9. File system operations

hw9.zip
23 Dec 2023, 01:43 PM

Part 1

A. Descriptions

In this assignment, you will learn how to read and write files in the C programming language.

Here are some hints for your reference:

fopen(): `FILE * fopen ( const char * filename, const char * mode );`

fread(): `size_t fread ( void * ptr, size_t size, size_t count, FILE * stream );`

fseek(): `int fseek ( FILE * stream, long int offset, int origin );`

fwrite(): `size_t fwrite ( const void * ptr, size_t size, size_t count, FILE * stream );`

fclose(): `int fclose ( FILE * stream );`

B. Implementation

You need to complete the following tasks by modify the example code hw9_part1.c

- Open the source file we provide (source.txt) for reading.
- Create an output file names “destination.txt” for writing.
- Read the contents of the source file into the buffer.
- Reverse the contents from source file and write to the destination file.
- Close the files.

Note:

- Use `gcc -o hw9_part1 hw9_part1.c` to compile your code.
- The output filename must be `destination.txt`
- You can check your answer by using `diff ans.txt destination.txt`

Part 2 InspectDirectory

A. Descriptions

In this assignment, you will learn how to read information about files in a directory and prints details such as name, size, type (regular file or directory), and modification time.

Here are some hints for your reference:

opendir(): `DIR *opendir(const char *dirname);`

readdir(): `struct dirent *readdir(DIR *dirp);`

closedir(): `int closedir(DIR *dirp);`

stat(): `int stat(const char *path, struct stat *buf);`

S_ISDIR(*m*): Test for a directory.

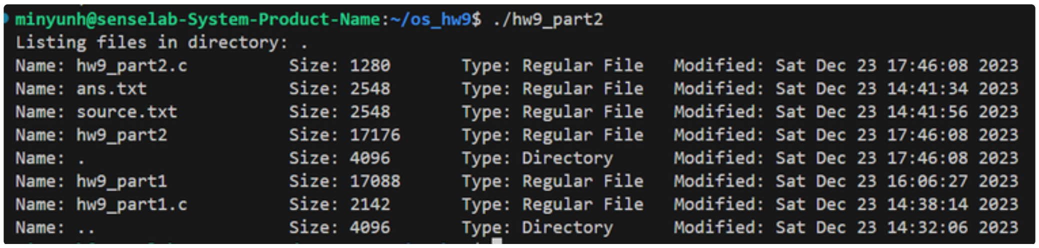
S_ISREG(*m*): Test for a regular file.

ctime(): `char *ctime(const time_t *clock);`

B. Implementation

You need to complete the following tasks by modify the example code hw9_part2.c

- Open the current working directory for reading.
- Read entries (files and directories) in the directory.
- For each entry, determine and print the following information:
    - Name
    - Size (in bytes)
    - Type (Regular File or Directory)
    - Modification time



```
minyunh@senselab-System-Product-Name:~/os_hw9$ ./hw9_part2
Listing files in directory: .
Name: hw9_part2.c          Size: 1280       Type: Regular File   Modified: Sat Dec 23 17:46:08 2023
Name: ans.txt              Size: 2548       Type: Regular File   Modified: Sat Dec 23 14:41:34 2023
Name: source.txt           Size: 2548       Type: Regular File   Modified: Sat Dec 23 14:41:56 2023
Name: hw9_part2            Size: 17176      Type: Regular File   Modified: Sat Dec 23 17:46:08 2023
Name: .                    Size: 4096       Type: Directory      Modified: Sat Dec 23 17:46:08 2023
Name: hw9_part1            Size: 17088      Type: Regular File   Modified: Sat Dec 23 16:06:27 2023
Name: hw9_part1.c          Size: 2142       Type: Regular File   Modified: Sat Dec 23 14:38:14 2023
Name: ..                   Size: 4096       Type: Directory      Modified: Sat Dec 23 14:32:06 2023
```

Note:

- Use `gcc -o hw9_part2 hw9_part2.c` to compile your code.
- To run the program, use the command `./hw9_part2 .`

Submission

Please submit a zip file to E3 which contains your source code and report.

Source Code (80%):

- hw9_part1.c (50%)
- hw9_part2.c (30%)

The program must implemented using C.

Make sure your code can be compiled on **Ubuntu 22.04 AMD64**.

Make sure your outputs are correct.

Report (20%): (English or Chinese)

- Part I (10%)
    - Explain how you implement your code clearly with screenshots of your functions.
- Part II (10%)

    - Explain how you implement your code clearly.
    - A screenshot of your test results.
- **<stduent_id>.zip**

  |- **<student_id>/**

  |- **hw9_part1.c**
  |- **hw9_part2.c**
  |- **destination.txt**

  |- **hw9.pdf**
- For questions, please contact TA Min-Yun Hsieh <minyunh.cs10@nycu.edu.tw>