← Project review - APG1 Bootcamp. Day02

🗎 Type of project		Individual
🕐 Duration		30 min
☑ Passed Peer Reviews		1/3

ssh://git@repos-ssh.21-school.ru:2289/students/Go_Day02.ID_376221/kthucydi_stude...

Copy link   Open

Student ⌃

kthucydi@student.21-school.ru   level 10

About ⌃

Introduction

The methodology of School 21 makes sense only if peer-to-peer reviews are done seriously. Please read all guidelines carefully before starting the review.
- Please, stay courteous, polite, respectful and constructive in all communications during this review.

- Highlight possible malfunctions of the work done by the person and take the time to discuss and debate it. The pedagogy is valid only and only if peer-evaluation is conducted seriously.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and the scope of features. Please, stay open-minded to the vision of the other.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.

Guidelines

- Evaluate only the files that are in src folder on the GIT repository of the student or group.
- Ensure to start reviewing a group project only when the team is present in full.
- Use special flags in the checklist to report, for example, an "empty work" if repository does not contain the work of the student (or group) in the src folder of the develop branch, or "cheat" in case of cheating or if the student (or group) are unable to explain their work at any time during review as well as if one of the points below is not met. However, except for cheating cases, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them at the next review.
- Doublecheck that the GIT repository is the one corresponding to the student or the group.
- Meticulously check that nothing malicious has been used to mislead you.
- In controversial cases, remember that the checklist determines only the general order of the check. The final decision on project evaluation remains with the reviewer.

MAIN PART ⌃

Exercise 00 - Finding Things

Check that application finds recursively and prints files, directories and symlinks if given a directory which has those
Check that application finds and prints only files if given a '-f' option
Check that application finds and prints only files only files with a given extension if given '-f' and '-ext' options
Check that '-ext' option is only applied when '-f' is specified
Check that application finds and prints only directories if given a '-d' option
Check that application finds and prints only symlinks if given a '-sl' option
Check that application successfully resolves symlinks and prints both link and original paths for them, separated by '->'
Check that application prints both link and '[broken]' for broken symlinks
Check that application works correctly with two or three options from ['-f', '-d', '-sl'] specified simultaneously
Check that application prints a meaningful error message when no directory path is given or when it doesn't exist

No		✓ Yes

Exercise 01 - Counting Things

Check that application calculates a number of lines in one file correctly
Check that application calculates a number of words in one file correctly
Check that application calculates a number of characters in one file correctly
Check that application calculates all three parameters for multiple files simultaneously
Check that if multiple files are specified, they are processed concurrently in goroutines
Check that application's output is separated by tab symbol

No	✓ Yes

Exercise 02 - Running Things

Check that application creates a proper tar.gz archive with a file inside
Check that application correctly determines original file's mtime
Check that if multiple files are specified, they are processed concurrently in goroutines

No	✓ Yes

Exercise 03 - Archiving Things

Check that application creates a proper tar.gz archive with a file inside
Check that application correctly determines original file's mtime
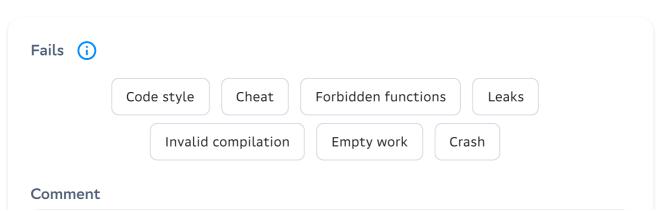Check that if multiple files are specified, they are processed concurrently in goroutines

No	✓ Yes

Feedback ⌃

Fails ⓘ

Code style   Cheat   Forbidden functions   Leaks

Invalid compilation   Empty work   Crash

Comment

Leave a comment...

✓ Review