05-wcr

目标：实现 wc 命令

实现

- 参数解析

```rust
if [words, bytes, chars, lines].iter().all(|v| v == &false) {
    lines = true;
    words = true;
    bytes = true;
}
```

- 如果都为 false，则设置 lines、words、bytes 为 true

其他方法：

- `Iterator::any` will return true if even one evaluation of the closure for an item returns true.
- `Iterator::filter` will find all elements for which the predicate is true.
- `Iterator::map` will apply a closure to each element and return a `std::iter::Map` with the transformed elements.
- `Iterator::find` will return **the first element** of an iterator that satisfies the predicate as Some(value) or None if all elements evaluate to false.
- `Iterator::position` will return the index of the first element that satisfies the predicate as Some(value) or None if all elements evaluate to false.
- `Iterator::cmp`, `Iterator::min_by`, and `Iterator::max_by` have predicates that accept pairs of items for comparison or to find the minimum and maximum.

- 计数

Rust

```rust
#[derive(Debug, PartialEq)]
pub struct FileInfo {
    num_lines: usize,
    num_words: usize,
    num_bytes: usize,
    num_chars: usize,
}

pub fn count(mut file: impl BufRead) -> MyResult<FileInfo> {
    let mut num_lines = 0;
    let mut num_words = 0;
    let mut num_bytes = 0;
    let mut num_chars = 0;
    let mut line = String::new();

    loop {
        let line_bytes = file.read_line(&mut line)?;
        if line_bytes == 0 {
            break;
        }
        num_bytes += line_bytes;
        num_lines += 1;
        num_words += line.split_whitespace().count();
        num_chars += line.chars().count();
        line.clear();
    }

    Ok(FileInfo {
        num_lines,
        num_words,
        num_bytes,
        num_chars,
    })
}
```

- Use the `str::split_whitespace` method to break the string on whitespace and use `Iterator::count` to find the number of words.

- 测试 count 函数

Rust

```rust
 1  #[cfg(test)]
 2  mod tests {
 3      use super::{count, FileInfo};
 4      use std::io::Cursor;
 5
 6      #[test]
 7      fn test_count() {
 8          let text = "I don't want the world. I just want your half.\r\n";
 9          let info = count(Cursor::new(text));
10          assert!(info.is_ok());
11          let expected = FileInfo {
12              num_lines: 1,
13              num_words: 10,
14              num_chars: 48,
15              num_bytes: 48,
16          };
17          assert_eq!(info.unwrap(), expected);
18      }
19  }
```

- 第1行指定编译器只在测试的时候编译下面的模块
- 第2行定义一个模块
- 第3行引用 count 函数和 FileInfo 结构体
- 第17行的比较，需要结构体 derive `PartialEq`