

1.

Critical section not locked: balance ++

```
ubuntu@ip-172-26-10-114:~/Computer-System/chapter26-28$ valgrind --tool=helgrind ./main-race
==201510== Helgrind, a thread error detector
==201510== Copyright (C) 2007-2017, and GNU GPL'd, by OpenWorks LLP et al.
==201510== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==201510== Command: ./main-race
==201510==
```

```

Possible data race during read of size 4 at 0x10C014 by thread #1
Locks held: none
   at 0x109236: main (in /home/ubuntu/Computer-System/chapter26-28/main-race)

This conflicts with a previous write of size 4 by thread #2
Locks held: none
   at 0x1091BE: worker (in /home/ubuntu/Computer-System/chapter26-28/main-race)
   by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
   by 0x48FCAC2: start_thread (pthread_create.c:442)
   by 0x498DBF3: clone (clone.S:100)
Address 0x10c014 is 0 bytes inside data symbol "balance"
```

2.

One of balance ++ removed

```
int balance = 0;
void* worker(void* arg) {
    balance++; // unprotected access
    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;
}
```

0 error after removal.

Add one lock: still possible data race

```
void* worker(void* arg) {
    pthread_mutex_lock(&m1);
    balance++; // unprotected access
    pthread_mutex_unlock(&m1);
    return NULL;
}
```

Add 2 locks: no possible data race

3. lock order violated

4.

```
Thread #3: lock order "0x10C040 before 0x10C080" violated
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/ubuntu/Computer-System/chapter26-28/main-deadlock)

  by 0x485396A: ??? (in /usr/libexec/valgrind/vgpreload_helgrind-amd64-linux.so)
  by 0x48FCAC2: start_thread (pthread_create.c:442)
  by 0x498DBF3: clone (clone.S:100)      I
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/ubuntu/Computer-System/chapter26-28/main-deadlock)
```

5.

Yes it still has lock order violated error, and helgrind reports it.

Check deadlock using reliable tools in multi-threaded programs.

6.

Because when the parent (main) waits, it is in a loop checking the signal "done" to become 1.

7. Report:

```
Possible data race during read of size 4 at 0x10C014 by thread #1
Locks held: none
  at 0x109245: main (in /home/ubuntu/Computer-System/chapter26-28/main-signal)
This conflicts with a previous write of size 4 by thread #2
```

The code still has errors.

8-9. Both correctness and performance. Helgrind doesn't show errors.