109590004 呂育瑋 作業系統 HW#1

系統環境：VMware, Linux, Ubuntu 18.04

* Chap.2 Programming problem: 2.15

程式碼：Source code/filecopy.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#define BUFF\_MAX 4086

#define FILE\_IN "Source"

#define FILE\_OUT "Destination"

int openFile(char value[], char type[]);

int main(int argc, char \*argv[]) {

    if (argc != 3) {

        printf("[Error] Use: ./fileCopy.o <source file name> <destination file name>\n");

        return 1;

    }

    int fd\_source;

    int fd\_dest;

    char buff[BUFF\_MAX] = { 0 };

    fd\_source = openFile(argv[1], FILE\_IN);

    read(fd\_source, buff, BUFF\_MAX);

    close(fd\_source);

    fd\_dest = openFile(argv[2], FILE\_OUT);

    write(fd\_dest, buff, strlen(buff));

    close(fd\_dest);

    printf("[Success] Copy file content.\n");

    return 0;

}

// Open file with error check

int openFile(char value[], char type[]) {

    int fd = -1;

    if (strcmp(type, FILE\_IN) == 0) {

        fd = open(value, O\_RDONLY);

    } else if (strcmp(type, FILE\_OUT) == 0) {

        fd = open(value, O\_TRUNC | O\_WRONLY);

    }

    if (fd == -1) {

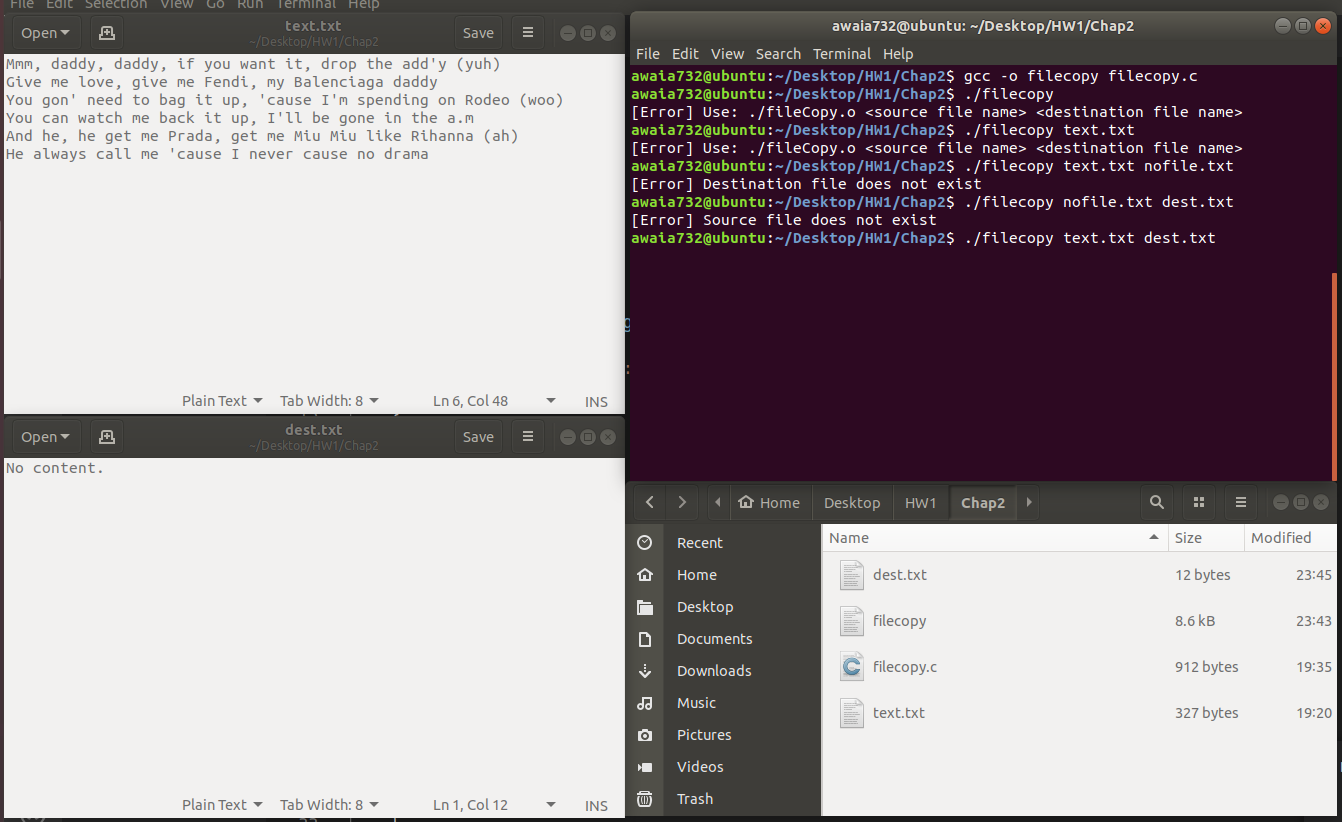
        printf("[Error] %s file does not exist\n", type);

        exit(1);

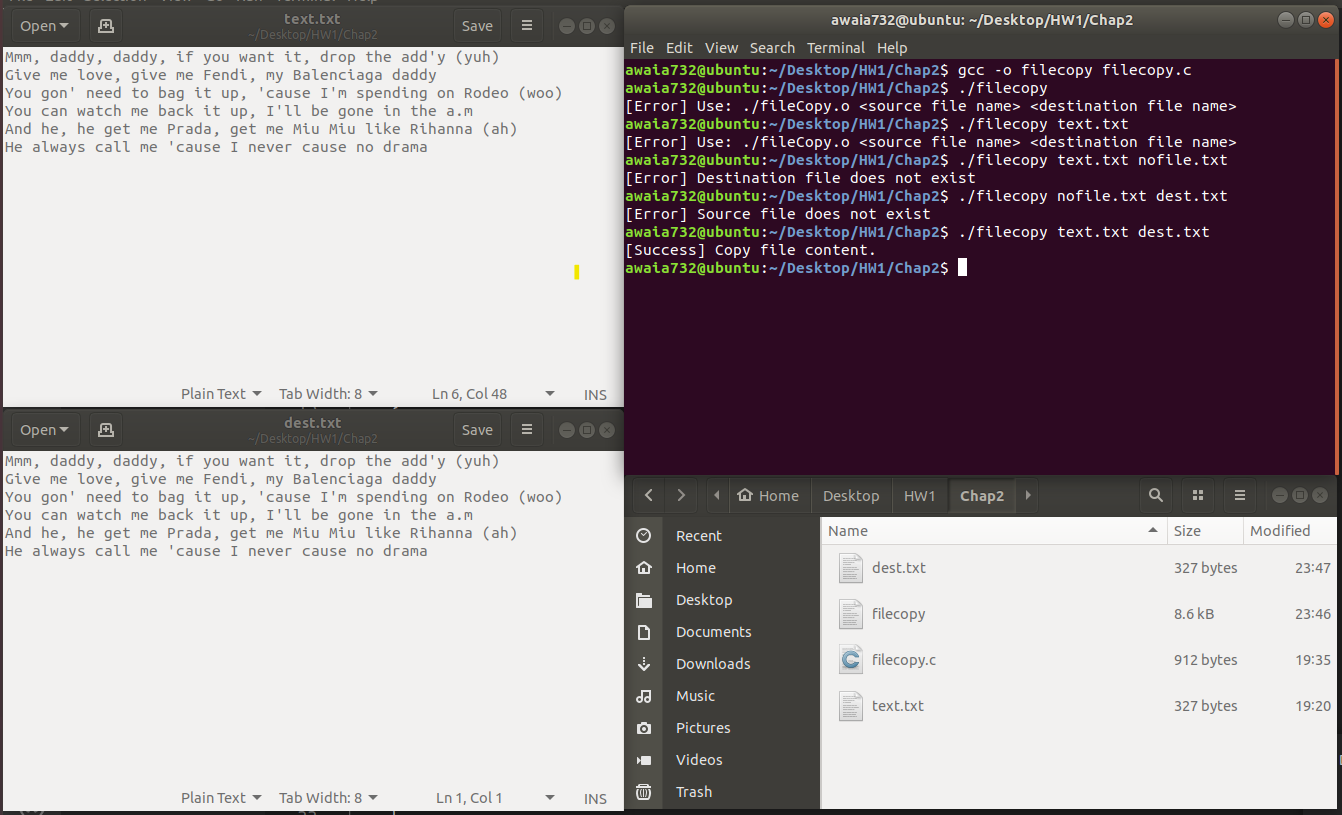
    }

}

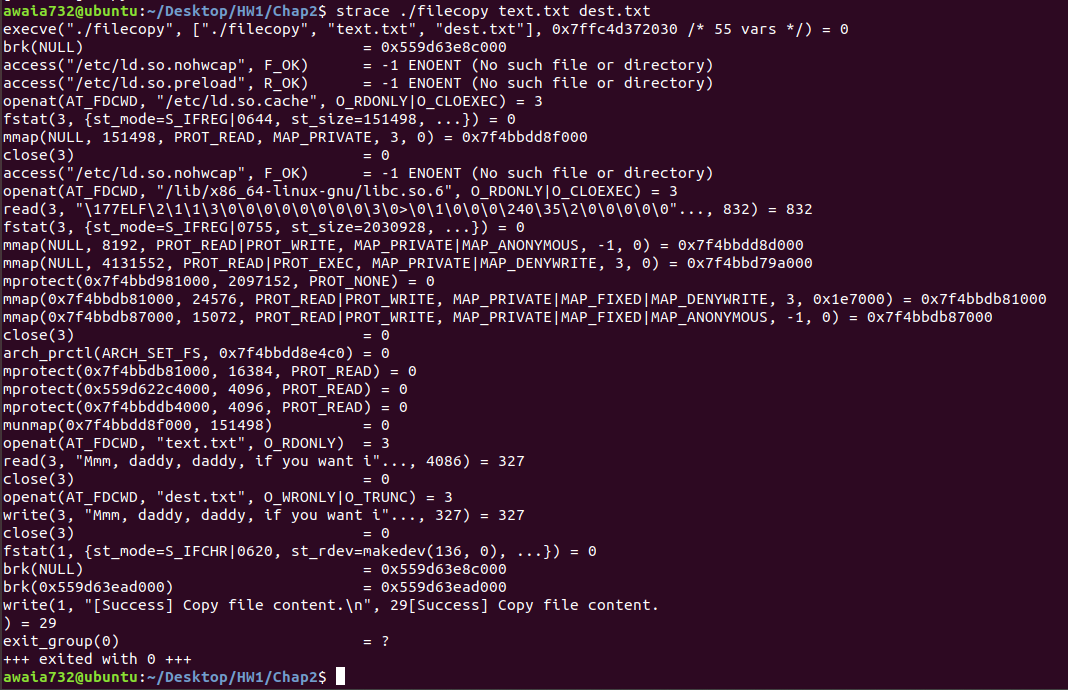
利用System call實作檔案內容複製功能，將程式編譯並執行，在呼叫執行檔時需加入兩個參數，前者為複製來源檔；後者複製目標檔，如果缺少執行參數或輸入的檔案並未被找到，則會回報對應的錯誤內容，如下圖。



如果參數的檔案皆存在，則開始進行資料複製，使用到的System call函式有：open, close, read, write



因為系統是Linux，要查看執行程式用到那些System call可以使用strace指令，能看到除了自己寫的System call function以外，還呼叫了許多由系統自行執行的System call。



* Chap.3 Programming problem: 3.14

程式碼：Source code/counting.c

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <unistd.h>

#define BUFF\_MAX 4086

int main(int argc, char \*argv[]) {

    if (argc != 2) {

        printf("[Error] Use: ./counting <positive number>\n");

        return 1;

    }

    int num = atoi(argv[1]);

    if (num < 1) {

        printf("[Error] Number must be positive.\n");

        return 1;

    }

    pid\_t pid = fork();

    if (pid > 0) {

        printf("[Parent] Waiting child process finish ...\n");

        wait(NULL);

        printf("[Parent] Process is done.\n");

    } else if (pid == 0) {

        printf("[Child] Process working ...\n");

        printf("[Child] Number sequence: %d", num);

        while (num != 1) {

            if (num % 2 == 0) {

                num /= 2;

            } else {

                num = 3 \* num + 1;

            }

            printf(", %d", num);

        }

        printf("\n[Child] Process is done.\n");

    } else {

        perror("fork()");

        exit(-1);

    }

    return 0;

}

將程式編譯執行，並加入一個參數為要求的數列初始值，如果並沒有加入參數，或是參數不為正整數，都匯回報錯誤訊息，如果通過參數檢查，則利用fork()建立父程序與子程序，父程序等待子程序完成才繼續執行程式使用wait()，子程序負責計算數列並輸出。

