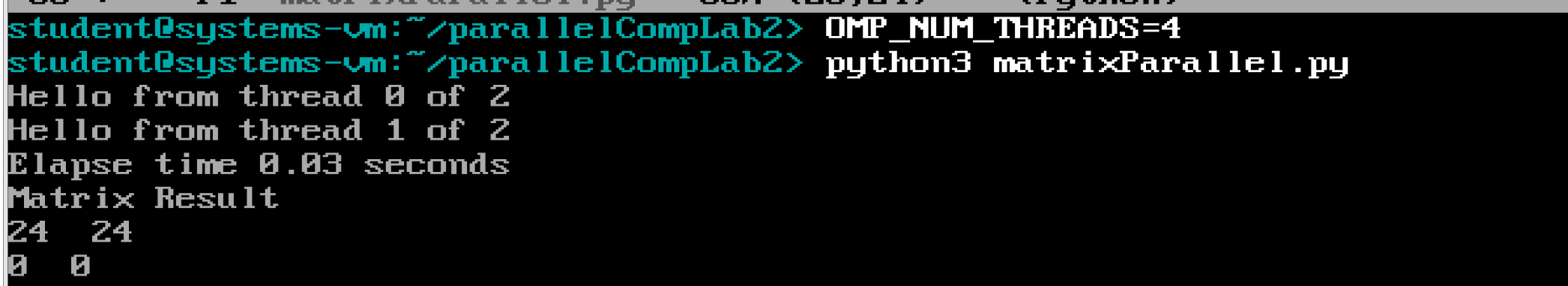
The goal of this lab is to create a parallel matrix multiplication program using Pymp. One of many problems I encountered in this lab was getting used to the Vmware, connecting through network importing pymp, updating all of python modules that I was using until I can finally start coding the parallel program.

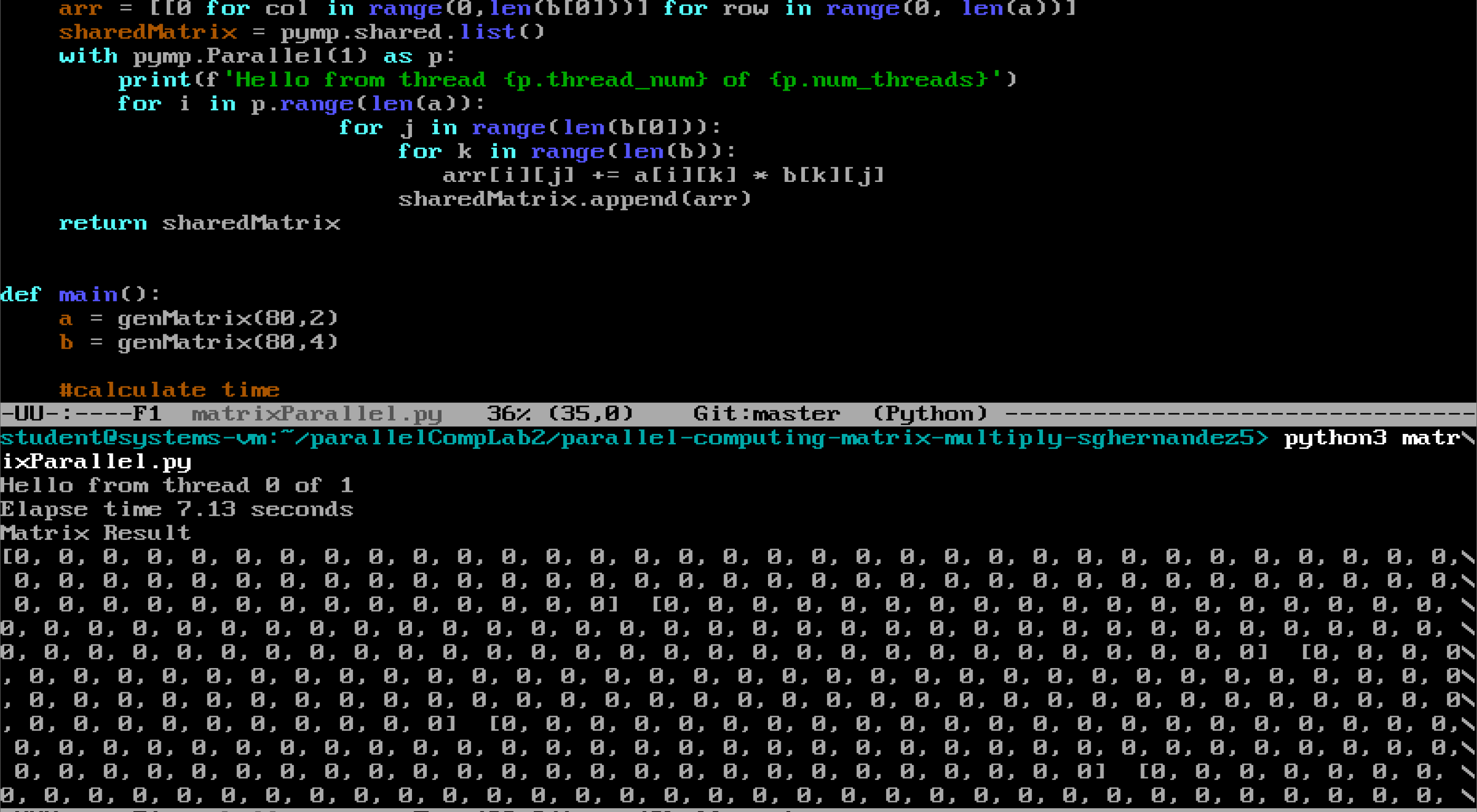
As for the parallel program one of the problems I had was understanding how my shared matrix was working. I know my implementation is wrong because when I print 1-8 threads it loses reference to the past matrix and prints out all zeros. Another issue I had was hardcoding the number of threads with the command OMP\_SET\_NUM\_THREADS = …python3 …. For some reason, it was not recognizing it but I was able to fix it and hardcode within the code itself after asking Professor Pruitt. Another problem I had, was although the number of threads were increasing the elapsed time was too, instead of making it faster I believe this is by my implementation and not using the sharedMatrix correctly. However I this problem was somehow fixed when I added the sharedMatrix.append(arr) line, however the subarray loses reference to the shared.



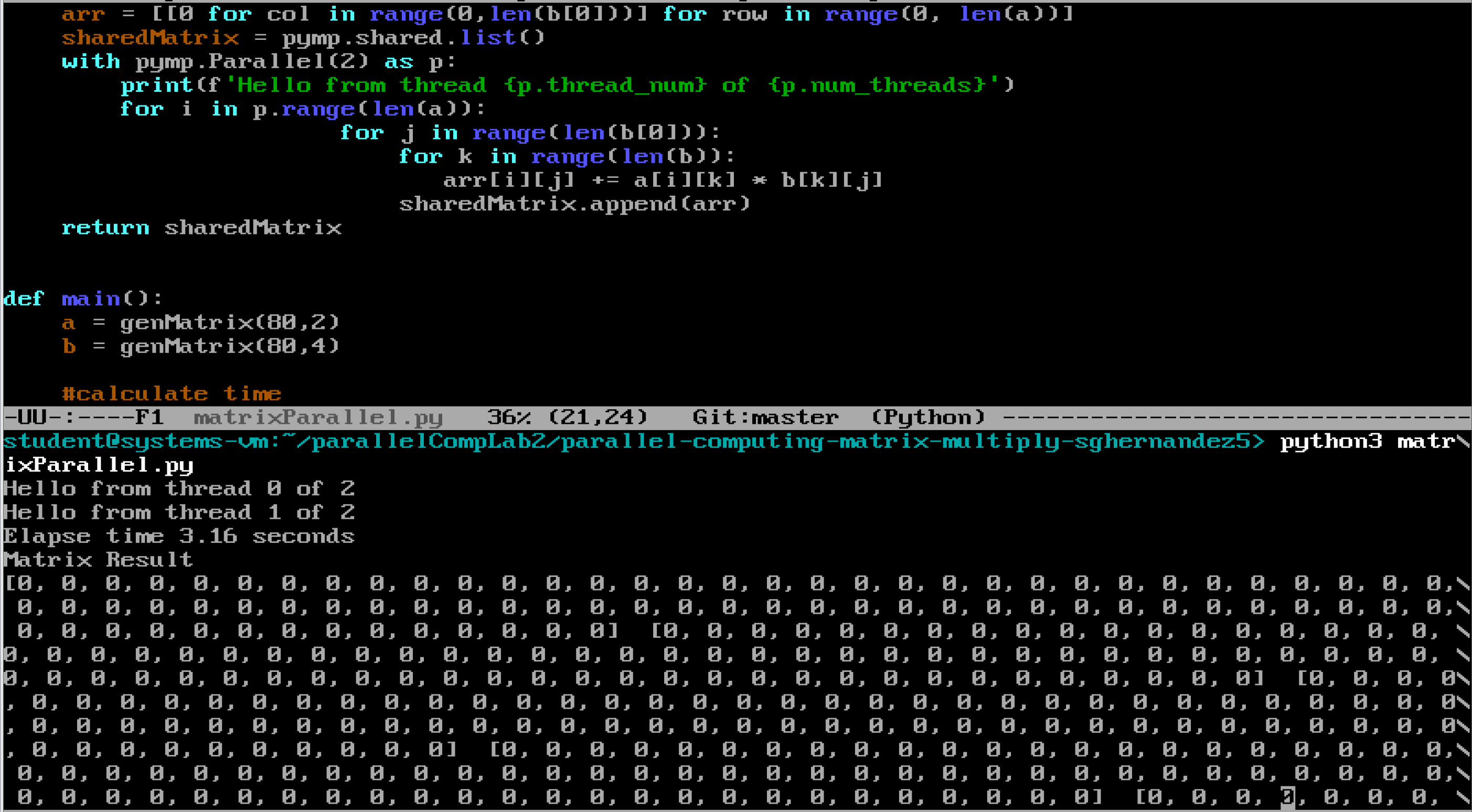
**Figure 1** shows me testing my small matrix, which outputs zeros, one of the errors I encountered.

**Below is the elapse time for 1 Thread:**

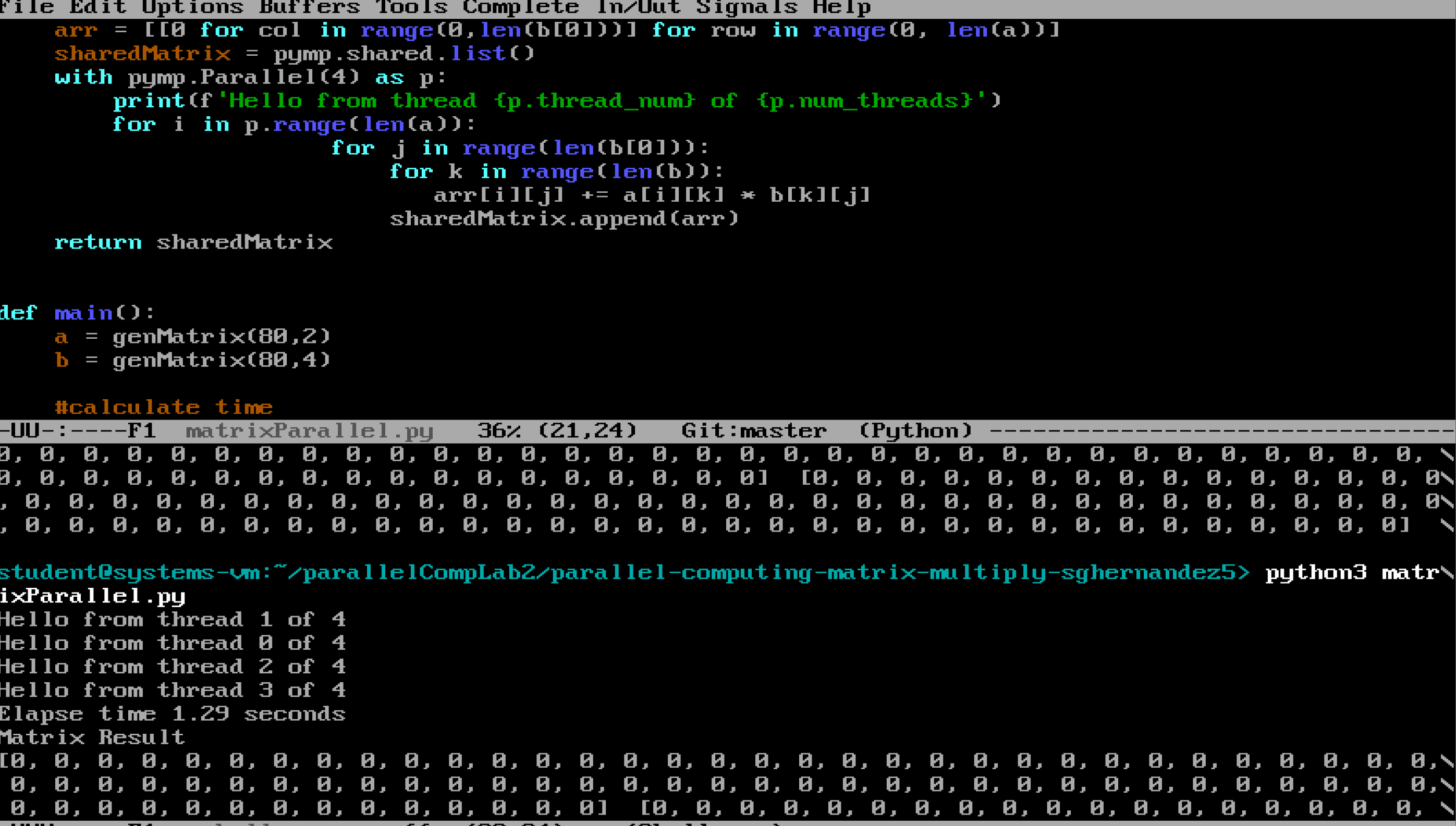
**The output subarray matrix is losing reference to the shared matrix.**



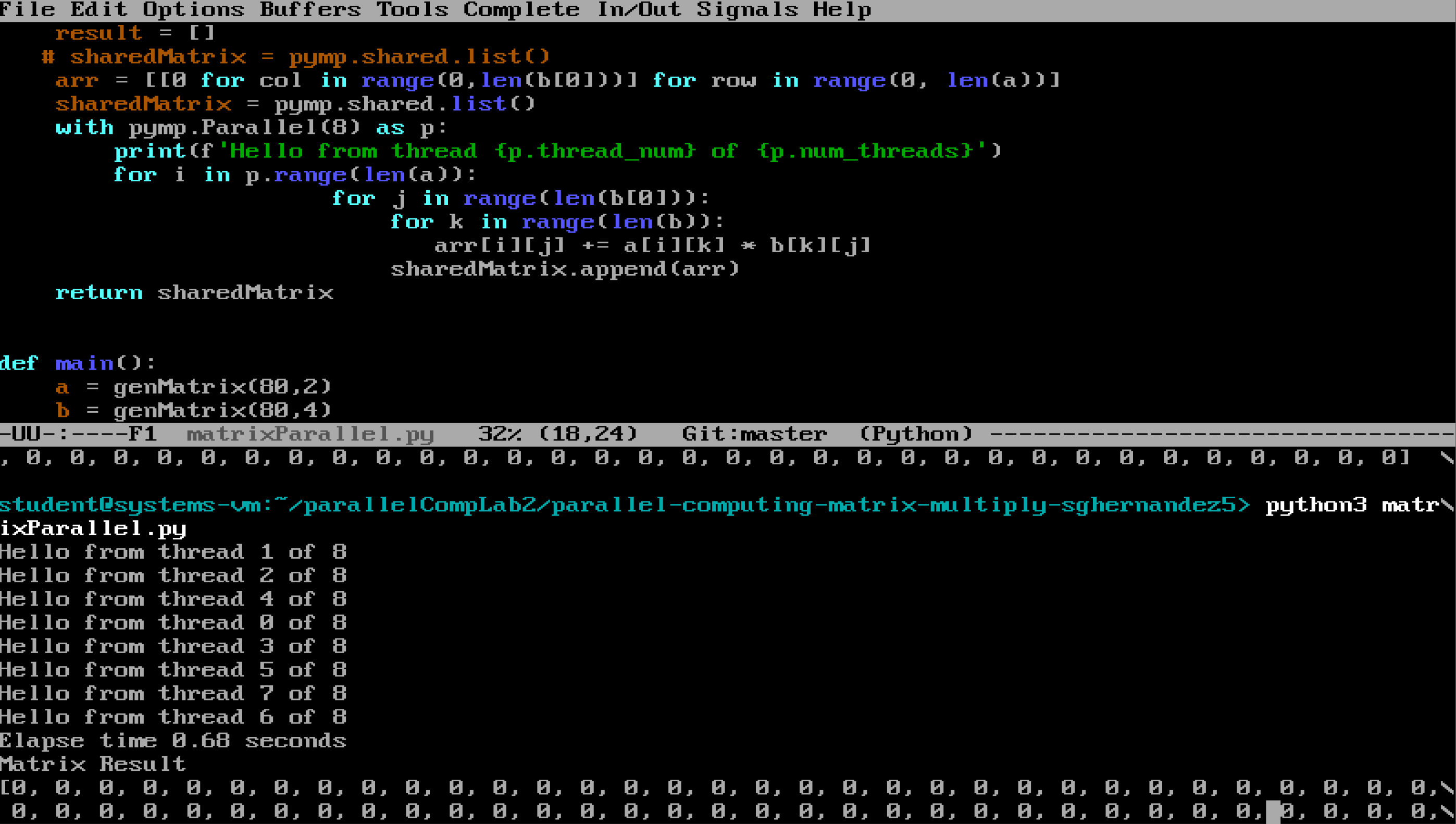
**Below is the elapse time for 2 threads:**



**Below is the elapse time for 4 threads:**



**Below is the elapse time for 8 threads:**



As a result, my parallel program does indeed get faster as the use of threads increase however my implementation of my sharedMatrix got rambled and I was left with a subarray with 0’s. All together this lab took me about 12 hours due to errors and trying to get myself accustomed with the vmware and looking up errors I encountered with the Pymp library.

Below is my CPU information :

