

Open Source!

Если этот файл Вы взяли из архива, пожалуйста, посмотрите в архиве примеры подписанных фотографий, чтобы понимать, что делает этот скрипт. Если этот файл Вы получили вместе с диском, то примеры подписанных фотографий Вы найдёте на диске.

Примечание!

При просмотре фотографий разверните их на весь экран.

Также в конце этого материала есть пример подписанной фотографии.



При использовании этого материала ссылка на автора обязательна!

Perl как Basic, только пишется иначе... ;)

```
#
# ...
# -----|-----
# | Программа (скрипт) подписывает цифровые фотографии, плюс |
# | переименовывает подписанные файлы по заданному шаблону. |
# -----|-----
#

# Open Source!
# Этот скрипт выручит тогда, когда нужно подписать 100-200 (и более) фотографий.
# Например, фотографии пос. Червоный Донец или фотографии
# г.Харькова (например, район Барабашово) и т.п.

# Text: UTF-8. Perl ver. 5.20.3. Program: signature-images.pl. Version: 1.0.
# Author: (C) Demidov S.V. Date: August-September, 2017y.
# Platform: FreeBSD 10.3 (думаю, без проблем будет работать и в Linux).
# -----

# -1-
# | В системе должен быть установлен ImageMagick ver.6.9.2-10 (консольный Gimp)!
# | # cd /usr/ports/graphics/ImageMagick
# | # make install clean
# ---
# | Документация по ImageMagick на http://www.imagemagick.org/Usage/
# ---

#
# Используются утилиты ImageMagick'a:
# -----
# convert - произвести действия над изображением и сохранить в другом файле.
# mogrify - произвести действия над самим изображением.
#
# Примечание.
# -----
# convert и mogrify - принимают одинаковые ключи...
#
# identify - для получения разрешения изображения (ширина x высота).
#

# -2-
# | В системе должен быть установлен ExifTool ver.9.90 (просмотр и редактирование метаданных)!
# | # cd /usr/ports/graphics/p5-Image-ExifTool
# | # make install clean
# ---

# -3-
# | В системе должен быть установлен unix2dos ver.1.3 (добавить CR+LF текстовому файлу)!
# | # cd /usr/ports/converters/unix2dos
# | # make install clean
# ---
# | |
# | | Перевод строки.
# | | Возврат каретки.

# -4-
# | Дополнительно в системе должны быть ещё утилиты:
# |
# | iconv, stat, cp.
# |
# | | |
# | | | Копирование файла.
# | | | Информация о файле.
# | | | Перекодировать текстовый файл из UTF-8 в CP1251.
# |
```

```

# ---

# Для объявления переменных.
use strict;

# В случае возникновения каких-либо проблем нужно остановить работу скрипта.
use warnings FATAL => 'all';

# Сообщаем, что символы в переменных могут быть в utf8.
use utf8;

# Все потоки STDIN, STDOUT, STDERR должны работать с utf8.
use open qw(:std :utf8);

# Модуль Encode.
# Для перекодировки строк из внутреннего представления в utf-8
# (или в другую кодировку).
use Encode;

# ---
# Шрифт: LiberationMono-Bold.ttf
# Лицензия на шрифт: SIL Open Font License v1.1.
# Шрифт должен быть там, где и сам скрипт.
my $namefont = 'signature-images-font.ttf'; # Имя шрифта (для удобства имя изменено).

# Каталоги где находятся изображения.
my $catinimage = '/Temp_1/'; # Каталог для входящих изображений.
my $catoutimage = '/Temp_2/'; # Каталог для исходящих изображений.

# Имя файлов для исходящих изображений (по умолчанию).
my $titlefile = 'Пос. Червоный Донец'; # Название для исходящего файла.
# С какого номера начинать нумерацию.
my $numberfile = 1;
#
# Результат будет: Пос. Червоный Донец (2017-08-21, 17:35:06) - 000000.jpg
#                      Пос. Червоный Донец (2017-08-21, 17:35:06) - 000001.jpg
#                      ...
#                      |
#                      Это будет добавлено...

# Что написать на фотографии (строка 1).
# Берётся из аргументов командной строки.
# Аргументы командной строки в массиве: @ARGV.
#
# Пример: > perl signature-images.pl 'Пос. Червоный Донец, ул. Спортивная, 7.'

# Что написать на фотографии (строка 2).
my $textimage_camera = 'Цифровая камера: Nikon, CoolPIX S4200.';
my $textimage_size = 'Разрешение:';
my $textimage_date = 'Дата';
my $textimage_utc = 'UTC+02:00';
my $textimage_author = '(С) Демидов С.В.';

# Высота нижнего прямоугольника в px.
my $heightrectangle = 128;

# Цвет нижнего прямоугольника.
my $colorrectangle = '#000000';

# Цвет линиям.
my $colorline = '#ffffff';

# Смещение линии (в px) от верха нижнего прямоугольника.
my $offsettop = 7;

# Смещение линии (в px) от низа нижнего прямоугольника.
my $offsetbottom = 7;

# Толщина линий в px.
my $depthline = 5;

# Смещение для первой строки (px), от низа изображения.
my $offsettext_1 = 64;

# Смещение для второй строки (px), от низа изображения.
my $offsettext_2 = 23;

# Размер шрифта.
my $sizefont = 36;

# Яркость по умолчанию.
my $brightness = 20;

```

```

# Контрастность по умолчанию.
my $contrast = 25;

# Пикселей/in или dpi (кому как удобно). DPI = Dots Per Inch.
my $resolution = 300; # По умолчанию на выходе 300dpi.

# Длина имени файла при выводе его на экран.
my $length_filename = 5;
# ---

# ---
# Ввод данных с клавиатуры.

# Очистить консоль.
system('clear');

print "\n";
print '-----' . "\n";
print 'Программа подписывает цифровые фотографии.' . "\n";
print '-----' . "\n";
print "\n";
# ---

# ---
# -1-
#
print 'Имя файлов для исходящих изображений, ' . "\n";
print 'по умолчанию: ' . $titlefile . ' (год-месяц-число, часы:минуты:секунды) - номер.расширение' . "\n";
print '      где дата и время в скобках, номер, расширение, будут подставлены автоматически...' . "\n";
print "\n";
print '      Т.е. результат будет такой (пример): ' . $titlefile . ' (2017-08-18, 11:35:06) - 000001.jpg' . "\n";
print '      ' . $titlefile . ' (2017-08-18, 13:20:11) - 000002.jpg' . "\n";
print '      ...' . "\n";

print "\n";
print 'Имя для файлов: ';

# + Ввод данных с клавиатуры.
my $titlefile_temp = <STDIN>;

# Удалить последний символ строки.
chop($titlefile_temp); #
# |---|
if ($titlefile_temp ne '') # | 1 |
{ # |---|
    $titlefile = $titlefile_temp; #
} #

#
# -2-
#
# Дополнить номер нулями слева.
my $startnumber = sprintf("%06d", $numberfile);
print 'Номер будет начинаться (по умолчанию: ' . $startnumber . '): ';

# + Ввод данных с клавиатуры.
my $numberfile_temp = <STDIN>;

# Удалить последний символ строки.
chop($numberfile_temp); #
# |---|
if ($numberfile_temp ne '') # | 2 |
{ # |---|
    $numberfile = $numberfile_temp; #
} #

#
# -3-
#
print "\n";
print 'Входящий каталог с изображениями (по умолчанию: ' . $catinimage . '): ';

# + Ввод данных с клавиатуры.
my $catinimage_temp = <STDIN>;

# Удалить последний символ строки.
chop($catinimage_temp); #
# |---|
if ($catinimage_temp ne '') # | 3 |
{ # |---|
    $catinimage = $catinimage_temp; #
}

```

```

    }
#

#
# -4-
#
print 'Исходящий каталог с изображениями (по умолчанию: ' . $catoutimage . '): ';

# + Ввод данных с клавиатуры.
my $catoutimage_temp = <STDIN>;

# Удалить последний символ строки.
chop($catoutimage_temp);
#
# |---|
if ($catoutimage_temp ne '') # | 4 |
{
# |---|
    $catoutimage = $catoutimage_temp;
#
}

#
# -5-
#
if (defined $ARGV[0] != 1) # Если неинициализирован @ARGV, значит пусто.
{
    $ARGV[0] = '';
}

my $ARG = $ARGV[0]; # Получить аргумент из командной строки.

if ($ARG eq '')
{
    $ARG = $titlefile . '.';
}

print "\n";
print 'Что написать на фотографии, строка 1 (по умолчанию: ' . $ARG . ')...' . "\n";
print 'Что написать: ';

# + Ввод данных с клавиатуры.
my $str_temp = <STDIN>;

# Удалить последний символ строки.
chop($str_temp);
#
# |---|
if ($str_temp ne '') # | 5 |
{
# |---|
    $ARG = $str_temp;
#
}

#
# -6-
#
print "\n";
print 'Что написать на фотографии, строка 2.' . "\n";
print 'Информация о камере (по умолчанию: ' . $textimage_camera . '): ';

# + Ввод данных с клавиатуры.
my $textimage_camera_temp = <STDIN>;

# Удалить последний символ строки.
chop($textimage_camera_temp);
#
# |---|
if ($textimage_camera_temp ne '') # | 6 |
{
# |---|
    $textimage_camera = $textimage_camera_temp;
#
}

#
# -7-
#
print 'Кто фотографировал (по умолчанию: ' . $textimage_author . '): ';

# + Ввод данных с клавиатуры.
my $textimage_author_temp = <STDIN>;

# Удалить последний символ строки.
chop($textimage_author_temp);
#
# |---|
if ($textimage_author_temp ne '') # | 7 |
{
# |---|
    $textimage_author = $textimage_author_temp;
#
}

```

```

#
# -8-
#
print 'Комментарий к времени (по умолчанию: ' . $textimage_utc . '): ';

# + Ввод данных с клавиатуры.
my $textimage_utc_temp = <STDIN>;

# Удалить последний символ строки.
chop($textimage_utc_temp);

if ($textimage_utc_temp ne '')
{
    $textimage_utc = $textimage_utc_temp;
}

#
# -9-
#
print "\n";
print 'Добавить/уменьшить яркость и контрастность (y/n)? ';

# + Ввод данных с клавиатуры.
my $light = <STDIN>;

# Удалить последний символ строки.
chop($light);

if ($light eq 'Y' || $light eq 'y' || $light eq 'H' || $light eq 'h')
{
    print '    Яркость от -100 до +100 (по умолчанию: ' . $brightness . '): ';

    # + Ввод данных с клавиатуры.
    my $brightness_temp = <STDIN>;

    # Удалить последний символ строки.
    chop($brightness_temp);

    if ($brightness_temp ne '')
    {
        $brightness = $brightness_temp;
    }

    print '    Контрастность от -100 до +100 (по умолчанию: ' . $contrast . '): ';

    # + Ввод данных с клавиатуры.
    my $contrast_temp = <STDIN>;

    # Удалить последний символ строки.
    chop($contrast_temp);

    if ($contrast_temp ne '')
    {
        $contrast = $contrast_temp;
    }
}

#
# -10-
#
print "\n";
print 'Для Nikon CoolPIX S4200 взять дату и время можно из jpg файла!' . "\n";
print 'Взять дату и время из jpg (y/n)? ';

# + Ввод данных с клавиатуры.
my $CoolPIX = <STDIN>;

# Удалить последний символ строки.
chop($CoolPIX);

if ($CoolPIX eq 'Y' || $CoolPIX eq 'y' || $CoolPIX eq 'H' || $CoolPIX eq 'h')
{
    print '    Внимание! Дата и время будут взяты из jpg файла...' . "\n";
}

#
# -11-
#
print "\n";
print 'Разрешение DPI (по умолчанию: ' . $resolution . '): ';

# + Ввод данных с клавиатуры.

```

```

my $resolution_temp = <STDIN>;

# Удалить последний символ строки.
chop($resolution_temp);

if ($resolution_temp ne '')
{
    $resolution = $resolution_temp;
}

# ---

# ---
# Начало измерения времени.
use Time::HiRes qw(gettimeofday);
my $start_time = gettimeofday;
# ---

my @filename_array;      # Массив для имён файлов.
my $filename;            # Текущий файл.
my $countfile;          # Текущее значение счётчика без нулей.
my $zerocf;             # Текущее значение счётчика с нулями слева (нумерация).
my $filename_cut;       # Текущий файл, только имя обрезанное.
my $exiftool;           # Результат работы программы exiftool (текст).
my $pos;               # Позиция Create Date в $exiftool.
my $exiftool_2;        # Текст от $pos и до конца.
my $pos_2;             # Позиция ":" в $exiftool_2;
my $exiftool_datetime; # Текущая дата создания файла в виде 2017:05:28 09:00:10.
my $year;              # Текущий год.
my $month;             # Текущий месяц.
my $day;               # Текущий день.
my $hour;              # Текущие часы.
my $minutes;          # Текущие минуты.
my $seconds;          # Текущие секунды.
my $stat;              # Результат работы программы stat (информация о файле - дата).
my $date_1;           # Год-месяц-день.
my $date_2;           # Часы:минуты:секунды.
my $number;           # Номер файла в имени файла (это для удобства).
my $newfilename;      # Новое название файла изображения с расширением.
my $newfilename_UTF8_txt; # Название файлов *.txt (это для дополнительной информации).
my $newfilename_1251_txt; # Название файлов *.txt (windows).
my $sizeimage;        # Размер изображения (ширина, высота).
my $ext;              # Здесь находится расширение файла (jpg).
my $sizeimage_x;      # Текущая ширина изображения.
my $sizeimage_y;      # Текущая высота изображения.
my $y;                # Текущая координата по вертикали, откуда будет начинаться прямоугольник.
my $convert;          # Готовая команда для ImageMagick.
my $null;             # Ненужный результат.
my $y_1;              # Координата, откуда по вертикали будет начинаться линия (верхняя линия).
my $y_2;              # Здесь координата, которая образует толщину линии (верхняя линия).
my $mogrify;          # Готовая команда для ImageMagick.
my $y_3;              # Координата, откуда по вертикали будет начинаться линия (нижняя линия).
my $y_4;              # Здесь координата, которая образует толщину линии (нижняя линия).
my $textimage;        # Текущая надпись во второй строке.
my $iconv;            # Готовая команда для iconv (перекодировка текста).
my $elapsed_time;     # Затраченное время в $elapsed_time (в секундах).
my $elapsed_sec;      # Затраченное время в секундах (4-ре знака после запятой).
my $elapsed_min;      # Затраченное время в минутах (4-ре знака после запятой).
my $elapsed_hour;     # Затраченное время в часах (4-ре знака после запятой).

print "\n";
opendir(DIRHANDLE, $catinimage) or die "Не могу открыть каталог $catinimage";

print 'Чтение каталога ' . $catinimage . '... ';

$countfile = 0;
while ( defined ($filename = readdir(DIRHANDLE)) )
{
    if ($filename ne '.' && $filename ne '..')
    {
        # Используем модуль Encode, без этого в $filename кракозябры...
        $filename = decode('utf8', $filename);

        # Имена файлов в массив.
        $filename_array[$countfile] = $filename;
        $countfile++;
    }
}

closedir(DIRHANDLE);

print 'Готово!' . "\n";

```

```

print 'Сортировка... ';
# sort - Буквы верхнего регистра предшествуют всем
# буквам нижнего регистра, а цифры предшествуют буквам.
@filename_array = sort(@filename_array);
print 'Готово!' . "\n";
print "\n";

for ($countfile = 0; $countfile < @filename_array;)
{
    $filename = $filename_array[$countfile];
    $countfile++;

    # Дополнить счетчик нулями слева.
    $zerocf = sprintf("%06d", $countfile);
    print $zerocf . ' . ';

    # Обрезать имя файлу (если нужно).
    if (length($filename) > $length_filename)
    {
        $filename_cut = substr($filename, 0, $length_filename) . '...';
    }
    else
    {
        $filename_cut = $filename;
    }

    print 'Файл: ' . $filename_cut . ' ';

if ($CoolPIX eq 'Y' || $CoolPIX eq 'y' || $CoolPIX eq 'H' || $CoolPIX eq 'h')
{
    # Получаем дату и время создания изображения из файла jpg (из метаданных).
    # Для этого в системе должен быть установлен exiftool.
    # # cd /usr/ports/graphics/p5-Image-ExifTool
    # # make install clean

    $exiftool = `exiftool '$catinimage$filename'`;

    # Ищем надпись Create Date.
    $pos = index($exiftool, 'Create Date', 0);

    if ($pos ne '-1')
    {
        # Вырезаем текст от $pos и до конца.
        $exiftool_2 = substr($exiftool, $pos, length($exiftool) - $pos);

        # Ищем ":".
        $pos_2 = index($exiftool_2, ':', 0);

        # Вырезаем дату и время.
        $exiftool_datetime = substr($exiftool_2, $pos_2 + 2, 19);

        # Достаём дату.
        $year = substr($exiftool_datetime, 0, 4); # Год.
        $month = substr($exiftool_datetime, 5, 2); # Месяц.
        $day = substr($exiftool_datetime, 8, 2); # День.

        # Достаём время.
        $hour = substr($exiftool_datetime, 11, 2); # Часы.
        $minutes = substr($exiftool_datetime, 14, 2); # Минуты.
        $seconds = substr($exiftool_datetime, 17, 2); # Секунды.
    }
    else
    {
        # Нет даты и времени.
        $year = 'xxxx'; # Год.
        $month = 'xx'; # Месяц.
        $day = 'xx'; # День.

        $hour = 'xx'; # Часы.
        $minutes = 'xx'; # Минуты.
        $seconds = 'xx (нет даты)'; # Секунды.
    }
}
else
{
    # Получаем дату и время создания изображения (файла).
    # Используется команда stat (см. ниже).

    # Приблизительно это так:

    # -
    # atime %Sa - время последнего доступа к файлу

```

```
#          (пример: загрузка текстового файла в текстовый редактор).

# mtime %Sm - время последнего изменения файла
#          (пример: изменение содержимого файла).

# ctime %Sc - время последнего изменения атрибутов файла
#          (пример: команды chmod, chown).

# birthtime %SB - время создания файла.
#          (пример: команда touch).
#
#          Примечание.
#          Birth - переводится как рождение.
# -
```

```
$stat = `stat -f %SB -t %Y%m%d%H%M%S '$catinimage$filename'`;
```

В \$stat возвращаемый результат (строка):

```

#
#
#      Месяц
#      |
#      | Часы
#      |
#      | Секунды.
#      |
#      ---
# Строка: 20070427111533
#      ---
#      |
#      |
#      | Минуты
#      |
#      | Число
#      |
#      | Год
#

```

```
# Достаём дату.
```

```
$year = substr($stat, 0, 4); # Год.  
$month = substr($stat, 4, 2); # Месяц.  
$day = substr($stat, 6, 2); # День.
```

Достаём время.

```
$hour = substr($stat, 8, 2); # Часы.  
$minutes = substr($stat, 10, 2); # Минуты.  
$seconds = substr($stat, 12, 2); # Секунды.
```

}

```
$date_1 = $year . '-' . $month . '-' . $day;
$date_2 = $hour . ':' . $minutes . ':' . $seconds;
```

```
print 'Data: ' . $date_1 . ', ' . $date_2 . ' . ';
```

```
# Дополнить номер нулями слева.
$number = sprintf("%06d", $numberfile);
```

```
# Новое название файла.
$newfilename = $titlefile . ' ' . '(' . $date_1 . ' , ' . $date_2 . ')' . ' - ' ;
$newfilename.= $number;
```

```
# Название файлов *.txt (для дополнительной информации).
$newfilename_UTF8_txt = $newfilename . ' - ' . 'UTF8.txt';
$newfilename_1251_txt = $newfilename . ' - ' . '1251.txt';
```

```
# Получить расширение файла.
$ext = substr($filename, length($filename) - 3, 3);
```

```
# Расширение в нижней регистр.
$ext = lc($ext);
```

```
# Новое название файла изображения с расширением.
$newfilename.= '.' . $ext;
```

Получаем размер изображения (ширина, высота).

```
# +++
$sizeimage = "identify -format \"%wx%h\"";
$sizeimage.= " ";
$sizeimage.= "\\\". $catinimage . $filename . "\\\";
$sizeimage = ` $sizeimage `;
```

```
print 'Разрешение: ' . $sizeimage . ' . ';
```

```
$sizeimage_x = substr($sizeimage, 0, 4); # Ширина изображения.  
$sizeimage_y = substr($sizeimage, 5, 4); # Высота изображения.
```

```
# Рисуем нижний прямоугольник.
# +++
```



```

#          Координата вверх/вниз
#          | Координата влево/вправо.
#          | |
# rectangle 0,50 50,0
#          | |
#          | По вертикали (y). | ---> Стать в позицию на изображение
#          По горизонтали (x). | (от левого верхнего угла).

# Получаем координату по вертикали, откуда будет начинаться прямоугольник...
$y = $sizeimage_y - $heightrectangle;

$convert = "convert -fill \"${colorrectangle}\" -draw"; #
$convert.= " \"rectangle 0,$sizeimage_y $sizeimage_x,$y\" "; #
$convert.= "\"\" . $catinimage . $filename . "\"\""; # Рисуем прямоугольник.
$convert.= "\"\" . $catoutimage . $newfilename . "\"\""; #
$null = ` $convert `; #

# Рисуем линии на нижнем прямоугольнике.
# +++

# Верхняя линия.
$y_1 = $y + $offsettop; # Координата, откуда по вертикали будет начинаться линия.
$y_2 = $y_1 + $depthline - 1; # Толщина линии.

$mogrify = "mogrify -fill \"${colorline}\" -draw"; #
$mogrify.= " \"rectangle 0,$y_2 $sizeimage_x,$y_1\" "; # Рисуем верхнюю линию
$mogrify.= "\"\" . $catoutimage . $newfilename . "\"\""; # (это тоже прямоугольник).
$null = ` $mogrify `; #

# Нижняя линия.
$y_3 = $sizeimage_y - $offsetbottom - $depthline; # Ещё одна координата...
$y_4 = $y_3 + $depthline - 1; # Толщина линии.

$mogrify = "mogrify -fill \"${colorline}\" -draw"; #
$mogrify.= " \"rectangle 0,$y_4 $sizeimage_x,$y_3\" "; # Рисуем нижнюю линию
$mogrify.= "\"\" . $catoutimage . $newfilename . "\"\""; # (это тоже прямоугольник).
$null = ` $mogrify `; #

# Пишем текстом. Надпись 1 (строка 1).
# +++
$mogrify = "mogrify -font " . $namefont . " -fill white -pointsize $sizefont";
$mogrify.= " " . "-gravity South -annotate +0+$offsettext_1 ";
$mogrify.= "\"\" . $ARG . "\"\"";
$mogrify.= "\"\" . $catoutimage . $newfilename . "\"\"";
$null = ` $mogrify `;

# Пишем текстом. Надпись 2 (строка 2).
# +++
$textimage = $textimage_camera . ' '; # Цифровая камера: Nikon, CoolPIX S4200.
$textimage.= $textimage_size . ' '; # Разрешение:
$textimage.= $sizeimage . ' '; # 3264x2448.
$textimage.= $textimage_date . ' '; # Дата
$textimage.= '(' . $textimage_utc . ')'; # UTC+02:00
$textimage.= $date_1 . ' ' . $date_2 . ' '; # Сама дата, плюс время.
$textimage.= $textimage_author; # Кто фотографировал.

$mogrify = "mogrify -font " . $namefont . " -fill white -pointsize $sizefont";
$mogrify.= " " . "-gravity South -annotate +0+$offsettext_2 ";
$mogrify.= "\"\" . $textimage . "\"\"";
$mogrify.= "\"\" . $catoutimage . $newfilename . "\"\"";
$null = ` $mogrify `;

# Добавить/уменьшить яркость и контрастность.
# +++
if ($light eq 'Y' || $light eq 'y' || $light eq 'H' || $light eq 'h')
{
    $mogrify = "mogrify -brightness-contrast ";
    $mogrify.= "$brightness/$contrast ";
    $mogrify.= "\"\" . $catoutimage . $newfilename . "\"\"";
    $null = ` $mogrify `;
}

# Устанавливаем разрешение (DPI).
# +++
$mogrify = "mogrify -units PixelsPerInch -density $resolution ";
$mogrify.= "\"\" . $catoutimage . $newfilename . "\"\"";
$null = ` $mogrify `;

# Копируем файл из $catinimage в $catoutimage.
# Попутно изменяем ему имя (это файл без подписи).
$null = `cp -p '$catinimage$filename' '$catoutimage$newfilename.no.signature'`;

```

```

# Изменяем права доступа файла $catoutimage$newfilename.no.signature.
# Файл доступен только владельцу (rw- --- ---).
chmod 0600, $catoutimage . $newfilename . '.no.signature';

# ---
# | Дополнительные файлы, дополнительная информация.
# ---

# *
# Создаём файл $newfilename_UTF8_txt.
open(OutFile, '>', $catoutimage . $newfilename_UTF8_txt);
close (OutFile);
# *

# *
# Записать дополнительную информацию в файл $newfilename_UTF8_txt.
open(OutFile, '>>', $catoutimage . $newfilename_UTF8_txt);

# => Строка 1.
# Перевод строки.
print OutFile "\n";

# => Строка 2.
# Что за фото (предварительно разделив строку
# на несколько строк, если нужно).
use Text::Wrap;
my $longstring = $ARG;
$Text::Wrap::columns = 78; # 78 - сколько символов в строке.
my $textsplit = wrap(' ', ' ', $longstring) . "\n";

print OutFile $textsplit;

# => Строка 3.
# Перевод строки.
print OutFile "\n";

# => Строка 4.
# Цифровая камера: Nikon, CoolPIX S4200.
print OutFile ' ' . $textimage_camera . "\n";

# => Строка 5.
# Разрешение: ширина x высота.
print OutFile ' ' . $textimage_size . ' ' . $sizeimage . '.' . "\n";

# => Строка 6.
# Дата: Г-М-Ч, Ч:М:С.
print OutFile ' ' . $textimage_date . ' ' .
    '(' . $textimage_utc . '): ' .
    $date_1 . ', ' . $date_2 . '.' . "\n";

# => Строка 7.
# Кто фотографировал.
print OutFile ' ' . $textimage_author . "\n";

close (OutFile);
# *

# *
# Перекодируем из UTF-8 в CP1251 (создаём ещё один файл,
# это для пользователей Windows).
$iconv = "iconv -c -f utf-8 -t cp1251 \" . $catoutimage . $newfilename_UTF8_txt . "\" ";
$iconv.= ">\" . $catoutimage . $newfilename_1251_txt . "\"";
$null = `$iconv`;
# *

# *
# Добавляем CR+LF.
$null = `unix2dos '$catoutimage$newfilename_1251_txt'`;
# *

#
# Конец дополнительным файлам и дополнительной информации.
#

print 'Готово!';
print ' ';
print 'См. в: ' . substr($catoutimage, 0, 8);
print "\n";

$numberfile++;
}

```



```
if ($countfile == 0)
{
    print 'Внимание!' . "\n";
    print '-----' . "\n";
    print 'Во входящем каталоге ' . $catinimage . ' нет файлов...';
}
```

Пример подписанной фотографии (не забудьте посмотреть фотографии в архиве или на диске)...



Украина, Харьковская обл., Балаклейский р-н, пос. Червоный Донец, Река СЕВЕРСКИЙ ДОНЕЦ.
Цифровая камера: Nikon, CoolPIX S4200. Разрешение: 2850x2050. Дата (UTC+02:00): 2017-06-06, 18:35:45. (С) Демидов С.В.


```

}
else
{
print "\n";

# Конец измерения времени.
my $end_time = gettimeofday;

# Затраченное время в $elapsed_time.
$elapsed_time = $end_time - $start_time;

# '%.4f' - Оставить 4-ре знака после запятой.
$elapsed_sec = sprintf('%.4f', $elapsed_time);
$elapsed_min = sprintf('%.4f', $elapsed_time / 60);
$elapsed_hour = sprintf('%.4f', $elapsed_time / 60 / 60);

print 'Затраченное время: ';

print $elapsed_sec . 'сек.';
print ' ';
print $elapsed_min . 'мин.';
print ' ';
print $elapsed_hour . 'час.';

print "\n";
print "\n";
print 'Всё! Нажмите "Enter"...';
$null = <STDIN>;
}

```



Демидов С.В.

Что содержит архив (файлы):

Файл: **signature-images - 1251.txt** - Надписи для фотографий.
 Файл: **signature-images - UTF8.txt** - Надписи для фотографий.
 Файл: **signature-images-font.ttf** - Шрифт **LiberationMono (bold)**.
 Файл: **signature-images.pl** - Сам скрипт.
 Файл: **signature-images.pdf** - Это описание.

Каталог **Images Examples** содержит (файлы) результат работы скрипта **signature-images.pl** (см. ниже).

Г. Балаклея. Центр (2017-05-06, 17:33:28) - 000001 - 1251.txt
 Г. Балаклея. Центр (2017-05-06, 17:33:28) - 000001 - UTF8.txt
 Г. Балаклея. Центр (2017-05-06, 17:33:28) - 000001.jpg
 Г. Балаклея. Центр (2017-05-06, 17:33:28) - 000001.jpg.no.signature
 ...
 ...
 ...
 Г. Балаклея. Центр (2017-05-07, 10:21:32) - 000005 - 1251.txt
 Г. Балаклея. Центр (2017-05-07, 10:21:32) - 000005 - UTF8.txt
 Г. Балаклея. Центр (2017-05-07, 10:21:32) - 000005.jpg
 Г. Балаклея. Центр (2017-05-07, 10:21:32) - 000005.jpg.no.signature

 Г. Харьков, сквер влюблённых на Пушкинской улице (2017-07-13, 11:03:54) - 000001 - 1251.txt
 Г. Харьков, сквер влюблённых на Пушкинской улице (2017-07-13, 11:03:54) - 000001 - UTF8.txt
 Г. Харьков, сквер влюблённых на Пушкинской улице (2017-07-13, 11:03:54) - 000001.jpg
 Г. Харьков, сквер влюблённых на Пушкинской улице (2017-07-13, 11:03:54) - 000001.jpg.no.signature
 ...
 ...
 ...
 Г. Харьков, сквер влюблённых на Пушкинской улице (2017-07-13, 11:04:49) - 000005 - 1251.txt
 Г. Харьков, сквер влюблённых на Пушкинской улице (2017-07-13, 11:04:49) - 000005 - UTF8.txt
 Г. Харьков, сквер влюблённых на Пушкинской улице (2017-07-13, 11:04:49) - 000005.jpg
 Г. Харьков, сквер влюблённых на Пушкинской улице (2017-07-13, 11:04:49) - 000005.jpg.no.signature

 Пос. Червоный Донец. Река СЕВЕРСКИЙ ДОНЕЦ (2017-06-06, 18:35:45) - 000001 - 1251.txt
 Пос. Червоный Донец. Река СЕВЕРСКИЙ ДОНЕЦ (2017-06-06, 18:35:45) - 000001 - UTF8.txt
 Пос. Червоный Донец. Река СЕВЕРСКИЙ ДОНЕЦ (2017-06-06, 18:35:45) - 000001.jpg
 Пос. Червоный Донец. Река СЕВЕРСКИЙ ДОНЕЦ (2017-06-06, 18:35:45) - 000001.jpg.no.signature
 ...
 ...
 ...
 Пос. Червоный Донец. Река СЕВЕРСКИЙ ДОНЕЦ (2017-06-06, 18:43:48) - 000005 - 1251.txt
 Пос. Червоный Донец. Река СЕВЕРСКИЙ ДОНЕЦ (2017-06-06, 18:43:48) - 000005 - UTF8.txt
 Пос. Червоный Донец. Река СЕВЕРСКИЙ ДОНЕЦ (2017-06-06, 18:43:48) - 000005.jpg
 Пос. Червоный Донец. Река СЕВЕРСКИЙ ДОНЕЦ (2017-06-06, 18:43:48) - 000005.jpg.no.signature

В архиве есть файл **signature-images.sla** - это файл **Scribus'a** (см. каталог **Scribus**).
 Плюс прибуды к **sla**-файлу...

Примечание.
 Использовался **Scribus** версии **1.4.5**.

Всё!

Коммерческое программное обеспечение, Не использовалось!