


```
void complex_fnt( std::string& s)
{
    std::stringstream buff;
    std::ofstream f("./test.txt");
    //write to file string passed + thread id
    if(f.is_open())
    {
        buff << s << std::this_thread::get_id() << std::endl;
        f << buff.str();
        f.close();
    }
}
```

```
std::string s = "Hello world ";
```

```
//complex object passed by reference
```


```
std::thread t(complex_fnt, std::ref(s));
```

```
//assure job is done and return
```

```
t.join();
```



join vs detach



```
void complex_fnt( std::string& s)
{
    std::stringstream buff;
    std::ofstream f("./test.txt");
    //write to file string passed + thread id
    if(f.is_open())
    {
        buff << s << std::this_thread::get_id() << std::endl;
        f << buff.str();
        f.close();
    }
}
```

```
std::string s = "Hello world ";
```

```
//complex object passed by reference
std::thread t(complex_fnt, std::ref(s));
```

```
//assure job is done and return
t.join();
```

join vs detach

```
nik@Nicolas-MacBook-Air:~/GitHub/cpp_sandbox/multithreading/thread_spawn$ ./a.out 1
-- Join 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 0x103c27000
nik@Nicolas-MacBook-Air:~/GitHub/cpp_sandbox/multithreading/thread_spawn$
```