

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із лабораторної роботи №1
із дисципліни «Розподілені і хмарні обчислення»

Виконав:

студент групи КМ-01

Романецький М.С.

Керівник:

Доцент кафедри ПМА

Ліскін В. О.

Мета роботи: навчитись працювати з потоками

Опис програми: Для демонстрації роботи буде продемонстрована програма з користувацьким інтерфейсом інкремент/декремент. Користувач може змінювати пріоритет потоку на максимальний або мінімальний. Операції виконуються в різних потоках.

```
Decrementing: -1  
Decrementing: -2  
Decrementing: -3  
Incrementing: -2  
Incrementing: -1  
Incrementing: 0  
Incrementing: 1  
Incrementing: 2  
Incrementing: 3  
Incrementing: 4  
Incrementing: 5  
Incrementing: 6  
Incrementing: 7  
Incrementing: 8  
Incrementing: 9  
Incrementing: 10  
Decrementing: 9  
Decrementing: 8  
Decrementing: 7  
Decrementing: 6  
Decrementing: 5  
Decrementing: 4  
Decrementing: 3  
Decrementing: 2  
Decrementing: 1  
Decrementing: 0  
Decrementing: -1  
Decrementing: -2  
Decrementing: -3  
Decrementing: -4  
Decrementing: -5  
Decrementing: -6  
Decrementing: -7
```

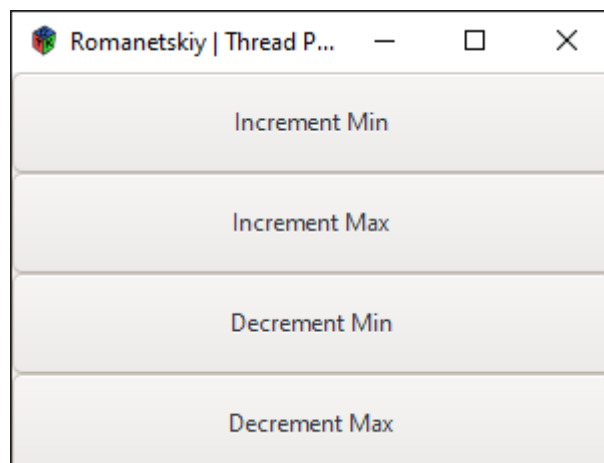
Рівний пріоритет

```
Incrementing: 1  
Incrementing: 2  
Incrementing: 3  
Incrementing: 4  
Decrementing: 3  
Decrementing: 2  
Decrementing: 1  
Decrementing: 0  
Decrementing: -1  
Decrementing: -2  
Decrementing: -3  
Decrementing: -4  
Decrementing: -5  
Decrementing: -6  
Decrementing: -7  
Decrementing: -8  
Decrementing: -9  
Decrementing: -10  
Decrementing: -11  
Decrementing: -12  
Decrementing: -13  
Decrementing: -14  
Decrementing: -15  
Decrementing: -16  
Decrementing: -17  
Decrementing: -18  
Decrementing: -19
```

Пріоритет
декременту

```
Decrementing: -1  
Decrementing: -2  
Decrementing: -3  
Decrementing: -4  
Decrementing: -5  
Decrementing: -6  
Decrementing: -7  
Decrementing: -8  
Decrementing: -9  
Decrementing: -10  
Decrementing: -11  
Decrementing: -12  
Decrementing: -13  
Incrementing: -12  
Incrementing: -11  
Incrementing: -10  
Incrementing: -9  
Incrementing: -8  
Incrementing: -7  
Incrementing: -6  
Incrementing: -5  
Incrementing: -4  
Incrementing: -3  
Incrementing: -2  
Incrementing: -1  
Incrementing: 0  
Incrementing: 1  
Incrementing: 2  
Incrementing: 3  
Incrementing: 4  
Incrementing: 5  
Incrementing: 6  
Incrementing: 7  
Incrementing: 8  
Incrementing: 9  
Incrementing: 10
```

Пріоритет інкременту



Користувацький інтерфейс

Код програми:

```
// cd /d/KPI/Distributed_computing/Labs/lab_1

#[macro_use]
extern crate lazy_static;

use gtk::prelude::*;
use gtk::{Button, Window, WindowType, Box, Orientation};
use std::sync::{Arc, Mutex};
use std::thread;
use std::time::Duration;
use thread_priority::*;
use std::os::windows::io::AsRawHandle;
use winapi::ctypes::c_void as winapi_c_void;

lazy_static! {
    static ref PRIORITY_MIN: ThreadPriority =
ThreadPriority::Os(WinAPIThreadPriority::Lowest.into());
    static ref PRIORITY_MAX: ThreadPriority =
ThreadPriority::Os(WinAPIThreadPriority::Highest.into());
}

fn increment(refr: Arc<Mutex<i32>>) {
    loop {
        let mut num = refr.lock().unwrap();
        *num += 1;
        println!("Incremented: {}", num);
        thread::sleep(Duration::from_millis(100));
    }
}

fn decrement(refr: Arc<Mutex<i32>>) {
    loop {
        let mut num = refr.lock().unwrap();
        *num -= 1;
        println!("Decrementing: {}", num);
        thread::sleep(Duration::from_millis(100));
    }
}

fn main() {
    if gtk::init().is_err() {
        println!("Failed to initialize GTK.");
        return;
    }

    let counter = Arc::new(Mutex::new(0));

    let window = Window::new(WindowType::Toplevel);
    window.set_title("Romanetskiy | Thread Priority Control");
```

```

window.set_default_size(300, 200);

let vbox = Box::new(Orientation::Vertical, 0);

let btn_inc_min = Button::with_label("Increment Min");
let btn_inc_max = Button::with_label("Increment Max");
let btn_dec_min = Button::with_label("Decrement Min");
let btn_dec_max = Button::with_label("Decrement Max");

vbox.pack_start(&btn_inc_min, true, true, 0);
vbox.pack_start(&btn_inc_max, true, true, 0);
vbox.pack_start(&btn_dec_min, true, true, 0);
vbox.pack_start(&btn_dec_max, true, true, 0);

window.add(&vbox);

let counter_clone_inc = Arc::clone(&counter);
let inc_thread = thread::spawn(move || {
    increment(counter_clone_inc);
});

let counter_clone_dec = Arc::clone(&counter);
let dec_thread = thread::spawn(move || {
    decrement(counter_clone_dec);
});

let inc_handle = inc_thread.as_raw_handle() as *mut winapi_c_void;
let dec_handle = dec_thread.as_raw_handle() as *mut winapi_c_void;

btn_inc_min.connect_clicked(move |_| {
    set_thread_priority(inc_handle, *PRIORITY_MIN).expect("Failed to set
thread priority");
});

btn_inc_max.connect_clicked(move |_| {
    set_thread_priority(inc_handle, *PRIORITY_MAX).expect("Failed to set
thread priority");
});

btn_dec_min.connect_clicked(move |_| {
    set_thread_priority(dec_handle, *PRIORITY_MIN).expect("Failed to set
thread priority");
});

btn_dec_max.connect_clicked(move |_| {
    set_thread_priority(dec_handle, *PRIORITY_MAX).expect("Failed to set
thread priority");
});

window.show_all();
window.connect_delete_event(|_, _| {

```

```
        gtk::main_quit();  
        Inhibit(false)  
    });  
  
    gtk::main();  
}
```