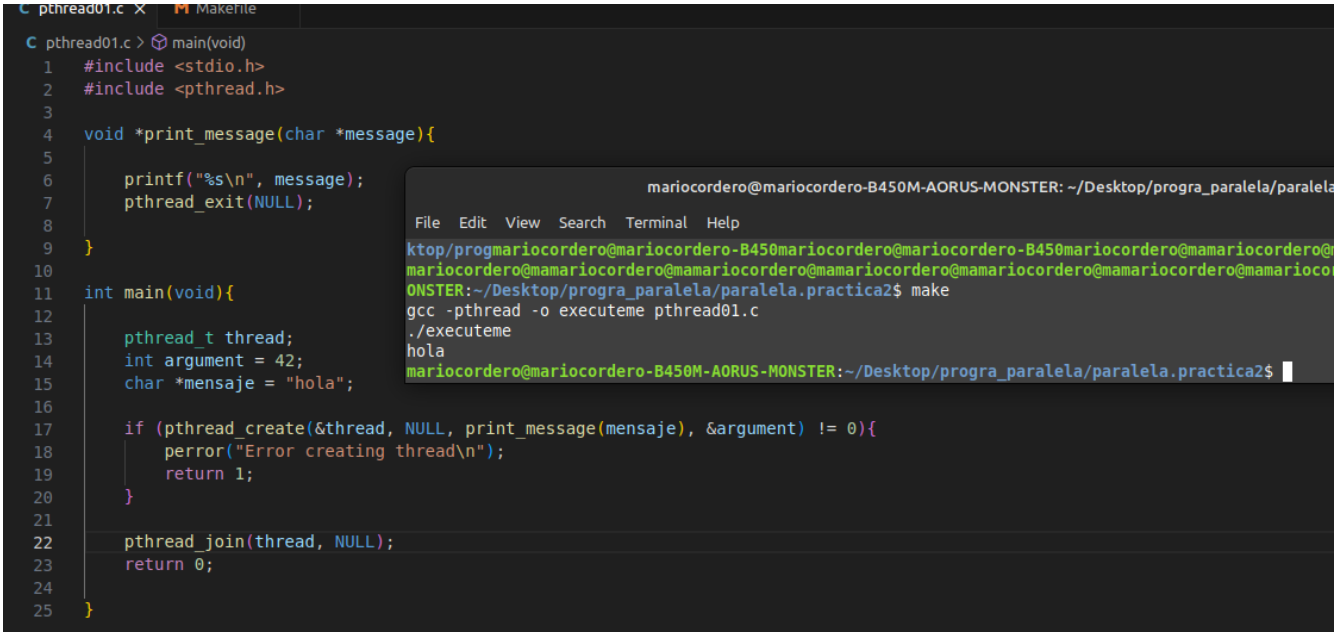
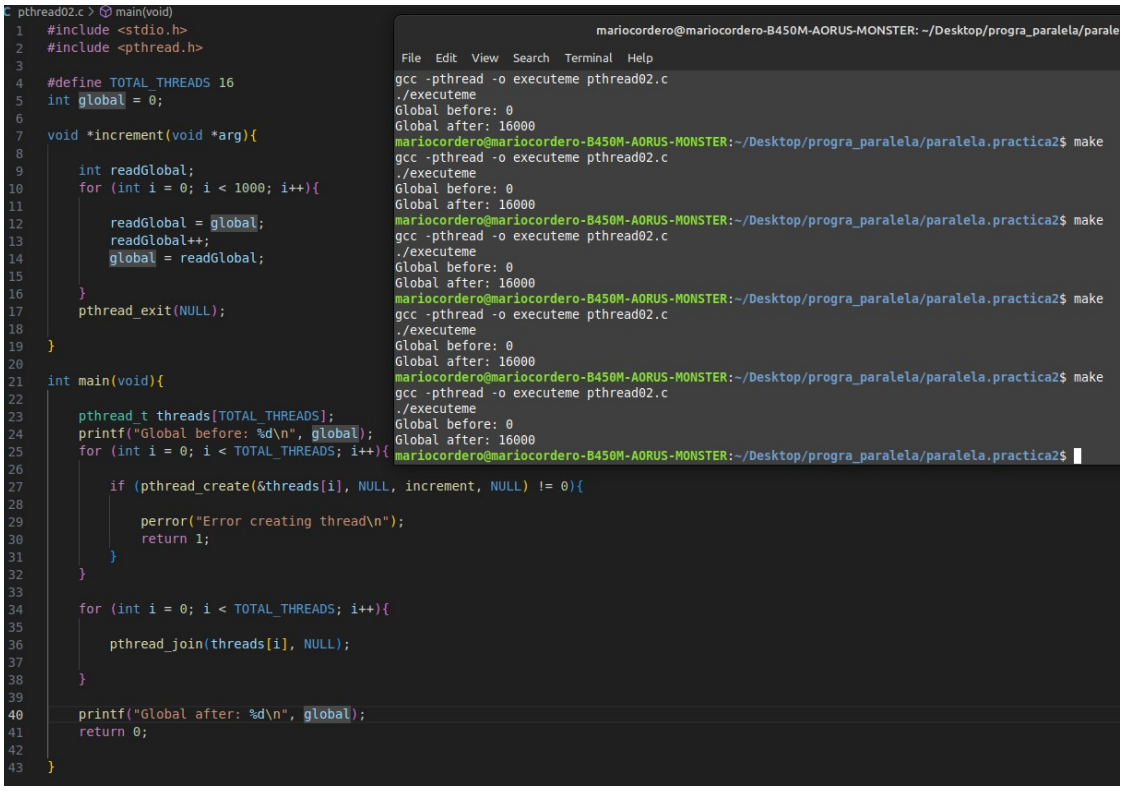
REPORTE

PRIMERA PARTE

Nota: Cabe destacar que hay que enviar una cadena por parametro a la funcion para que todo esto funcione :D



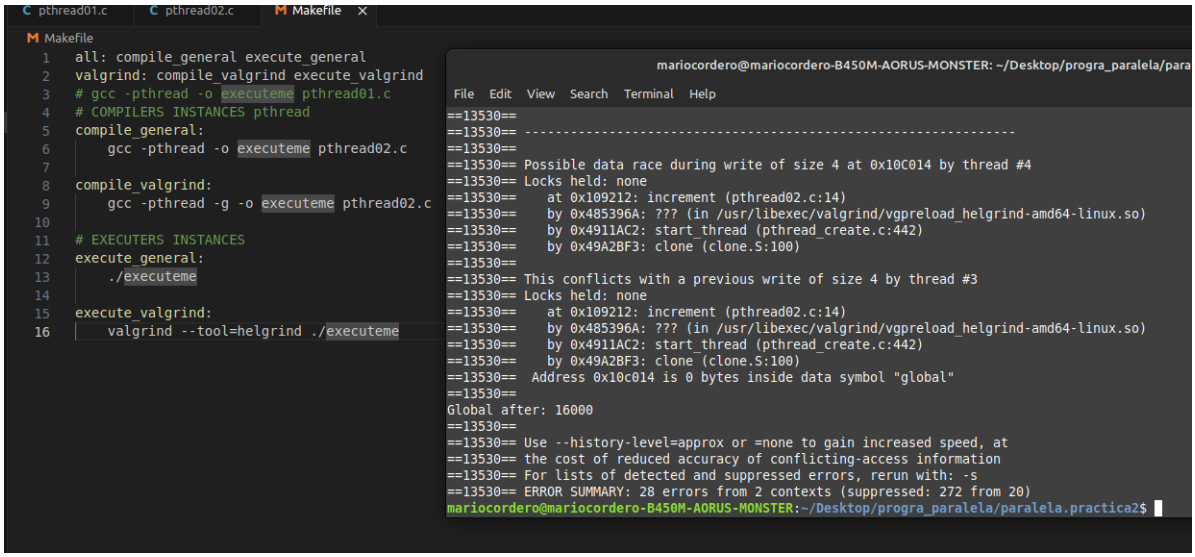
SEGUNDA PARTE



```c
#include <stdio.h>
#include <pthread.h>

#define TOTAL_THREADS 16
int global = 0;

void *increment(void *arg){

    int readGlobal;
    for (int i = 0; i < 1000; i++){

        readGlobal = global;
        readGlobal++;
        global = readGlobal;

    }
    pthread_exit(NULL);

}

int main(void){

    pthread_t threads[TOTAL_THREADS];
    printf("Global before: %d\n", global);
    for (int i = 0; i < TOTAL_THREADS; i++){

        if (pthread_create(&threads[i], NULL, increment, NULL) != 0){

            perror("Error creating thread\n");
            return 1;
        }
    }

    for (int i = 0; i < TOTAL_THREADS; i++){

        pthread_join(threads[i], NULL);

    }

    printf("Global after: %d\n", global);
    return 0;

}
```

```
gcc -pthread -o executeme pthread02.c
./executeme
Global before: 0
Global after: 16000
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ make
gcc -pthread -o executeme pthread02.c
./executeme
Global before: 0
Global after: 16000
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ make
gcc -pthread -o executeme pthread02.c
./executeme
Global before: 0
Global after: 16000
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ make
gcc -pthread -o executeme pthread02.c
./executeme
Global before: 0
Global after: 16000
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ make
gcc -pthread -o executeme pthread02.c
./executeme
Global before: 0
Global after: 16000
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ 
```

Que los hilos comparten la misma memoria pues afectan a la variable global



```makefile
all: compile_general execute_general
valgrind: compile_valgrind execute_valgrind
# gcc -pthread -o executeme pthread01.c
# COMPILERS INSTANCES pthread
compile_general:
    gcc -pthread -o executeme pthread02.c

compile_valgrind:
    gcc -pthread -g -o executeme pthread02.c

# EXECUTERS INSTANCES
execute_general:
    ./executeme

execute_valgrind:
    valgrind --tool=helgrind ./executeme
```

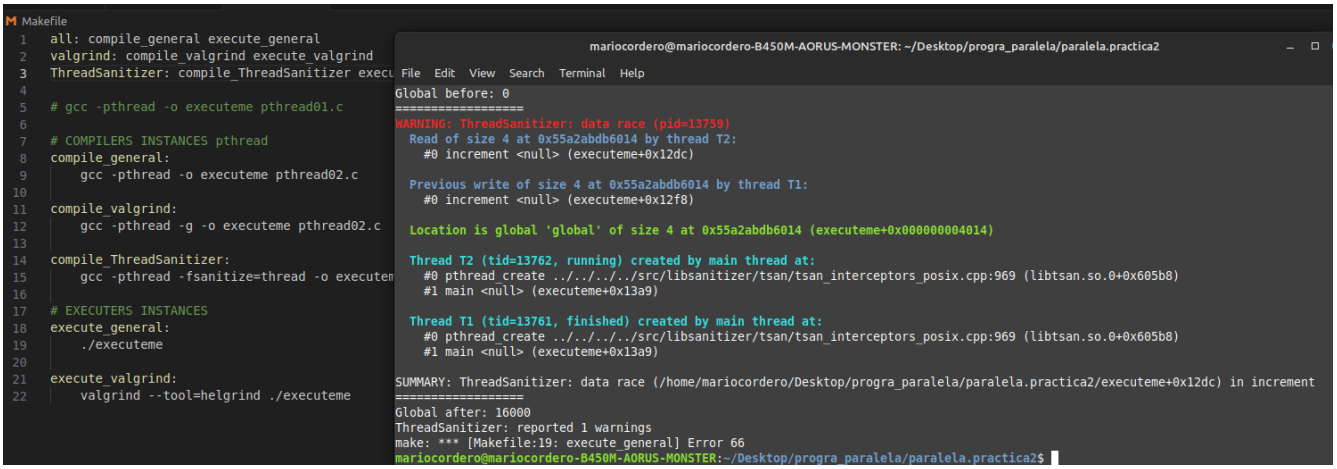
```
==13530==
==13530== ----------------------------------------------------------------
==13530==
==13530== Possible data race during write of size 4 at 0x10C014 by thread #4
==13530== Locks held: none
==13530==    at 0x109212: increment (pthread02.c:14)
==13530==    by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==13530==    by 0x4911AC2: start_thread (pthread_create.c:442)
==13530==    by 0x49A2BF3: clone (clone.S:100)
==13530==
==13530== This conflicts with a previous write of size 4 by thread #3
==13530== Locks held: none
==13530==    at 0x109212: increment (pthread02.c:14)
==13530==    by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==13530==    by 0x4911AC2: start_thread (pthread_create.c:442)
==13530==    by 0x49A2BF3: clone (clone.S:100)
==13530==  Address 0x10c014 is 0 bytes inside data symbol "global"
==13530==
Global after: 16000
==13530==
==13530== Use --history-level=approx or =none to gain increased speed, at
==13530== the cost of reduced accuracy of conflicting-access information
==13530== For lists of detected and suppressed errors, rerun with: -s
==13530== ERROR SUMMARY: 28 errors from 2 contexts (suppressed: 272 from 20)
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ 
```

con valgrind dice que hay 28 errores

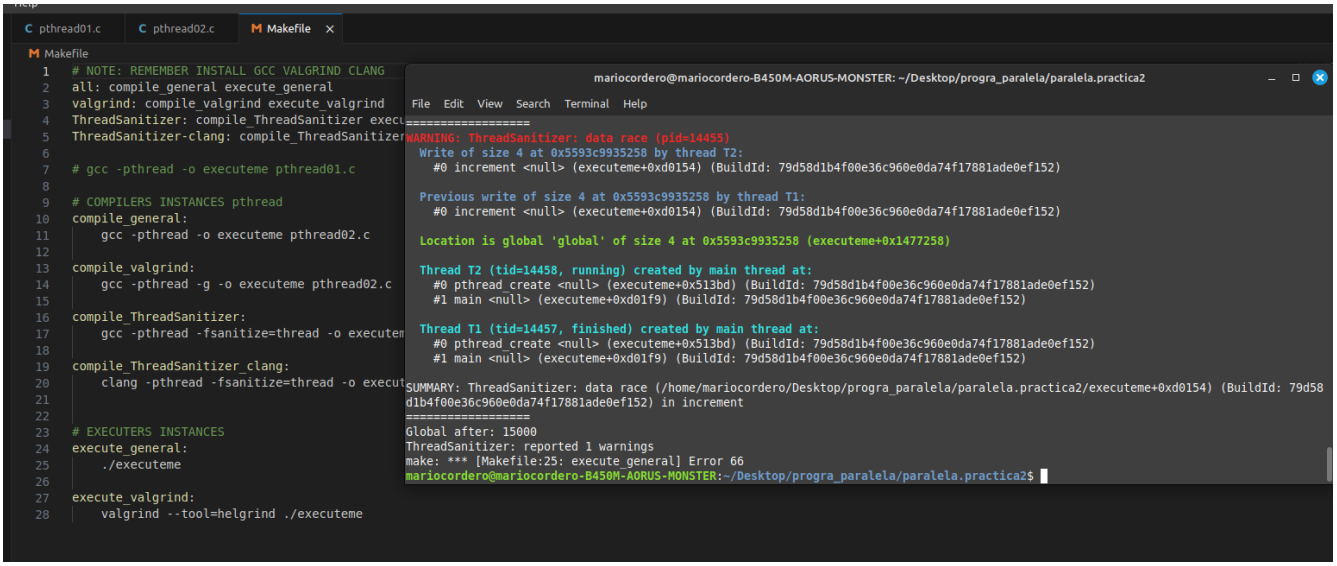
Mario Cordero Aguilar C22306

ThreadSanitizer

con gcc

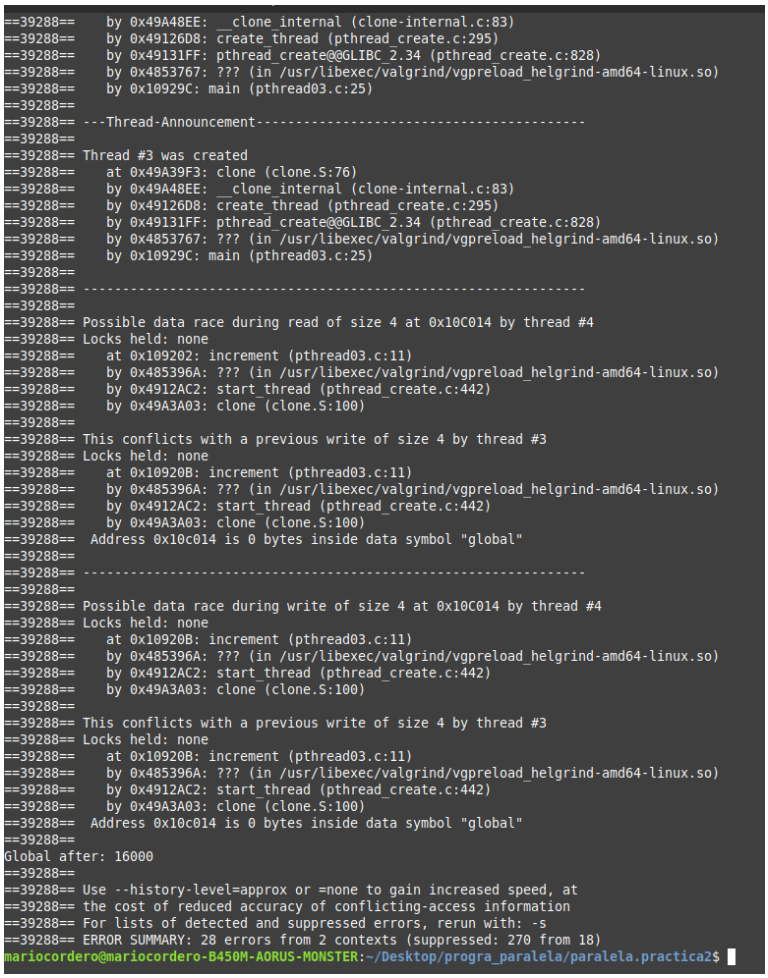


con clang



Conclusión: Reportan los mismos errores

TERCERA PARTE



```
==39288==    by 0x49A48EE: __clone_internal (clone-internal.c:83)
==39288==    by 0x49126D8: create_thread (pthread_create.c:295)
==39288==    by 0x49131FF: pthread_create@@GLIBC_2.34 (pthread_create.c:828)
==39288==    by 0x4853767: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==39288==    by 0x10929C: main (pthread03.c:25)
==39288==
==39288== ---Thread-Announcement----------------------------------------
==39288==
==39288== Thread #3 was created
==39288==    at 0x49A39F3: clone (clone.S:76)
==39288==    by 0x49A48EE: __clone_internal (clone-internal.c:83)
==39288==    by 0x49126D8: create_thread (pthread_create.c:295)
==39288==    by 0x49131FF: pthread_create@@GLIBC_2.34 (pthread_create.c:828)
==39288==    by 0x4853767: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==39288==    by 0x10929C: main (pthread03.c:25)
==39288==
==39288== ----------------------------------------------------------------
==39288==
==39288== Possible data race during read of size 4 at 0x10C014 by thread #4
==39288== Locks held: none
==39288==    at 0x109202: increment (pthread03.c:11)
==39288==    by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==39288==    by 0x4912AC2: start_thread (pthread_create.c:442)
==39288==    by 0x49A3A03: clone (clone.S:100)
==39288==
==39288== This conflicts with a previous write of size 4 by thread #3
==39288== Locks held: none
==39288==    at 0x10920B: increment (pthread03.c:11)
==39288==    by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==39288==    by 0x4912AC2: start_thread (pthread_create.c:442)
==39288==    by 0x49A3A03: clone (clone.S:100)
==39288==  Address 0x10c014 is 0 bytes inside data symbol "global"
==39288==
==39288== ----------------------------------------------------------------
==39288==
==39288== Possible data race during write of size 4 at 0x10C014 by thread #4
==39288== Locks held: none
==39288==    at 0x10920B: increment (pthread03.c:11)
==39288==    by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==39288==    by 0x4912AC2: start_thread (pthread_create.c:442)
==39288==    by 0x49A3A03: clone (clone.S:100)
==39288==
==39288== This conflicts with a previous write of size 4 by thread #3
==39288== Locks held: none
==39288==    at 0x10920B: increment (pthread03.c:11)
==39288==    by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==39288==    by 0x4912AC2: start_thread (pthread_create.c:442)
==39288==    by 0x49A3A03: clone (clone.S:100)
==39288==  Address 0x10c014 is 0 bytes inside data symbol "global"
==39288==
Global after: 16000
==39288==
==39288== Use --history-level=approx or =none to gain increased speed, at
==39288== the cost of reduced accuracy of conflicting-access information
==39288== For lists of detected and suppressed errors, rerun with: -s
==39288== ERROR SUMMARY: 28 errors from 2 contexts (suppressed: 270 from 18)
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$
```

Pasa esto pero no sé que concluir.

INSTRUCCIONES ATOMICAS

```
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ make valgrind
gcc -pthread -g -o executeme pthread03atomic.c
valgrind --tool=helgrind ./executeme
==39638== Helgrind, a thread error detector
==39638== Copyright (C) 2007-2017, and GNU GPL'd, by OpenWorks LLP et al.
==39638== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==39638== Command: ./executeme
==39638==
Global before: 0
Global after: 16000
==39638==
==39638== Use --history-level=approx or =none to gain increased speed, at
==39638== the cost of reduced accuracy of conflicting-access information
==39638== For lists of detected and suppressed errors, rerun with: -s
==39638== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 270 from 18)
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$
```

Usando el `atomic_int global = 0;` ya valgrind no tira ningún error

Sincronización de hilos con mutex

Sin la linea `printf("Thread %ld\n", pthread_self());`

```
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ make valgrind
gcc -pthread -g -o executeme pthread03mutex.c
valgrind --tool=helgrind ./executeme
==40219== Helgrind, a thread error detector
==40219== Copyright (C) 2007-2017, and GNU GPL'd, by OpenWorks LLP et al.
==40219== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==40219== Command: ./executeme
==40219==
Global before: 0
Global after: 16000
==40219==
==40219== Use --history-level=approx or =none to gain increased speed, at
==40219== the cost of reduced accuracy of conflicting-access information
==40219== For lists of detected and suppressed errors, rerun with: -s
==40219== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 42326 from 25)
```

Con la linea `printf("Thread %ld\n", pthread_self());`

```
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Thread 174802496
Global after: 16000
==40159==
==40159== Use --history-level=approx or =none to gain increased speed, at
==40159== the cost of reduced accuracy of conflicting-access information
==40159== For lists of detected and suppressed errors, rerun with: -s
==40159== ERROR SUMMARY: 253 errors from 3 contexts (suppressed: 106829 from 90)
```

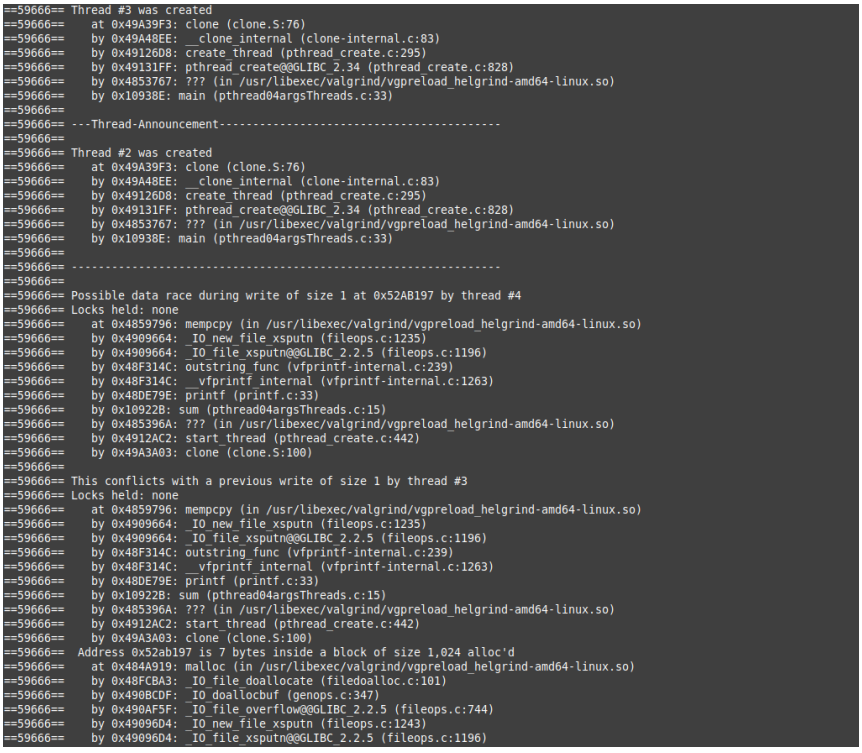
Observo que me da error cuando uso la función **pthread_self();**

Paso de argumentos a un hilo

```
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ make valgrind
gcc -pthread -g -o executeme pthread04args.c
valgrind --tool=helgrind ./executeme
==40625== Helgrind, a thread error detector
==40625== Copyright (C) 2007-2017, and GNU GPL'd, by OpenWorks LLP et al.
==40625== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==40625== Command: ./executeme
==40625==
Result: 65
==40625==
==40625== Use --history-level=approx or =none to gain increased speed, at
==40625== the cost of reduced accuracy of conflicting-access information
==40625== For lists of detected and suppressed errors, rerun with: -s
==40625== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

0 errores con valgrind. Noto que es capaz de enviarle argumentos por medio de Structs al hilo para que este pueda operar con el, me imagino que esto va muy orientado a la programación orientada a objetos y como un hilo puede operar con uno.

Paso de argumentos a múltiples hilos



```
==59666== Thread #3 was created
==59666==    at 0x49A39F3: clone (clone.S:76)
==59666==    by 0x49A48EE: __clone_internal (clone-internal.c:83)
==59666==    by 0x49126D8: create_thread (pthread_create.c:295)
==59666==    by 0x49131FF: pthread_create@@GLIBC_2.34 (pthread_create.c:828)
==59666==    by 0x4853767: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==59666==    by 0x10938E: main (pthread04argsThreads.c:33)
==59666==
==59666== ---Thread-Announcement------------------------------------------
==59666==
==59666== Thread #2 was created
==59666==    at 0x49A39F3: clone (clone.S:76)
==59666==    by 0x49A48EE: __clone_internal (clone-internal.c:83)
==59666==    by 0x49126D8: create_thread (pthread_create.c:295)
==59666==    by 0x49131FF: pthread_create@@GLIBC_2.34 (pthread_create.c:828)
==59666==    by 0x4853767: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==59666==    by 0x10938E: main (pthread04argsThreads.c:33)
==59666==
==59666== ----------------------------------------------------------------
==59666==
==59666== Possible data race during write of size 1 at 0x52AB197 by thread #4
==59666== Locks held: none
==59666==    at 0x4859796: mempcpy (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==59666==    by 0x4909664: _IO_new_file_xsputn (fileops.c:1235)
==59666==    by 0x4909664: _IO_file_xsputn@@GLIBC_2.2.5 (fileops.c:1196)
==59666==    by 0x48F314C: outstring_func (vfprintf-internal.c:239)
==59666==    by 0x48F314C: __vfprintf_internal (vfprintf-internal.c:1263)
==59666==    by 0x48DE79E: printf (printf.c:33)
==59666==    by 0x10922B: sum (pthread04argsThreads.c:15)
==59666==    by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==59666==    by 0x4912AC2: start_thread (pthread_create.c:442)
==59666==    by 0x49A3A03: clone (clone.S:100)
==59666==
==59666== This conflicts with a previous write of size 1 by thread #3
==59666== Locks held: none
==59666==    at 0x4859796: mempcpy (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==59666==    by 0x4909664: _IO_new_file_xsputn (fileops.c:1235)
==59666==    by 0x4909664: _IO_file_xsputn@@GLIBC_2.2.5 (fileops.c:1196)
==59666==    by 0x48F314C: outstring_func (vfprintf-internal.c:239)
==59666==    by 0x48F314C: __vfprintf_internal (vfprintf-internal.c:1263)
==59666==    by 0x48DE79E: printf (printf.c:33)
==59666==    by 0x10922B: sum (pthread04argsThreads.c:15)
==59666==    by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==59666==    by 0x4912AC2: start_thread (pthread_create.c:442)
==59666==    by 0x49A3A03: clone (clone.S:100)
==59666==  Address 0x52ab197 is 7 bytes inside a block of size 1,024 alloc'd
==59666==    at 0x484A919: malloc (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
==59666==    by 0x48FCBA3: _IO_file_doallocate (filedoalloc.c:101)
==59666==    by 0x490BCDF: _IO_doallocbuf (genops.c:347)
==59666==    by 0x490AF5F: _IO_file_overflow@@GLIBC_2.2.5 (fileops.c:744)
==59666==    by 0x49096D4: _IO_new_file_xsputn (fileops.c:1243)
==59666==    by 0x49096D4: _IO_file_xsputn@@GLIBC_2.2.5 (fileops.c:1196)
```

Hice structs manualmente con argumentos distintos para empezar a sumarlos, Valgrind me indica que dos o más hilos están escribiendo en la misma ubicación de memoria al mismo tiempo sin sincronización. En este caso, los hilos parecen estar escribiendo en la salida estándar al mismo tiempo, sin control alguno, lo que llega a causar desincronizaciones o salidas incoherentes. La solución ahorita creo sería un Mutex en la función para que espere a los otros hilos y estén sincronizados.

```
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$ make valgrind
gcc -pthread -g -o executeme pthread04argsThreads.c
valgrind --tool=helgrind ./executeme
==60207== Helgrind, a thread error detector
==60207== Copyright (C) 2007-2017, and GNU GPL'd, by OpenWorks LLP et al.
==60207== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==60207== Command: ./executeme
==60207==
Result: 3
Result: 7
Result: 11
Result: 15
Result: 19
Result: 23
Result: 39
Result: 47
Result: 31
Result: 55
Result: 59
Result: 51
Result: 35
Result: 43
Result: 63
Result: 27
==60207==
==60207== Use --history-level=approx or =none to gain increased speed, at
==60207== the cost of reduced accuracy of conflicting-access information
==60207== For lists of detected and suppressed errors, rerun with: -s
==60207== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 368 from 25)
mariocordero@mariocordero-B450M-AORUS-MONSTER:~/Desktop/progra_paralela/paralela.practica2$
```

Lo propuesto anterirormente funcionó, ya supe usar valgrind :DDD