

Додаток 2 до лабораторної роботи «Програмне відновлення вилучених файлів» з навчальної дисципліни вільного вибору «**Основи інформаційної та кібербезпеки**»

Завдання Б. Напишіть програму, яка відновлює JPEG зображення з карти пам'яті.

Передісторія

Уявимо себе в ролі хакера, який отримав доступ до карти пам'яті, яку необережно використовували. Карта пам'яті була використана фотоапаратом на яку були зроблені секретні фотографії, а потім вона була "обачливо" очищена (файли видалені). Ми отримали доступ до цієї карти пам'яті та зробили її цифровий відбиток замість вас. Цей відбиток являється лише послідовністю нулів та одиниць (байтів) що записані в один файл для зручності (RAW файл), він повністю відповідає стану цифрового носія після видалення файлів.

Теорія

Незважаючи на те, що JPEG є більш складними, ніж BMP, JPEG мають "підписи", шаблони байтів, які можуть відрізнити їх від інших форматів файлів. Зокрема, перші три байти JPEG 0xff 0xd8 0xff від першого байта до третього байта, зліва направо. Тим часом четвертий байт - це 0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee або 0xef. Іншими словами, перші чотири біти четвертого байта - це 1110.

Швидше за все, якщо ви знайдете цей шаблон із чотирьох байт на носіях, які, як відомо, зберігають фотографії (наприклад, на моїй карті пам'яті), вони визначають початок JPEG. Чесно кажучи, ви можете випадково зустріти ці закономірності на якомусь диску, тому відновлення даних не є точною наукою.

На щастя, цифрові камери, як правило, постійно зберігають фотографії на картах пам'яті, завдяки чому кожна фотографія зберігається відразу після зробленої раніше фотографії. Відповідно, початок JPEG зазвичай позначає кінець іншого. Однак цифрові камери часто ініціалізують картки файловою системою FAT, чий "розмір блоку" становить 512 байт (B). Наслідком є те, що ці камери записують на ці картки лише в одиницях 512 B. Фотографія розміром 1 МБ (тобто 1048 576 B), таким чином, займає $1048576 \div 512 = 2048$ "блоків" на карті пам'яті. Але так само і фото, яке, скажімо, на один байт менше (тобто, 1048 575 B)! Даремно витрачений простір на диску називається «вільним місцем». Криміналісти часто розглядають вільний простір, щоб знайти залишки підозрілих даних.

Наслідком усіх цих деталей є те, що ви, можете написати програму, яка переглядає копію моєї картки пам'яті, шукаючи підписи JPEG. Кожного разу, коли ви знайдете підпис, ви можете відкрити новий файл для запису і почати заповнювати цей файл байтами з моєї картки пам'яті, закриваючи цей файл лише після того, як ви зустрінете інший підпис. Більше того, замість того, щоб читати байти моєї картки пам'яті по черзі, ви можете для ефективності прочитати 512 з них за раз. Завдяки FAT

ви можете довіряти, що підписи JPEG будуть "вирівняні за блоками". Тобто вам потрібно шукати ці підписи лише у перших чотирьох байтах блоку.

Файли JPEG можуть охоплювати суміжні блоки. Проте останній байт JPEG може не потрапити в самий кінець блоку. Згадайте про можливість вільного місця. Оскільки ця карта пам'яті була абсолютно новою, коли починали знімати фотографії, що спрощує нам завдання і означає що будь-яке вільне місце буде заповнене 0. Це нормально, якщо ці кінцеві нулі потрапляють у JPEG, які ви відновлюєте.

Практика

Отже, ми створили «криміналістичне зображення» картки, зберігаючи її вміст байт за байтом, у файлі який називається "**card.raw**". Ми зобразили лише перші кілька мегабайт карти пам'яті. Але в підсумку ви повинні виявити 50 JPEG зображень.

Щоб завантажити card.raw перейдіть за посиланням <http://cdn.cs50.net/2020/fall/psets/4/recover/recover.zip>

Завдання

Створити програму, яка відновлює JPEG з криміналістичного зображення.

Програма повинна прийняти рівно один аргумент командного рядка, назва криміналістичного зображення, з якого можна відновити JPEG.

Якщо ваша програма не виконується рівно з одним аргументом командного рядка, вона повинна нагадувати користувачеві про правильне використання, та повинна повертати 1.

Якщо криміналістичне зображення не можна відкрити для читання, ваша програма повинна повідомити про це користувача, та повинна повернути 1.

Кожен створений файл повинен мати назву ###.jpg, де ### - трицифрове десяткове число, починаючи з 000 для першого зображення.

Після запуску ваша програма повинна відновити кожен із файлів JPEG із card.raw, зберігаючи кожен як окремий файл у вашому поточному робочому каталозі. Ваша програма повинна нумерувати файли, які вона видає, називаючи кожен ###.jpg, де ### - трицифрове десяткове число від 000 і вище. Вам не потрібно намагатись відновити оригінальні назви JPEG. Щоб перевірити, чи правильні JPEG-файли, просто двічі клацніть і подивіться! Якщо кожна фотографія здається цілою, ваша операція, мабуть, була успішною!

Однак є шанси на те, що JPEG, які повертає ваша програма, будуть неправильними. (Якщо ви відкриваєте їх і нічого не бачите, вони, мабуть, неправильні!).

УВАГА!!!

Якщо у Вас достатньо знань і навичок з програмування спробуйте виконати це завдання самостійно.

На наступних сторінках показано реалізацію цього завдання мовою програмування C. Ви можете скористатись готовою програмою і відновити файли із криміналістичного зображення “**card.raw**”.

Встановимо компілятор мови програмування C та напишемо програму.

Надалі, будемо використовувати наступні позначення:

Для тих хто користується ОС Linux – **Linux**:

Для тих хто користується ОС Windows – **Win**:

Усі спільні дії будуть використовувати позначку – **All**:

Виконуємо всі інструкції послідовно, пропускаючи команди не своїх ОС. Початок інструкцій починається маркером відповідної ОС та закінчується початком іншого маркера.

Win:

Переходимо за посиланням та інсталуємо компілятор gcc

<http://www.equation.com/servlet/equation.cmd?fa=fortran>

(наприклад файл gcc-8.3.0-64.exe)

Як встановити компілятор gcc / g ++ для Windows **дивіться файл Додаток 3.**

Після встановлення компілятора gcc завантажуюмо та розпаковуємо архів

<http://cdn.cs50.net/2020/fall/psets/4/recover/recover.zip>

після розпакування в папці recover буде файл recover.c

Відкриваємо його за допомогою будь-якого текстового редактора, можна використати «блокнот» або Notepad++, його вміст повністю замінюємо текстом програми поданим нижче та закриваємо зі збереженням.

Linux:

На робочому столі натискаємо правою клавішею миші та вибираємо пункт «Відкрити в терміналі»

Пишемо наступне

```
sudo apt install build-essential wget unzip
```

Вводимо свій пароль від облікового запису, якщо він існує (символи не будуть зображені, просто вводимо пароль та натискаємо Enter)

```
wget http://cdn.cs50.net/2020/fall/psets/4/recover/recover.zip
unzip recover.zip -d mom_i_m_hacker
rm -f recover.zip
cd mom_i_m_hacker
```

```
konorth@konorth: ~/Стільниця/mom_i_m_hacker
241.16, ...
Встановлення з'єднання з cdn.cs50.net (cdn.cs50.net)|99.86.241.60|:80... з'єднан
о.
HTTP-запит надіслано, очікуємо на відповідь... 301 Moved Permanently
Розміщення: https://cdn.cs50.net/2020/fall/psets/4/recover/recover.zip [перехід]
--2021-02-05 03:53:31-- https://cdn.cs50.net/2020/fall/psets/4/recover/recover.
zip
Встановлення з'єднання з cdn.cs50.net (cdn.cs50.net)|99.86.241.60|:443... з'єдна
но.
HTTP-запит надіслано, очікуємо на відповідь... 200 OK
Довжина: 7878702 (7,5М) [application/zip]
Зберігаємо до «recover.zip»

recover.zip      100%[=====] 7,51M 4,89MB/s   за 1,5s
2021-02-05 03:53:33 (4,89 MB/s) - «recover.zip» збережено [7878702/7878702]

konorth@konorth:~/Стільниця$ unzip recover.zip -d mom_i_m_hacker
Archive: recover.zip
  inflating: mom_i_m_hacker/recover/card.raw
  inflating: mom_i_m_hacker/recover/recover.c
konorth@konorth:~/Стільниця$ rm -f recover.zip
konorth@konorth:~/Стільниця$ cd mom_i_m_hacker
konorth@konorth:~/Стільниця/mom_i_m_hacker$
```

далі

```
touch hello.c
gedit hello.c
```

відкриється текстовий редактор в нього копіюємо наступний текст

```
#include <stdio.h>
int main(void)
{
    printf("Hello, world!\n");
    return 0;
}
```

закриваємо зі збереженням

далі наступні команди

```
gcc hello.c -o hello
./hello
```

ми повинні побачити наступне

```
konorth@konorth: ~/Стільниця/mom_i_m_hacker
konorth@konorth:~/Стільниця/mom_i_m_hacker$ touch hello.c
konorth@konorth:~/Стільниця/mom_i_m_hacker$ gedit hello.c
konorth@konorth:~/Стільниця/mom_i_m_hacker$ gcc hello.c -o hello
konorth@konorth:~/Стільниця/mom_i_m_hacker$ ./hello
Hello, world!
konorth@konorth:~/Стільниця/mom_i_m_hacker$
```

Linux:

```
cd recover
gedit recover.c
```

відкриється текстовий редактор, його вміст повністю замінюємо текстом програми поданим нижче та закриваємо зі збереженням.

All: (текст програми на мові C, див. файл **recover.c** у папці з вказівками до роботи)

```
#include <stdio.h>
#include <stdlib.h>

typedef struct
{
    char bl[512];
}
BUF;

int main(int argc, char *argv[])
{
    // перевіряємо чи передано відбиток для відновлення
    if (argc != 2)
    {
        printf("Usage: ./recover image\n");
        return 1;
    }

    // отримуємо імя відбитку
    char *f = argv[1];
    // відкриваємо файл
    FILE *infile = fopen(f, "rb");
```

```

// якщо не вдалось відкрити файл, завершуємо програму
if (infile == NULL)
{
    fprintf(stderr, "Could not open %s.\n", f);
    return 2;
}

FILE *outfile;
BUF bf;

int a = 0;
// ім'я фотографії
char st[] = "000.jpg";
// читаємо card.raw до кінця блоками по 512 байт
while (fread(&bf, 512, 1, infile))
{
    // якщо бачимо на початку блоку підпис файлу JPEG
    if (bf.bl[0] == (char) 0xff && bf.bl[1] == (char) 0xd8 &&
bf.bl[2] == (char) 0xff)
    {
        // якщо фотографія не перша, завершуємо формувати фотографію
        if (a)
        {
            fclose(outfile);
        }
        // формуємо ім'я файлу
        st[1] = a / 10 + '0';
        st[2] = a++ % 10 + '0';
        // створюємо та відкриваємо новий файл
        outfile = fopen(st, "wb");
    }
    // переписуємо дані з card.raw в поточну фотографію
    if (a)
    {
        fwrite(&bf, 512, 1, outfile);
    }
}
// close
fclose(infile);
fclose(outfile);
return 0;
}

```

Win:

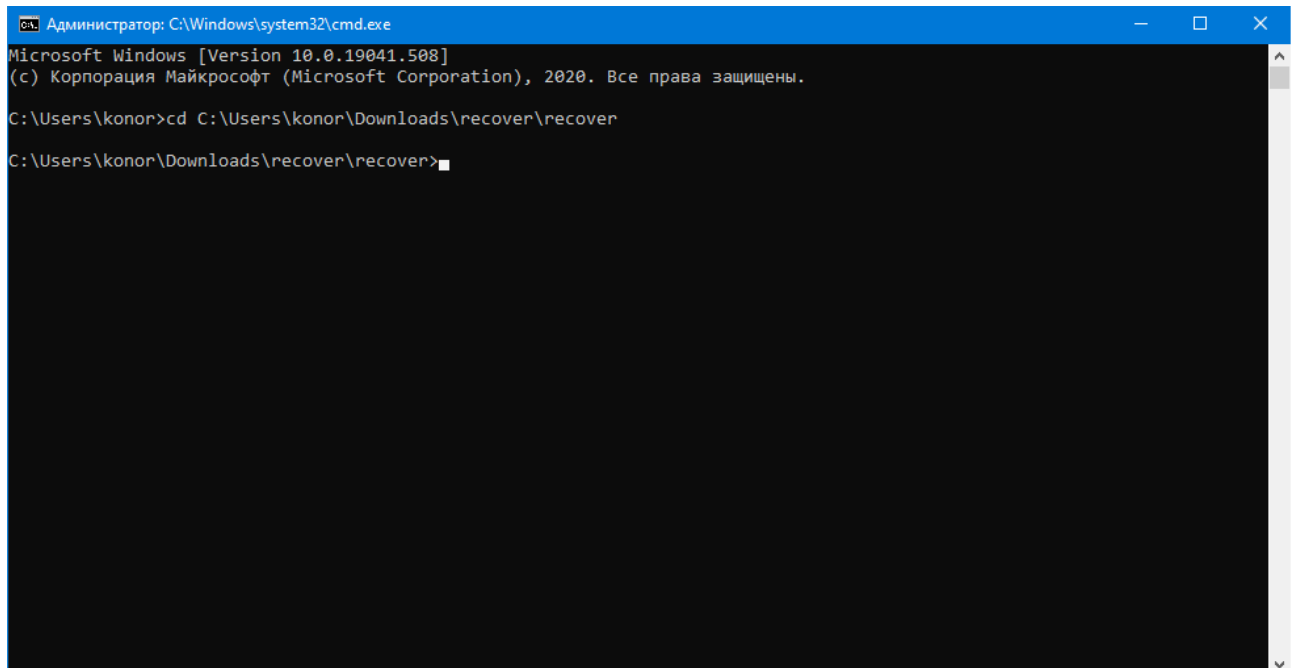
Зберігаємо файл.

Натискаємо комбінацію клавіш WIN + R якщо не можете знайти клавішу WIN скористайтесь пошуком в меню «пуск»

Вводимо cmd та натискаємо кнопку Ok

У відкритому вікні пишемо наступну команду

cd <повний шлях до вашої папки recover>



```
Администратор: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19041.508]
(c) Корпорация Майкрософт (Microsoft Corporation), 2020. Все права защищены.

C:\Users\konor>cd C:\Users\konor\Downloads\recover\recover
C:\Users\konor\Downloads\recover\recover>
```

All:

далі наступні команди

```
gcc recover.c -o recover
```

Linux:

```
./recover card.raw
```

Вітаю! Шукай фотографії на робочому столі в папці `mom_i_m_hacker/recover`

Win:

```
recover.exe card.raw
```

Шукай фотографії в папці `recover`.

У звіті про виконання **завдання Б** цієї лабораторної роботи потрібно надати один із 50 відновлених файлів зображень JPG.

Номер файлу у звіті визначається номером Вашої залікової книжки:

- якщо дві останні цифри номера залікової $xx \leq 49$, то вибираєте файл **0xx.jpg**
- якщо дві останні цифри номера залікової $xx = 50$, то вибираєте файл **000.jpg**
- якщо дві останні цифри номера залікової $xx \geq 51$, проведіть арифметичну дію $yy = xx - 50$ і виберіть файл **0yy.jpg**

Тобто студент із номером залікової книжки 2119124С у звіті представляє файл 024.jpg, а студент із номером залікової книжки 3119086С – файл 036.jpg.