```cpp
std::vector<std::thread> threads;
threads.emplace_back(fnt,0,"Hi");
threads.emplace_back(fnt,1,"Salut");
threads.emplace_back(fnt,2,"Ciao");
threads.emplace_back(fnt,3,"Hola");

for(auto& t: threads )
  t.join();
```

```
void fnt(int id, std::string s)
{
    std::cout << "id # = " << id
              << " - Functional object - I am a thread with ID = "
              << std::this_thread::get_id()
              << " custom msg = "<< s
              <<std::endl;
}
```

std::mutex and std::lock_guard

```cpp
void fnt(int id, std::string s)
{
   std::cout << "id # = " << id
         << " - Functional object - I am a thread with ID = "
         << std::this_thread::get_id()
         << " custom msg = "<< s
         <<std::endl;
}
```

```cpp
std::vector<std::thread> threads;
threads.emplace_back(fnt,0,"Hi");
threads.emplace_back(fnt,1,"Salut");
threads.emplace_back(fnt,2,"Ciao");
threads.emplace_back(fnt,3,"Hola");

for(auto& t: threads )
  t.join();
```

Output … ???

```
nik@Nicolas-MacBook-Air:~/GitHub/cpp_sandbox/multithreading/thread_spawn$
clang++ -std=c++11 thread4.cpp
nik@Nicolas-MacBook-Air:~/GitHub/cpp_sandbox/multithreading/thread_spawn$
./a.out  0

iiiidddd    ####    ====    0123    ----
FFFFuuuunnnncccctttttiiiioooonnnnaaaallll    oooobbbbjjjjeeeecccctttt    ----
IIII    aaaammmm    aaaa    tttthhhhrrrreeeeaaaadddd    wwwwiiiittttthhhh
IIIIDDDD    ====    0000xxxx1111000077773344080814 7a000000000000
ccccuuuusssstttttoooommmm    mmmmssssgggg    ====    HSCHiaio
laluoat


nik@Nicolas-MacBook-Air:~/GitHub/cpp_sandbox/multithreading/thread_spawn$
```