ESIR SPP –TP5 Bonus Work (non noté)

Exercise:

The goal of this exercise is to experiment with the POSIX threading library in C. C is similar to C++, but without any objects.

Task 1: Making sure your C installation works

Type the following program in a file test.c

```
#include <stdio.h>
int main(int argc, char **argv)
{
    printf("hello, world\n");
    return 0;
}
```

If you are working on windows, use gcc from within a cygwin terminal to compile this program. A cygwin terminal should be available under the "Émulateur" menu on your machine.

```
gcc test.c
```

You can now execute your program using (on Windows):

```
./a.exe
```

On Linux use:

```
./a.out
```

Notes:

- Your cygwin terminal runs a unix shell (Bash, the Linux shell). To find out how to manipulate files and directories from within a shell, look at http://www.ks.uiuc.edu/Training/Tutorials/Reference/unixprimer.html.

Task 2: Using Posix Threads

Have a look at the following tutorial to get acquainted with the C POSIX thread library:

http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html

Using this tutorial, write a multithreaded program in C in which 5 POSIX threads increment one shared long variable 1,000,000 times.

1 – Write a first version of your program without any synchronisation. What do you observe? Why do you think this is the case? Measure the execution time taken by your program.

2 – Make your program thread-safe with a POSIX lock. Measure the execution time taken by your program. Compare this time to the unsafe version.