実践情報セキュリティとアルゴリズム 第五回課題

28G23027 川原尚己

課題 1

以下の意味について解説せよ.

- クリティカルセクション
  レースコンディションを引き起こすようなプログラムコードの部分のこと.
- レースコンディション
  複数のプロセスやスレッドが予期せぬ依存関係により不具合が起きるような状態のこと.
- アトミック処理
  それ以上分割不可能な処理のこと．ある処理がアトミックであるならば，その処理の途中状態は他プロセスからアクセスできないかつ，もし処理が失敗した場合は完全ン位処理前の状態に戻るという性質を持つ.
- ミューテックス
  クリティカルセクションを実行可能なプロセス数を制限するような制御方法の一つ.最大一つのプロセスやスレッドのみがアクセス制限を課すことができるような制御方法のこと.

課題 2

Rust 言語のミューテックスは、Pthreads のミューテックスよりも幾つかの点で安全であると言える。それはどのような点で、なぜ安全だと言えるのか解説せよ。

他プロセスやスレッドにアクセス制限を課す動作のことを「ロックする」という．Pthreads のミューテックスでは，ロックすべき箇所をロックしていなかったり，ロックした後にロックを開放することをし忘れたということが起こったとき，実行を防ぐ方法が存在しなかった．Rust では，型システムによって構成されているため，コンパイル時にエラーを発見することができる.

課題 3

Rust のスピンロック実装で、それぞれ何を行っているか解説せよ.

以下に，解説に必要な型や関数などの説明を述べる.

- SpinLockGuard：ロックの開放や，ロック中に保護対象データを操作するため
- AtomicBool：スレッド間で安全に共有できる Bool 型．現在ロック中かどうかを格納するため.

- compare_exchange_weak：現在ロック中でないなら，ロック中であると書き換え，新たな処理を開始する．
- Sync：ある変数が複数のスレッドから同時に参照されても安全に処理できるよう，処理を同期するため．
- Send：ある変数の値を別のスレッドへ安全に move するため

“while self.lock.load(Ordering::Relaxed) {}”により，常にロック状況を監視する．もしロックがなされていない(“self.lock==false”)ならば，” compare_exchange_weak”関数によりロック状態がアトミックに true に変更され，新たな処理が実行される．さらに，SpinLock 型はスレッド間で共有可能であるから，ロック状態設定の競合が起きないよう，Sysc トレイト及び Send トレイトにより，適切な処理同期と所有権の move を保証している．
この挙動は，ロック状態の有無を常に監視し，ロックが解かれていれば処理を実行するというスピンロックの挙動と一致している．