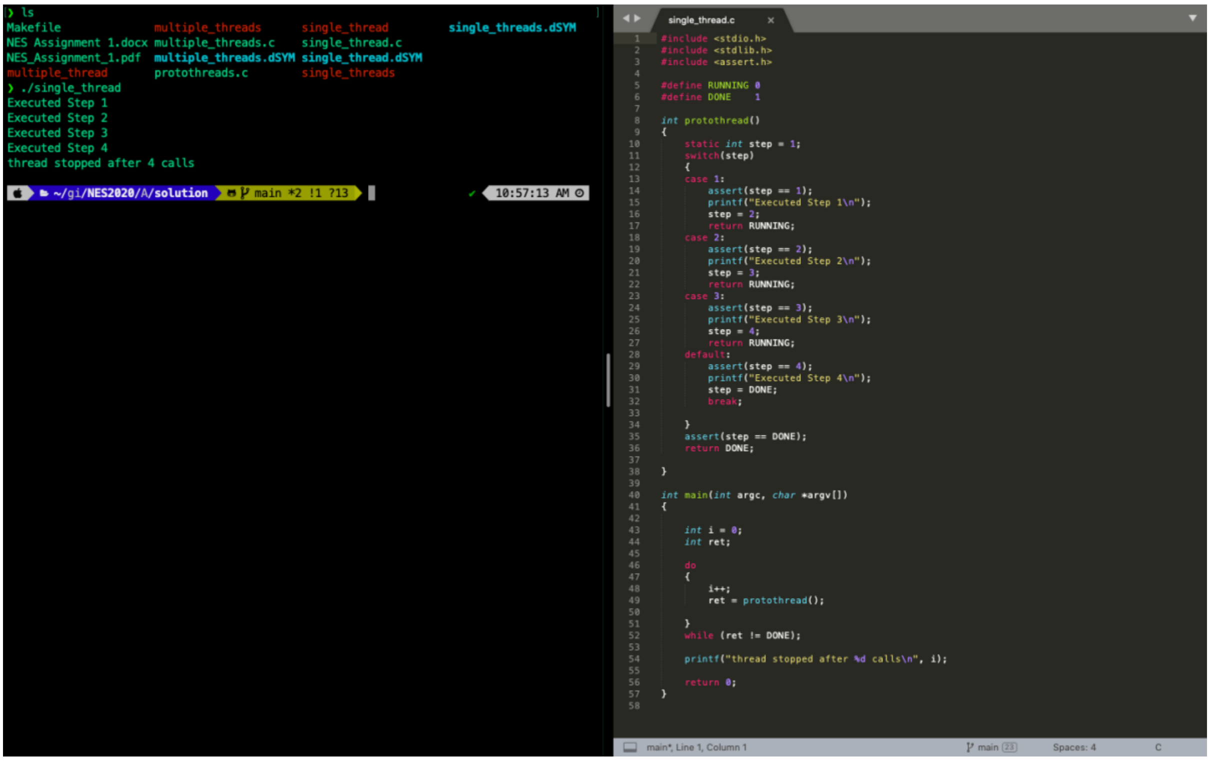
NES Assignment – 1

1. Single_thread.c

Description:

In this program, A single thread is created with a switch case to redirect to different execution step where a message is printed to show the current execution step. The execution is stopped when the thread returns DONE which is a constant 1.

Output screenshots:



2. Multithreading:

Description:

We added 2 protothreads here. The first with 4 steps/cases, each printing a message indicating the thread ID and the execution step. and the second with 2 execution steps, printing similar output.

Extend one protothread with a local variable (for example an integer that is incremented in each execution step). Output also the current value of the variable in each step. Reflect on the problems that might occur here. Why is the naive approach not working? Try to pinpoint the core of the problem to understand when normal local variables can be used and when that's not possible.

Implementation:

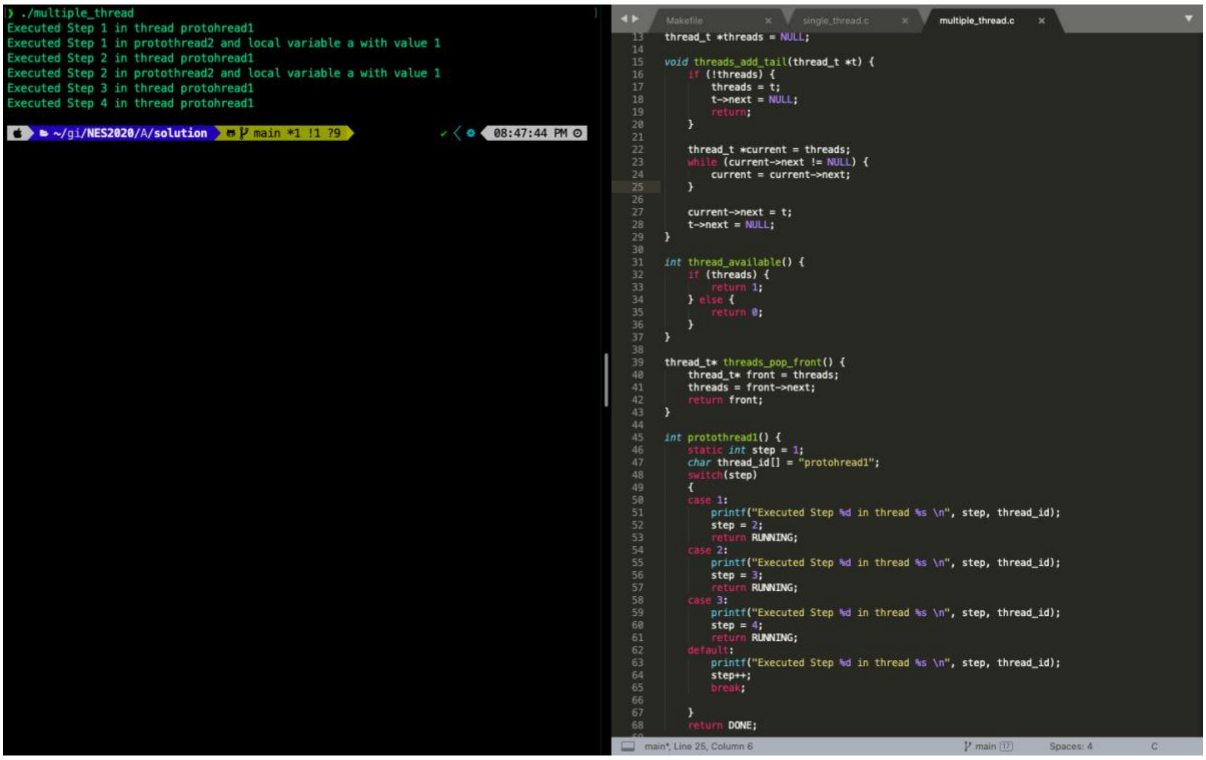
We have implemented a protothread with a local variable and the value of the local variable is changed in each execution step and printed in the output.

Observation:

The changed value of the local variable is not reflected in the output because the values of the local variables are not preserved but in the case of static variables the value is preserved across multiple executions.

Local variables have block scope so cannot be used across multiple execution. Local variables gets destroyed once block ends. It can be used to perform simple operations in a single execution.

Output screenshots:



3. **Protothreads.c:**

Descritpion:

Here we have implemented 3 protothreads. The first one is automatically started and it calls protothread 3 once a button is clicked. The second protothread is automatically started and prints a message/ The third protothread is started by protothread 1 only when the button is clicked and it prints a message.

You should now be able to answer questions like: • How are protothreads different from, for example, Linux threads?

- Protothreads ate stackless threads while Linux threads use stacks. This saves the overhead of large amounts of memory space,as it is anyway not required for Wireless sensor networks.
- Protothreads are independent of Operating system.

- Linux threads need a scheduler/handler for execution of threads but scheduling in case of protothreads have to be taken care by the developer.

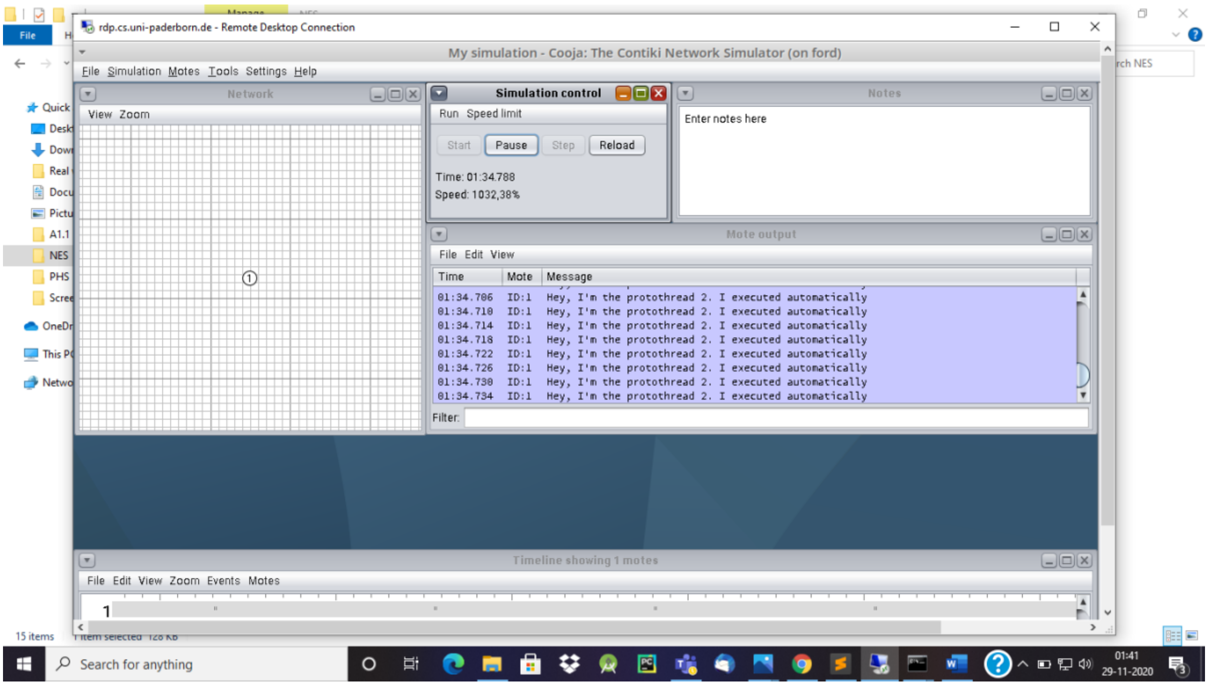
Would it be possible to just reimplement Linux threads for our sensor motes?

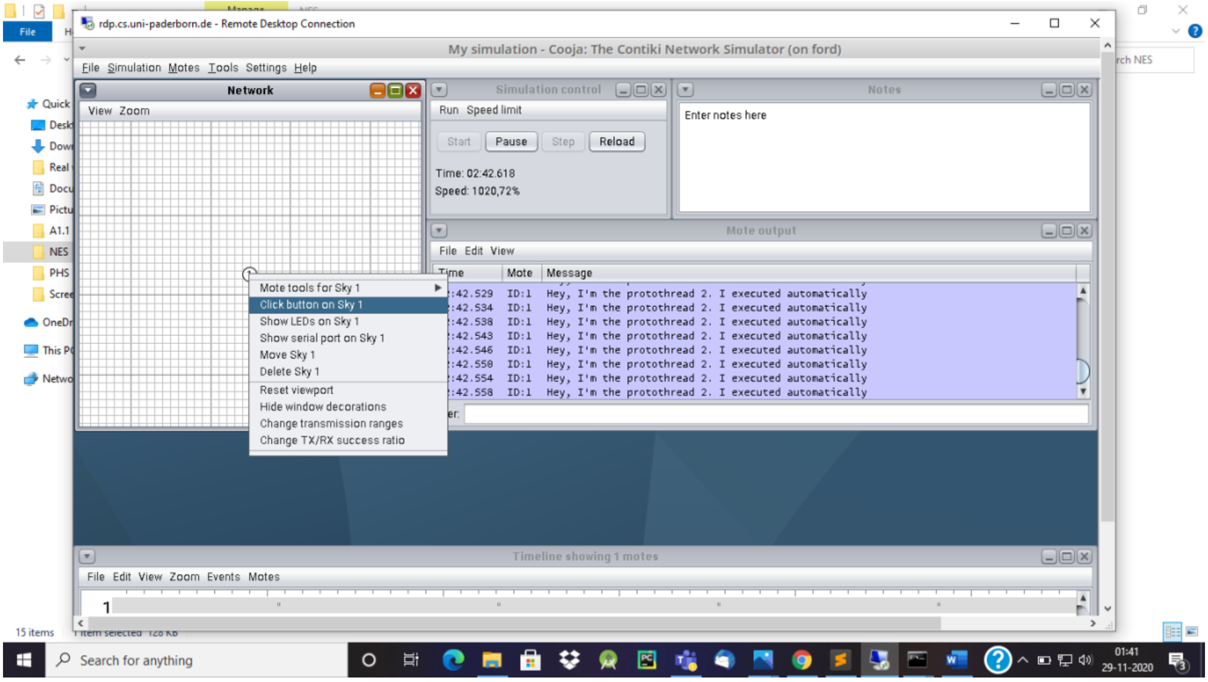
It is possible but it again requires large amounts of memory space.

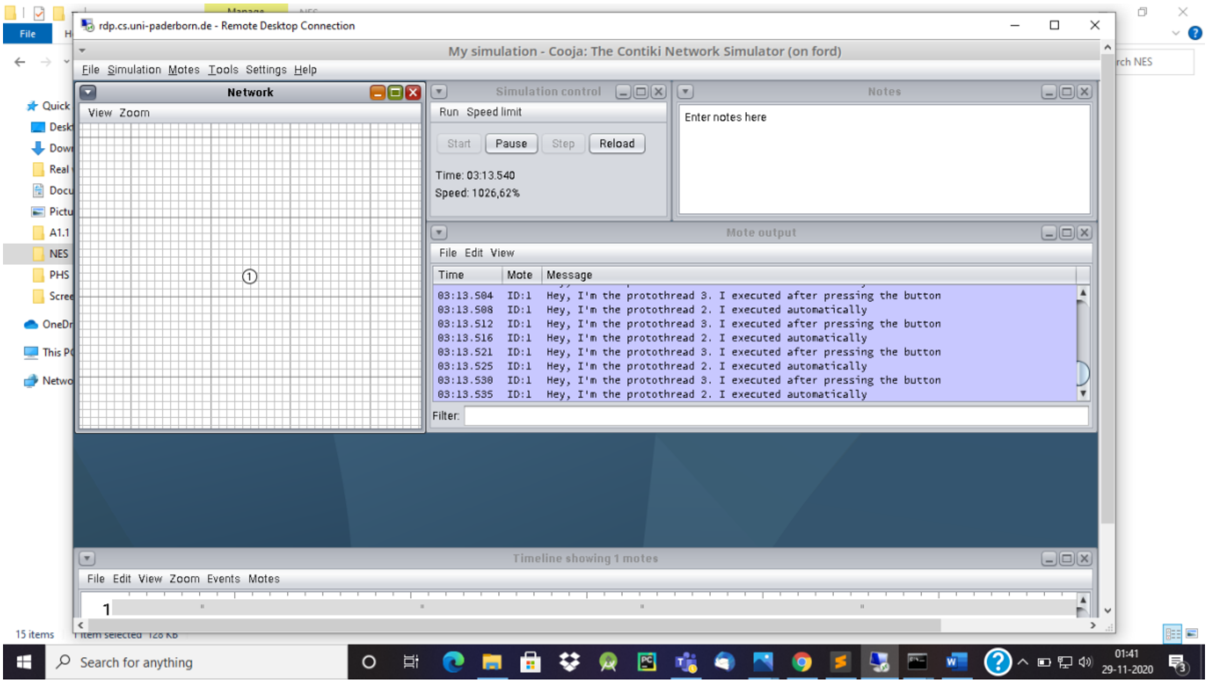
Protothread macros:

- *PROCESS_BEGIN()* - The process begins with this macro. The process thread starts from here.
- *PROCESS_END()* - This indicates the end of the process. The process is then removed from Kernel's list of active processes.
- *PROCESS_EXIT()* - A process can end in 2 ways. When it reaches PROCESS_END() macro,or when another process calls process_exit() function.
- *PROCESS_WAIT_EVENT()* - It waits for an event to occur,it could be any event.
- *PROCESS_WAIT_EVENT_UNTIL()* - It waits for an event to occur,but the occurance of event is specified with conditions.
- *PROCESS_YIELD()* - This macro also waits for any event to occur.
- *PROCESS_WAIT_UNTIL()* - It only waits for a condition. It might not yield the process.
- *PROCESS_PAUSE()* - It yields the process temporarily.

Output Screenshots:




rdp.cs.uni-paderborn.de - Remote Desktop Connection
My simulation - Cooja: The Contiki Network Simulator (on ford)
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
Start Pause Step Reload
Time: 02:42.618
Speed: 1020,72%
Notes
Enter notes here
Mote output
File Edit View
Time Mote Message
:42.529 ID:1 Hey, I'm the protothread 2. I executed automatically
:42.534 ID:1 Hey, I'm the protothread 2. I executed automatically
:42.538 ID:1 Hey, I'm the protothread 2. I executed automatically
:42.543 ID:1 Hey, I'm the protothread 2. I executed automatically
:42.546 ID:1 Hey, I'm the protothread 2. I executed automatically
:42.550 ID:1 Hey, I'm the protothread 2. I executed automatically
:42.554 ID:1 Hey, I'm the protothread 2. I executed automatically
:42.558 ID:1 Hey, I'm the protothread 2. I executed automatically
Mote tools for Sky 1
Click button on Sky 1
Show LEDs on Sky 1
Show serial port on Sky 1
Move Sky 1
Delete Sky 1
Reset viewport
Hide window decorations
Change transmission ranges
Change TX/RX success ratio
Timeline showing 1 motes
File Edit View Zoom Events Motes
Search for anything
01:41
29-11-2020


rdp.cs.uni-paderborn.de - Remote Desktop Connection
My simulation - Cooja: The Contiki Network Simulator (on ford)
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
Start Pause Step Reload
Time: 03:13.540
Speed: 1026,62%
Notes
Enter notes here
Mote output
File Edit View
Time Mote Message
03:13.504 ID:1 Hey, I'm the protothread 3. I executed after pressing the button
03:13.508 ID:1 Hey, I'm the protothread 2. I executed automatically
03:13.512 ID:1 Hey, I'm the protothread 3. I executed after pressing the button
03:13.516 ID:1 Hey, I'm the protothread 2. I executed automatically
03:13.521 ID:1 Hey, I'm the protothread 3. I executed after pressing the button
03:13.525 ID:1 Hey, I'm the protothread 2. I executed automatically
03:13.530 ID:1 Hey, I'm the protothread 3. I executed after pressing the button
03:13.535 ID:1 Hey, I'm the protothread 2. I executed automatically
Filter:
Timeline showing 1 motes
File Edit View Zoom Events Motes
Search for anything
01:41
29-11-2020