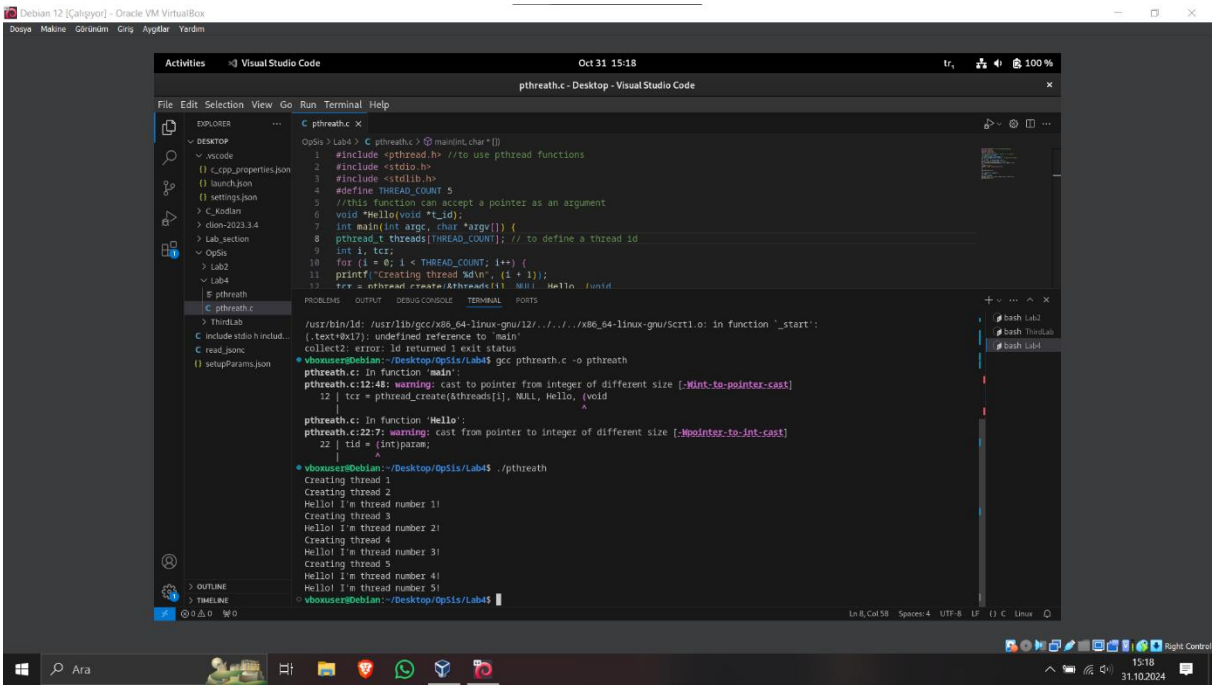
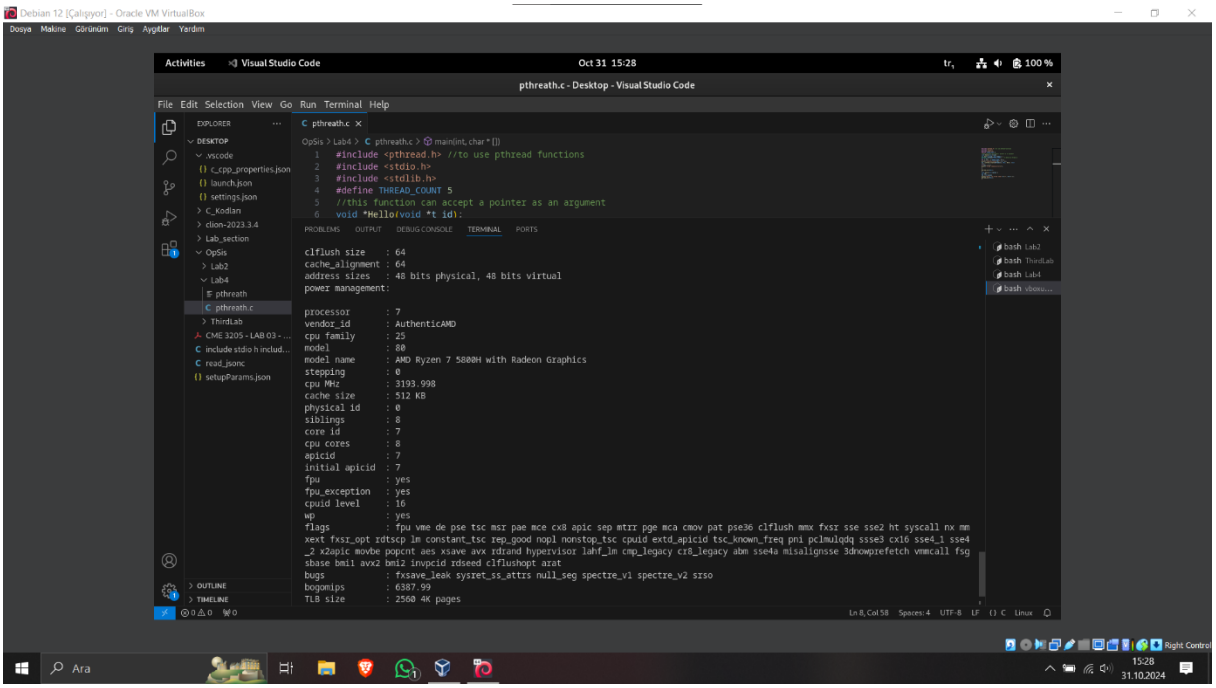
Çağrı AYDIN 2021510010 – Lab 3

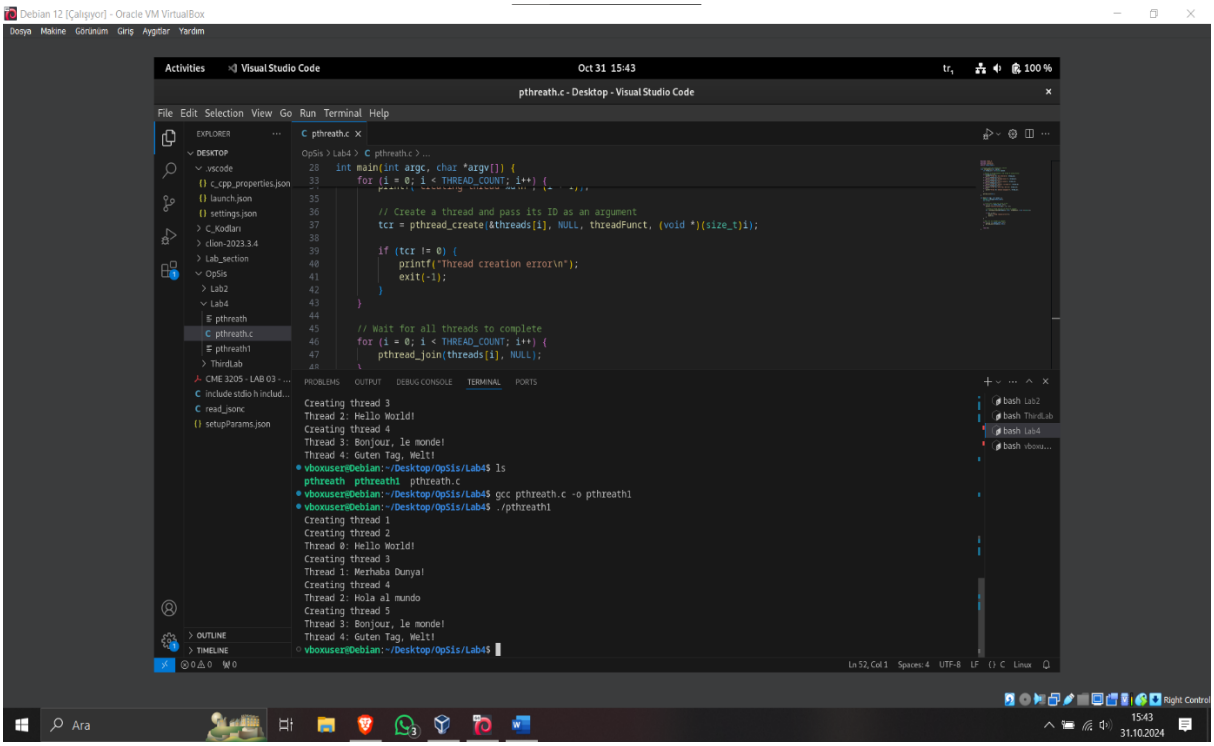
Code 1:



My Processor:

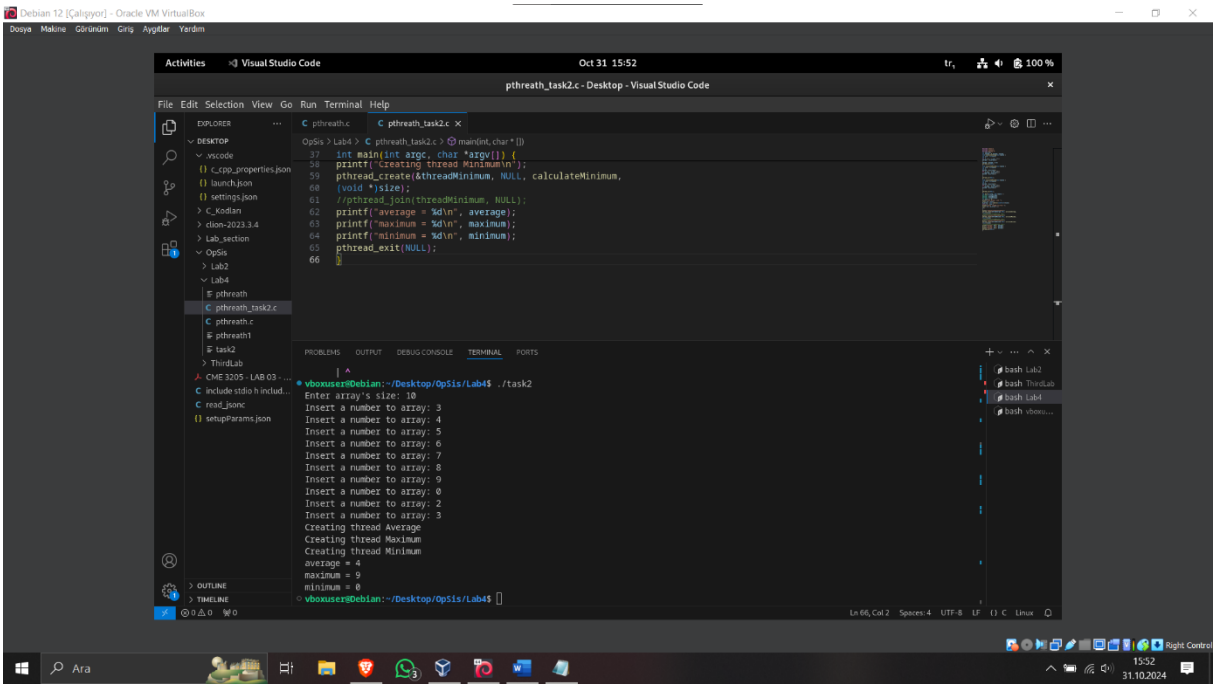


Task 1:


Debian 12 [Çalışıyor] - Oracle VM VirtualBox
Dosya Makine Görünüm Giriş Aygıtlar Yardım
Activities
Visual Studio Code
Oct 31 15:43
pthreath.c - Desktop - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
DESKTOP
.vscode
c_cpp_properties.json
launch.json
settings.json
C_Kodları
clion-2023.3.4
Lab_section
OpSis
Lab2
Lab4
pthreath
pthreath.c
pthreath1
ThirdLab
CME 3205 - LAB 03 - ...
include stdio.h includ...
read_jsonc
setupParams.json
OUTLINE
TIMELINE
OpSis > Lab4 > pthreath.c > ...
int main(int argc, char *argv[]) {
for (i = 0; i < THREAD_COUNT; i++) {
// Create a thread and pass its ID as an argument
tcr = pthread_create(&threads[i], NULL, threadFunct, (void *)(size_t)i);
if (tcr != 0) {
printf("Thread creation error\n");
exit(-1);
}
}
// Wait for all threads to complete
for (i = 0; i < THREAD_COUNT; i++) {
pthread_join(threads[i], NULL);
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Creating thread 3
Thread 2: Hello World!
Creating thread 4
Thread 3: Bonjour, le monde!
Thread 4: Guten Tag, Welt!
vboxuser@Debian:~/Desktop/OpSis/Lab4$ ls
pthreath pthreath1 pthreath.c
vboxuser@Debian:~/Desktop/OpSis/Lab4$ gcc pthreath.c -o pthreath1
vboxuser@Debian:~/Desktop/OpSis/Lab4$ ./pthreath1
Creating thread 1
Creating thread 2
Thread 0: Hello World!
Creating thread 3
Thread 1: Merhaba Dunya!
Creating thread 4
Thread 2: Hola al mundo
Creating thread 5
Thread 3: Bonjour, le monde!
Thread 4: Guten Tag, Welt!
vboxuser@Debian:~/Desktop/OpSis/Lab4$
bash Lab2
bash ThirdLab
bash Lab4
Ln 52, Col 1 Spaces: 4 UTF-8 LF C Linux
Ara
15:43
31.10.2024

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#define THREAD_COUNT 5

void *threadFunct(void *param) {
   int thread_id = (int)(size_t)param;

   if (thread_id == 0) {
      printf("Thread %d: Hello World!\n", thread_id);
   } else if (thread_id == 1) {
      printf("Thread %d: Merhaba Dunya!\n", thread_id);
   } else if (thread_id == 2) {
      printf("Thread %d: Hola al mundo\n", thread_id);
   } else if (thread_id == 3) {
      printf("Thread %d: Bonjour, le monde!\n", thread_id);
   } else if (thread_id == 4) {
      printf("Thread %d: Guten Tag, Welt!\n", thread_id);
   } else {
      printf("Thread %d: Unknown language!\n", thread_id);
   }
   pthread_exit(NULL);
}
int main(int argc, char *argv[]) {
   pthread_t threads[THREAD_COUNT];
   int i, tcr;
   // Create each thread and assign a task
   for (i = 0; i < THREAD_COUNT; i++) {
      printf("Creating thread %d\n", (i + 1));

      tcr = pthread_create(&threads[i], NULL, threadFunct, (void *)(size_t)i);

      if (tcr != 0) {
         printf("Thread creation error\n");
         exit(-1);
      }
   }
   for (i = 0; i < THREAD_COUNT; i++) {
      pthread_join(threads[i], NULL);
   }
   return 0;
}
```

Task 2:



Explanation: Yes, this code works parallel for calculating min, max and avg. number after getting the array size and array elements for time efficiency.

Task 3:

In this part our threats trying to access same global variables at the same time, so it occurs a data race. Threads access at the same time and trying to write different values so basically, we have a synchronization issue. To fix this we can make a flag if a threads is already using that data or we can make a pipeline between threads.