

ITF22519 Introduction to Operating Systems

Lab 6: Thread Programming 1

Exercise 1.

```
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ cc -pthread Exercise1.c -o Exercise1
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ ./Exercise1
Hello from main
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ ./Exercise1
Hello from main
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ ./Exercise1
Hello from main
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ ./Exercise1
Hello from main
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ ./Exercise1
Hello from main
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ ./Exercise1
Hello from main
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ ./Exercise1
Hello from main
```

There is no guarantee that the messages from the other threads (Thread1, Thread2) will be printed at all. This is because the main thread will usually exit before the other threads have been created and executed their functions. We could remedy this by calling `pthread_join` on both threads. When we call `pthread_join` in our main function we force the main thread to wait until the specified thread is terminated before it continues executing.

Exercise 2.

```
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ cc -pthread Exercise2.c -o Exercise2
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$ ./Exercise2
Your balance is: 1000
Enter deposit amount: 499
Your balance is now: 1499
jrlundqv@IdeaPad5:~/OneDrive/22høst/OS/labs/lab6$
```