result.