CS 551 Spring 2025 Quiz 4 KEY

February 25, 2025

Name:

Email:

Instructions: Put your name and email in the appropriate places. Answer, to the best of your ability the question(s) below. Additionally, sign your name below the academic honesty statement. If you deviate from these instructions in any way, you will receive a zero on the quiz.

Unless otherwise specified, you can assume that all necessary imports have been made and there are no deliberate typos in function or type names.

Consider the following stucts, traits, enums, and functions:

```
pub enum FizzBuzz<T> {
    Fizz,
    Buzz,
    Both,
    Neither(T),
}

impl<T: Integer> FizzBuzz<T> {
    /// creates a new ‘FizzBuzz<T>’
    fn new(number: T) -> Self {
        // assume the correct code is implemented here
    }
}

/// Create threads and use them to compute fizz buzz values
pub(crate) fn compute_fizz_buzz_via_channels(
    in_file: &str,
    num_threads: NonZeroU8,
    use_mutex: bool,
) {
    let file = File::open(in_file).unwrap();
    let reader = BufReader::new(file);

    let (tx_fizz_buzzer, rx_fizz_buzzer) = crossbeam_channel::unbounded();
    let (tx_printer, rx_printer) = crossbeam_channel::unbounded();

    let mutex = if use_mutex {
        Some(Arc::new(Mutex::new(HashMap::new())))
    } else {
        None
    };

    let mut handles = create_fizz_buzzer_threads(num_threads, rx_fizz_buzzer, tx_printer, mute

    handles.push(create_printer_thread(rx_printer));

    thread::spawn(move || {
        send_numbers(reader, tx_fizz_buzzer);
    });

    for handle in handles {
        handle.join().unwrap();
    }
}
```

Will the following program compile:

```
// assume`create_fizz_buzzer_thread` is properly implemented
fn create_fizz_buzzer_thread(...)
// assume `create_fizz_buzzer_threads` is properly implemented
fn create_fizz_buzzer_threads(...)

fn create_fizz_buzzer_thread_mutex(
    thread_id: i32,
    rx_fizz_buzzer: Receiver<i128>,
    tx_printer: Sender<String>,
    prev_results: Arc<Mutex<HashMap<i128, String>>>,
) -> JoinHandle<()> {
    thread::spawn(move || {
        while let Ok(number) = rx_fizz_buzzer.recv() {
            let fizz_buzz = {
                prev_results
                    .entry(number)
                    .or_insert(format!("{thread_id}: {}", FizzBuzz::new(number)))
                    .to_string()
            };

            tx_printer
                .send(fizz_buzz.to_string())
                .unwrap();
        }
    })
}

fn main() {
    compute_fizz_buzz_via_channels("in.txt", 10.into(), true);
}
```

(circle your answer)

No

Reason: We never lock `prev_results` mutex.

Will the following program compile:

```
 1 // assume‘create_fizz_buzzer_thread_mutex‘ is properly implemented
 2 fn create_fizz_buzzer_thread_mutex(...)
 3 // assume ‘create_fizz_buzzer_thread‘ is properly implemented
 4 fn create_fizz_buzzer_thread(..)
 5
 6 fn create_fizz_buzzer_threads(
 7     num_threads: NonZeroU8,
 8     rx_fizz_buzzer: Receiver<i128>,
 9     tx_printer: Sender<String>,
10     mutex: Option<Arc<Mutex<HashMap<i128, String>>>>,
11 ) -> Vec<JoinHandle<()>> {
12     (0..num_threads.into())
13         .map(|thread_id| {
14             if let Some(m) = &mutex {
15                 create_fizz_buzzer_thread_mutex(
16                     thread_id.into(),
17                     rx_fizz_buzzer,
18                     tx_printer,
19                     m,
20                 )
21             } else {
22                 create_fizz_buzzer_thread(
23                     thread_id.into(),
24                     rx_fizz_buzzer,
25                     tx_printer,
26                 )
27             }
28         })
29         .collect::<Vec<_>>()
30 }
31
32 fn main() {
33     compute_fizz_buzz_via_channels("in.txt", 10.into(), true);
34 }
```

(circle your answer)

No

Reason: The loop starting on line 12 takes ownership of the channels and mutex when calling one of the two thread creation functions.

Academic honesty statement: I have done this quiz completely on my own. I have not copied it from, nor have I given answers to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the quiz involved, my grade in the class will be reduced by at least one level (e.g., from A to B) for my offense, and that I will receive a grade of “F” for the course for any additional offense of any kind.