14-lsr

目标：实现 ls 命令

实现

- 查找文件

Rust

```rust
fn find_files(paths: &[String], show_hidden: bool) -> MyResult<Vec<PathBuf>> {
    let mut results = vec![];
    for name in paths {
        match fs::metadata(name) {
            Err(e) => eprintln!("{}: {}", name, e),
            Ok(meta) => {
                if meta.is_dir() {
                    for entry in fs::read_dir(name)? {
                        let entry = entry?;
                        let path = entry.path();
                        let is_hidden =
                            path.file_name().map_or(false, |file_name| {
                                file_name.to_string_lossy().starts_with('.')
                            });
                        if !is_hidden || show_hidden {
                            results.push(entry.path());
                        }
                    }
                } else {
                    results.push(PathBuf::from(name));
                }
            }
        }
    }
    Ok(results)
}
```

- 文件属性：

- You can find much of the data you need to fill in the cells with `PathBuf::metadata`. Here are some pointers to help you fill in the various columns:
  - `metadata::is_dir` returns a Boolean for whether or not the entry is a directory.
  - `metadata::mode` will return a u32 representing the permissions for the entry. In the next section, I will explain how to format this information into a display string.
  - You can find the number of links using `metadata::nlink`.
  - For the user and group owners, add use `std::os::unix::fs::MetadataExt` so that you can call `metadata::uid` to get the user ID of the owner and `metadata::gid` to get the group ID. Both the user and group IDs are integer values that must be converted into actual user and group names. For this, I recommend you look at the users crate that contains the functions get_user_by_uid and get_group_by_gid.
  - Use `metadata::len` to get the size of a file or directory.
  - Displaying the file's `metadata::modified` time is tricky. This method returns a `std::time::SystemTime` struct, and I recommend that you use `chrono::DateTime::format` to format the date using strftime syntax, a format that will likely be familiar to C and Perl programmers.
  - Use `Path::display` for the file or directory name.