

Assignment2

Team18

112062519 廖思愷

112062636 游竣量

111065547 游述宇

- Describe the implementation:

1. 標頭檔

首先，引入必要的標頭文件。

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/stat.h>
```

2. 定義緩衝區大小

接著，定義一個宏 `BUFFER_SIZE` 來設定讀寫緩衝區的大小，其值為 10。

```
#define BUFFER_SIZE 10
```

3. 自訂 `show_error` function

用於在發生錯誤時顯示錯誤訊息並終止程式。`show_error` 接受一個錯誤訊息的字串作為參數。

```
int show_error(char error_msg[]) {
    printf("%s\n", error_msg);
    exit(-1);
}
```

4. main function

main function 接受兩個 command line 參數，第一個是 source file 的路徑，第二個是 target file 的路徑，其中 `argc` 表示從 command line 傳進來的參數數量，`argv[]` 是一個字串陣列，用於儲存 command line 傳進來的參數。

```
int main(int argc, char* argv[]) {
    char *source_file = argv[1];           //Command Line 1: Source file path
    char *target_file = argv[2];           //Command Line 2: Target file path
    char my_buffer[BUFFER_SIZE] = "";
    int fd1;
    int fd2;
```

5. 打開 files

在程式碼中，首先嘗試打開 source file 和 target file。

```
if((fd1 = open(source_file, O_RDONLY)) < 0) {
    show_error("open error");
}

if((fd2 = open(target_file, O_CREAT | O_TRUNC | O_WRONLY, 0777)) < 0) {
    show_error("create error");
}
```

`O_RDONLY` 表示以唯讀模式打開 source file。

`O_CREAT` 表示如果 target file 不存在，就創建一個。

`O_TRUNC` 表示如果 target file 已存在，則清除其內容。

`O_WRONLY` 表示以唯寫模式打開 target file。

`0777` 是 target file 的權限設定，此處設定為最大權限。

6. 複製文件內容

使用 `read` 將 source file 的內容讀進 `my_buffer`，然後用 `write` 將其寫入 target file 中，在 while-loop 內直到 source file 被完全複製。

```
ssize_t read_size;
while((read_size = read(fd1, my_buffer, BUFFER_SIZE - 1)) > 0) {
    if(write(fd2, my_buffer, read_size) != read_size) {
        show_error("write error");
    }
}
```

7. 複製文件權限

使用 `stat` 來獲取 source file 的權限設定，然後使用 `fchmod` 將相同的權限設定用在 target file 上。

```
struct stat stat_buf;
stat(source_file, &stat_buf);
// printf("%o\n", stat_buf.st_mode & 0777);
fchmod(fd2, stat_buf.st_mode);
```

目的是確保 target file 具有與 source file 相同的權限。

8. 關閉文件

最後，關閉 source file 和 target file。

```
close(fd1);
close(fd2);
```

● Case1: target file 不存在

```
root@:/home/team18/Advanced-UNIX-Programming_Student/assignment2 # ls
Makefile      assignment2    assignment2.c  assignment2.o  source.txt
root@:/home/team18/Advanced-UNIX-Programming_Student/assignment2 # ./assignment2 source.txt target.txt
root@:/home/team18/Advanced-UNIX-Programming_Student/assignment2 # diff source.txt target.txt
root@:/home/team18/Advanced-UNIX-Programming_Student/assignment2 # cat source.txt
fbbfddnfdnn nfdnfdn
fdndfnd
fndf  nfdnfdn  hhdhfdhdfh
fdhdfhdfhdfhksg;lkgstlrr;ln
dbdf  grl;gre;l reegerroot@:/home/team18/Advanced-UNIX-Programming_Student/assignment2 # cat target.txt
fbbfddnfdnn nfdnfdn
fdndfnd
fndf  nfdnfdn  hhdhfdhdfh
fdhdfhdfhdfhksg;lkgstlrr;ln
dbdf  grl;gre;l reegerroot@:/home/team18/Advanced-UNIX-Programming_Student/assignment2 #
```

解釋：

1. 使用 ls 可以看到當前目錄中只有 source file。
2. 使用我們的程式來創建 target file 並複製 source file 的內容給它。
3. 下 diff 指令，結果沒有輸出，表示兩個 file 現在是完全相同的。
4. 使用 cat 顯示兩個文件實際的內容。

● Case2: target file 存在

```
team18@:~/Advanced-UNIX-Programming_Student/assignment2 $ diff source.txt target.txt
1c1
< llldnfdnn nfdnfdn
---
> ddnfdnn nfdnfdn
team18@:~/Advanced-UNIX-Programming_Student/assignment2 $ sudo ./assignment2 source.txt target.txt
team18@:~/Advanced-UNIX-Programming_Student/assignment2 $ diff source.txt target.txt
team18@:~/Advanced-UNIX-Programming_Student/assignment2 $
```

解釋：

1. 首先試著對兩個不同的文件下 diff 指令，會輸出兩個 file 不同之處。
2. 使用我們的程式來複製 source file 內容給 target file。
3. 再下 diff 指令，結果沒有輸出，表示兩個 file 現在是完全相同的。程式成功完成了複製工作！