```
std::string s = "Hello world ";
//complex object passed by reference
std::thread t(complex_fnt, std::ref(s));

//detach the thread... if the program exits before the threads
completes its job, no job is done
t.detach();

//wait 5ms in order to assure that file is written on disk
std::this_thread::sleep_for(std::chrono::nanoseconds(x));
```

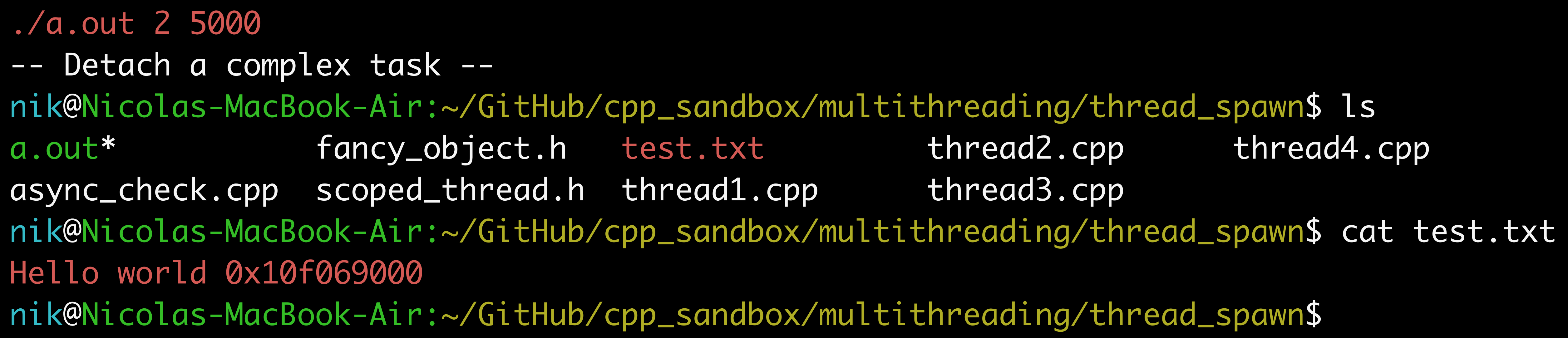
join vs detach

```
std::string s = "Hello world ";
//complex object passed by reference
std::thread t(complex_fnt, std::ref(s));

//detach the thread... if the program exits before the threads
completes its job, no job is done
t.detach();

//wait 5ms in order to assure that file is written on disk
std::this_thread::sleep_for(std::chrono::nanoseconds(x));
```

join vs detach



```
./a.out 2 5000
-- Detach a complex task --
nik@Nicolas-MacBook-Air:~/GitHub/cpp_sandbox/multithreading/thread_spawn$ ls
a.out*           fancy_object.h   test.txt         thread2.cpp      thread4.cpp
async_check.cpp  scoped_thread.h  thread1.cpp      thread3.cpp
nik@Nicolas-MacBook-Air:~/GitHub/cpp_sandbox/multithreading/thread_spawn$ cat test.txt
Hello world 0x10f069000
nik@Nicolas-MacBook-Air:~/GitHub/cpp_sandbox/multithreading/thread_spawn$
```