

# Lab 3

## Respuestas

1. Muestra las direcciones de memoria de las líneas que esta ejecutando, además muestra la dirección y nombre de la variable donde se genera la carrera.

```
Possible data race during read of size 4 at 0x10C014 by thread #1
```

```
Locks held: none
```

```
at 0x109236: main (in /home/jassito/S0/Lab3/main-race)
```

```
This conflicts with a previous write of size 4 by thread #2
```

```
Locks held: none
```

```
at 0x1091BE: worker (in /home/jassito/S0/Lab3/main-race)
```

```
by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
```

```
by 0x48FCAC2: start_thread (pthread_create.c:442)
```

```
by 0x498DA03: clone (clone.S:100)
```

```
Address 0x10c014 is 0 bytes inside data symbol "balance"
```

```
Possible data race during write of size 4 at 0x10C014 by thread #1
```

```
Locks held: none
```

```
at 0x10923F: main (in /home/jassito/S0/Lab3/main-race)
```

```
This conflicts with a previous write of size 4 by thread #2
```

```
Locks held: none
```

```
at 0x1091BE: worker (in /home/jassito/S0/Lab3/main-race)
```

```
by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
```

```
by 0x48FCAC2: start_thread (pthread_create.c:442)
```

```
by 0x498DA03: clone (clone.S:100)
```

```
Address 0x10c014 is 0 bytes inside data symbol "balance"
```

2. Al proteger o eliminar la variable compartida el programa no muestra ningún error.

```
int balance = 0;
static pthread_mutex_t candado;

void *worker(void *arg) {
    pthread_mutex_lock(&candado);
    balance++; // unprotected access
    pthread_mutex_unlock(&candado);
    return NULL;
}

int main(int argc, char *argv[]) {
    pthread_t p;
    Pthread_create(&p, NULL, worker, NULL);
    // balance++; // unprotected access
    Pthread_join(p, NULL);
    return 0;
}
```

```
→ Lab3 valgrind --tool=helgrind ./main-race
=15563= Helgrind, a thread error detector
=15563= Copyright (C) 2007-2017, and GNU GPL'd, by OpenWorks LLP et al.
=15563= Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
=15563= Command: ./main-race
=15563=
=15563= int main(int argc, char *argv[]) {
=15563=     pthread_t p;
=15563= Use --history-level=approx or =none to gain increased speed, at
=15563= the cost of reduced accuracy of conflicting-access information
=15563= For lists of detected and suppressed errors, rerun with: -s
=15563= ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

3. Me parece que el problema se genera al bloquear dos veces una región, por lo que podría generar algún tipo de error.
4. Helgrind notifica que el hilo viola el orden de bloqueo, ya que el orden de bloqueo fue diferente al inicial.

```

Thread #3: lock order "0x10C040 before 0x10C080" violated
pthread_mutex_t m1 = PTHREAD_MUTEX_INITIALIZER;
Observed (incorrect) order is: acquisition of lock at 0x10C080
  at 0x4850CCF: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x109288: worker (in /home/jassito/SO/Lab3/main-deadlock)
  by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x48FCAC2: start_thread (pthread_create.c:442)
  by 0x498DA03: clone (clone.S:100)
  followed by a later acquisition of lock at 0x10C040
  at 0x4850CCF: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x1092C3: worker (in /home/jassito/SO/Lab3/main-deadlock)
  by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x48FCAC2: start_thread (pthread_create.c:442)
  by 0x498DA03: clone (clone.S:100)

int argc, char *argv[] {
  Required order was established by acquisition of lock at 0x10C040
  at 0x4850CCF: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x10920E: worker (in /home/jassito/SO/Lab3/main-deadlock)
  by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x48FCAC2: start_thread (pthread_create.c:442)
  by 0x498DA03: clone (clone.S:100)

  followed by a later acquisition of lock at 0x10C080
  at 0x4850CCF: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x109249: worker (in /home/jassito/SO/Lab3/main-deadlock)

```

5. Muestra el mismo error que en el programa anterior. Me parece que es una herramienta sumamente útil para verificar si existen problemas relacionados a variables compartidas o hilos, la información que muestra es completa y útil.

```

Thread #3: lock order "0x10C080 before 0x10C0C0" violated
Observed (incorrect) order is: acquisition of lock at 0x10C0C0
  at 0x4850CCF: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x1092C3: worker (in /home/jassito/SO/Lab3/main-deadlock-global)
  by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x48FCAC2: start_thread (pthread_create.c:442)
  by 0x498DA03: clone (clone.S:100)
  followed by a later acquisition of lock at 0x10C080
  at 0x4850CCF: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x1092FE: worker (in /home/jassito/SO/Lab3/main-deadlock-global)
  by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x48FCAC2: start_thread (pthread_create.c:442)
  by 0x498DA03: clone (clone.S:100)

Required order was established by acquisition of lock at 0x10C080
  at 0x4850CCF: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x109249: worker (in /home/jassito/SO/Lab3/main-deadlock-global)
  by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x48FCAC2: start_thread (pthread_create.c:442)
  by 0x498DA03: clone (clone.S:100)

  followed by a later acquisition of lock at 0x10C0C0
  at 0x4850CCF: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x109284: worker (in /home/jassito/SO/Lab3/main-deadlock-global)
  by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)

```

6. El programa no es eficiente ya que se generan spinlocks. El hilo que espera a que el hijo termine se queda sin hacer nada, gastando ciclos de CPU de manera ineficiente.
7. Muestra que pueden existir problemas de carrera en la variable "done".

```
Possible data race during read of size 4 at 0x10C014 by thread #1
Locks held: none
  at 0x109245: main (in /home/jassito/SO/Lab3/main-signal)

This conflicts with a previous write of size 4 by thread #2
Locks held: none
  at 0x1091C8: worker (in /home/jassito/SO/Lab3/main-signal)
  by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x48FCAC2: start_thread (pthread_create.c:442)
  by 0x498DA03: clone (clone.S:100)
Address 0x10c014 is 0 bytes inside data symbol "done"
```

8. Esta versión es correcta, pero puede llegar a ser ineficiente con una cantidad considerable de hilos, ya que de igual manera se pueden generar spinlocks.
9. No muestra ningún error.

```
→ Lab3 valgrind --tool=helgrind ./main-signal-cv
==23706== Helgrind, a thread error detector
==23706== Copyright (C) 2007-2017, and GNU GPL'd, by OpenWorks LLP et al.
==23706== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==23706== Command: ./main-signal-cv
==23706==
this should print first
this should print last
==23706==
==23706== Use --history-level=approx or =none to gain increased speed, at
==23706== the cost of reduced accuracy of conflicting-access information
==23706== For lists of detected and suppressed errors, rerun with: -s
==23706== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 7 from 7)
```