



# INFORMATYKA III: INSTRUKCJA 2

## *Skrypty BASH*

### 1 Materiały

Materiały do tego laboratorium można znaleźć w katalogu `~llaniewski/lab2_gorne` lub `~llaniewski/lab2_dolne` na `info3.meil.pw.edu.pl` ( górne: sala 120, dolne: sala 20B ).

Pod linuxem należy w terminalu wykonać:

```
> scp [login]@orange.meil.pw.edu.pl:~llaniewski/lab2_gorne .
> ./lab2_gorne [login na info3.meil.pw.edu.pl]
```

Następnie wyłączyć terminal i włączyć nowy. Polecenia te zgrają przygotowaną wersję programu ImageMagick i zainstalują, a także utworzą katalog LAB2 w katalogu domowym, w który zawarte są przykładowe obrazki do obróbki.

### 2 Obróbka obrazków

#### 2.1 convert

Głównym programem którego będziemy używać to `convert` z biblioteki ImageMagick. Program ten służy do najróżniejszego typu konwersji i zmiany właściwości obrazów — lecz potrafi także dodawać elementy do obrazu, a nawet tworzyć obrazy od zera. Najłatwiej zobaczyć jego użycie na przykładach:

**UWAGA: Zanim zaczniesz, skopiuj katalog ze zdjęciami do jakiegoś tymczasowego katalogu!**

- `convert plik.gif plik.jpg` — przekonwertuje plik w formacie GIF na format JPEG
- `convert plik1.jpg -resize 50% plik2.jpg` — zmniejszy obrazek dwukrotnie
- `convert plik1.jpg -resize 100 plik2.jpg` — zmniejszy obrazek, tak by krótszy wymiar był 100 pikseli
- `convert plik1.jpg -resize 100x100 plik2.jpg` — zmniejszy obrazek tak, by mieścił się w kwadracie 100 na 100 pikseli

- `convert plik1.jpg -resize 100x100! plik2.jpg` — zmniejszy obrazek dokładnie do rozmiaru 100 na 100 pixeli
- `convert -size 320x85 canvas:none -font Bookman-DemiItalic -pointsize 72 -draw "text 25,60 'Magick'" -channel RGBA -blur 0x6 -fill darkred -stroke magenta -draw "text 20,55 'Magick'" fuzzy-magick.jpg` — stworzy obrazek `fuzzy-magick.jpg`, z tekstem "Magick"

Wykonaj powyższe operacje, sprawdź efekty.

#### Ćwiczenia

Napisz skrypt który:

- Zmniejszy wszystkie pliki `jpg`
- Napisz skrypt który: Zmniejszy wszystkie pliki `jpg` umieszczając je w innym katalogu
- Napisz skrypt który: Skonwertuje wszystkie pliki `jpg` na `gif`, dodając końcówkę: `plik.jpg` → `plik.jpg.gif`
- Napisz skrypt który: Skonwertuje wszystkie pliki `jpg` na `gif`, zamieniając końcówkę `plik.jpg` → `plik.gif`
- Na każde zdjęcie naniesie tekst używając `-pointsize rozmiar -draw "text x,y 'Tekst'"`
- Na każde zdjęcie naniesie aktualną datę (komenda `date`)
- Na każde zdjęcie naniesie datę utworzenia tego zdjęcia (można ją wyciągnąć przy pomocy `stat -c %y plik`)
- Zmniejszy wszystkie obrazki z katalogu `drop1` i połączy je w animację przy pomocy `convert *.jpg animacja.gif`

#### 2.2 Automatyzacja

Spróbuj napisać skrypty wykonujące następujące zadania:

- Dodanie do obrazka ramki i wypisanie na niej wybranej informacji EXIF (można je wyciągnąć przy pomocy `identify -format "[%EXIF:*)" plik.jpg`) — np. modelu aparatu



- Połączy wszystkie obrazki w danym katalogu, zmniejszone do rozmiaru 10x10 w jeden duży obraz JPEG (`-append` łączy obrazy w pionie, a `+append` w poziomie)
- Analogicznie połącz obrazy z katalogu `drop2` w jeden duży w kratce 10 na 10 obrazów.<sup>1</sup>
- Jednym ciągiem, rozłóż animację GIF na pojedyncze obrazy JPEG, dopisz do każdego tekst, lub przyłóż filtr, a następnie złóż w nową animację GIF (kasując pliki tymczasowe)

Używając przykładu:

```
convert -size 90x60 xc:white -size 90x30 gradient: -append \
-rotate 90 a.jpg \
IMG_0174.jpg -resize 90x90! \
\( IMG_0041.jpg -resize 90x90! -clone 0 \
-compose CopyOpacity +matte -composite -repage +60+0 \) \
\( IMG_0396.jpg -resize 90x90! -clone 0 \
-compose CopyOpacity +matte -composite -repage +120+0 \) \
\( IMG_0174.jpg -resize 90x90! -clone 0 \
-compose CopyOpacity +matte -composite -repage +180+0 \) \
-delete 0 -compose Over -mosaic overlap_series.jpg
```

Napisz skrypt który stworzy obrazek z 5/6 obrazków z katalogu w sposób automatyczny w następujących krokach:

- w pętli `for` wygeneruje w pliku `tmp` komendę analogiczną do powyższej, tylko dla większej ilości obrazków (używając `echo linia > tmp` i `echo kolejna linia >> tmp`)
- zmieni uprawnienia do pliku `tmp` na wykonywalne
- wywoła `tmp` jako skrypt

## 2.3 Wczytywanie danych binarnych

Program `convert` może wczytać jako obrazek dane binarne. Dla przykładu tablice wartości typu `float`. Napisz program `obrazek.c` postaci:

<sup>1</sup>Podpowiedz: połącz obrazki `drop-0*.jpg` w poziomie, później `drop-1*.jpg` itd. A następnie wszystkie te podłużne obrazki połącz w całość w pionie.

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

const int n=100, m=100;

int main() {
    int i;
    char * tab;
    tab = malloc(n*m*sizeof(char));
    for (i=0; i<n*m; i++) {
        tab[i] = 255*i/(n*m);
    }
    fwrite(tab, sizeof(char), n*m, stdout);
    free(tab);
    return 0;
}
```

Następnie go skompiluj przy pomocy `gcc obrazek.c -o obrazek` i wykonaj `./obrazek > plik`. Teraz mamy plik binarny z liczbami typu `char` (1 bajt) od 0 do 255. Możemy przerobić go na obrazek JPEG wpisując:

```
convert -size 100x100 gray:plik plik.jpg
```

**Uwaga:** Zamiast `char` moglibyśmy użyć tablicy typu `float` i liczb z przedziału 0 do 1. Wtedy musielibyśmy skonwertować obrazek za pomocą:

```
convert -size 100x100 -depth 32 -define quantum:format=floating-point
gray:plik plik.jpg
```

## Ćwiczenia

- Zmniejsz dowolne zdjęcie do rozmiarów (dokładnie) 100 na 100 pixeli
- Zmodyfikuj komendę powyżej, tak by przerobić to zdjęcie na plik binarny
- Na podstawie `obrazek.c` napisz program `filtr.c` który pierw wczytuje z wejścia tablice (`fread`), z każdą liczbą wykonuje  $\bar{x} = -x$  i z powrotem wypisuje.
- Spróbuj przepuścić wybrany obrazek przez taki 'filtr'
- Napisz skrypt który automatycznie wszystkie pliki z katalogu zmniejszy do rozmiaru 100x100, przerobi na binarne, przepuści przez `filtr` i z powrotem zmieni na JPEG.