



# INFORMATYKA III: INSTRUKCJA

## *Dodatkowe Informacje*

### 1 BASH: skrypty

Pisanie skryptów, polega na spisaniu w pliku komend, które normalnie wpisywalibyśmy w linii poleceń. Taki plik możemy następnie oznaczyć jako wykonywalny komendą `chmod +x plik` i wykonać komendą `./plik`. Linia poleceń (BASH) służy do uruchamiania programów — dlatego:

**każda linijka skryptu wygląda następująco: „program argumenty”.**

Przeanalizuj fragment kodu, z zaznaczonymi **programami** i **opcjami**:

```
i=1
while test $i -lt 10
do
echo $i
cp plik plik.$i
i=$((expr $i + 1))
done
```

Gdy zapamiętamy tą zasadę, łatwo zobaczyć, że:

- `i=1` piszemy bez spacji ponieważ wtedy BASH wie, że to przypisanie, a nie program `i` z opcjami `= i 1`.
- w wyrażeniu `expr $i + 1`, musimy zachować spacje, żeby program `expr` dostał trzy argumenty „`$i`”, „`+`” i „`1`”, a nie jeden „`i+1`”.
- w pętli `while`, nie możemy wpisać „`i<10`”, lecz musimy użyć jakiegoś programu. Do wszelkiego rodzaju testów stworzony został program `test`. W tym wypadku podajemy mu za argumenty „`$i`”, „`-lt`” i „`10`”, gdzie opcja `-lt` oznacza „less than”.

#### 1.1 Przydatne programy

Jeśli już wiemy, że każdy skrypt w BASH to seria wywołanych programów, to potrzebne jest nam dużo małych programów, z których będziemy mogli tworzyć skrypty.

- `echo tekst` — Wpisuje `tekst` na ekran.

- `cat plik` — Wypisuje zawartość `pliku` na ekran
- `grep tekst` — Czyta z klawiatury tekst i wypisuje tylko linie zawierające `tekst`
- `grep tekst pliki` — Wyszukuje `tekst` w `plikach`
- `cd katalog` — Wchodzi do `katalogu`
- `ls katalog` — Wypisuje zawartość `katalogu` na ekran
- `cp pliki katalog` — Kopiuje `pliki` do `katalogu`
- `cp plik1 plik2` — Kopiuje plik o nazwie `plik1` do pliku o nazwie `plik2`
- `mv pliki katalog` — Przenosi `pliki` do `katalogu`
- `mv plik1 plik2` — Zmienia nazwę pliku z `plik1` na `plik2`
- `sed 's/tekst1/tekst2/g'` — Czyta z klawiatury tekst i go wypisuje zamieniając „`tekst1`” na „`tekst2`”

#### 1.2 Przekierowanie wejścia wyjścia

Standardowo wszystkie programy czytają z klawiatury i piszą na ekran. Można jednak zarówno pierwsze jak i drugie przekierować.

- `program > plik` — To co program wypisałby na ekran, zostanie wpisane do `pliku` (`plik` zostanie nadpisany jeśli istnieje)
- `program >> plik` — To co program wypisałby na ekran, zostanie dopisane do `pliku` (`plik` zostanie utworzony jeśli nie istniał)
- `program < plik` — Program dostanie zawartość `pliku`, tak jakbyśmy ją wpisali z klawiatury
- `program1 | program2` — To co `program1` wypisałby na ekran, zostanie wpisane „z klawiatury” do `program2`
- `'program'` lub `$(program)` — To co program wypisałby na ekran, zostanie wklejone w tym miejscu kodu (patrz przykłady). Znak `'` jest na klawiaturze przy tyldzie `~`.

Przykłady:



- `echo Tekst > plik` — wypisze „Tekst” do pliku (plik zostanie nadpisany jeśli istnieje)
- `echo Tekst >> plik` — dopisze „Tekst” do pliku (plik zostanie utworzony jeśli nie istniał)
- `grep Tekst < plik` — wyszuka w pliku linie zawierające „Tekst” i je wypisze na ekran
- `echo Tekst | sed 's/st/a/g'` — Zamieni w „Tekst” każde wystąpienie „st” na „a”. Więc wypisze na ekran „Teka”.
- `echo $nazwa | sed 's/\.txt/.dat/g'` — Zastąpi w zmiennej nazwa końcówkę `.txt` na `.dat`. Rezultat wypisze na ekran.
- `echo $nazwa | sed 's/\.txt/.dat/g'` — Zastąpi w zmiennej nazwa końcówkę `.txt` na `.dat`. Rezultat wypisze na ekran.
- `nazwa2=$(echo $nazwa | sed 's/\.txt/.dat/g')` — Jak poprzednio, lecz rezultat wypisze do zmiennej nazwa2.
- `ls katalog > plik` — wypisze zawartość katalogu do pliku (plik zostanie nadpisany jeśli istnieje)
- `cp 'ls' katalog` albo `cp $(ls) katalog` — skopiuje pliki do katalogu według listy zwróconej przez `ls`.
- `cp 'cat plik' katalog` bądź `cp $(cat plik) katalog` — skopiuje pliki do katalogu według listy zawartej w pliku.

### 1.3 Pętle i wyrażenia warunkowe

- ```
if program argumenty
then
polecenia1
else
polecenia2
fi
```

Jeśli wykonanie „program argumenty” się powiedzie (program zwróci 0), to wykonane zostaną `polecenia1`. W przeciwnym wypadku wykonane zostaną `polecenia2`.

- ```
while program argumenty
do
polecenia
done
```

Pętla, która będzie wykonywać `polecenia`, puki „program argumenty” będzie wykonywany z powodzeniem.
- ```
for i in lista
do
polecenia
done
```

Pętla, która po kolei każdy element listy wstawi do zmiennej `i`, a następnie wykona `polecenia`.

Dla przykładu:

```
for i in *.jpg
do
mv $i IMG/a_$i
done
```

Przeniesie każdy plik o końcówce `.jpg`, do katalogu `IMG` dodając im przedrostek `a_` (np.: `obrazek.jpg` zamieni na `IMG/a_obrazek.jpg`).



## 2 Obróbka obrazków

### 2.1 convert

- `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"

### 2.2 Wszystkie opcje convert

- `-adaptive-blur geometry` — adaptively blur pixels; decrease effect near edges
- `-adaptive-resize geometry` — adaptively resize image with data dependent triangulation.
- `-adaptive-sharpen geometry` — adaptively sharpen pixels; increase effect near edges
- `-adjoin` — join images into a single multi-image file
- `-affine matrix` — affine transform matrix
- `-alpha` — on, activate, off, deactivate, set, opaque, copy", transparent, extract, background, or shape the alpha channel
- `-annotate geometry text` — annotate the image with text

- `-antialias` — remove pixel-aliasing
- `-append` — append an image sequence
- `-authenticate value` — decipher image with this password
- `-auto-gamma` — automagically adjust gamma level of image
- `-auto-level` — automagically adjust color levels of image
- `-auto-orient` — automagically orient image
- `-background color` — background color
- `-bench iterations` — measure performance
- `-bias value` — add bias when convolving an image
- `-black-threshold value` — force all pixels below the threshold into black
- `-blue-primary point` — chromaticity blue primary point
- `-blue-shift factor` — simulate a scene at nighttime in the moonlight
- `-blur geometry` — reduce image noise and reduce detail levels
- `-border geometry` — surround image with a border of color
- `-bordercolor color` — border color
- `-brightness-contrast geometry` — improve brightness / contrast of the image
- `-caption string` — assign a caption to an image
- `-cdl filename` — color correct with a color decision list
- `-channel type` — apply option to select image channels
- `-charcoal radius` — simulate a charcoal drawing
- `-chop geometry` — remove pixels from the image interior
- `-clamp` — restrict colors from 0 to the quantum depth
- `-clip` — clip along the first path from the 8BIM profile



- `-clip-mask filename` — associate clip mask with the image
- `-clip-path id` — clip along a named path from the 8BIM profile
- `-clone index` — clone an image
- `-clut` — apply a color lookup table to the image
- `-contrast-stretch geometry` — improve the contrast in an image by ‘stretching’ the range of intensity value
- `-coalesce` — merge a sequence of images
- `-colorize value` — colorize the image with the fill color
- `-color-matrix matrix` — apply color correction to the image.
- `-colors value` — preferred number of colors in the image
- `-colorspace type` — set image colorspace
- `-combine` — combine a sequence of images
- `-comment string` — annotate image with comment
- `-compose operator` — set image composite operator
- `-composite` — composite image
- `-compress type` — image compression type
- `-contrast` — enhance or reduce the image contrast
- `-convolve coefficients` — apply a convolution kernel to the image
- `-crop geometry` — crop the image
- `-cycle amount` — cycle the image colormap
- `-decipher filename` — convert cipher pixels to plain
- `-debug events` — display copious debugging information
- `-define format:option` — define one or more image format options
- `-deconstruct` — break down an image sequence into constituent parts
- `-delay value` — display the next image after pausing
- `-delete index` — delete the image from the image sequence
- `-density geometry` — horizontal and vertical density of the image
- `-depth value` — image depth
- `-despeckle` — reduce the speckles within an image
- `-direction type` — render text right-to-left or left-to-right
- `-display server` — get image or font from this X server
- `-dispose method` — layer disposal method
- `-distort type coefficients` — distort image
- `-dither method` — apply error diffusion to image
- `-draw string` — annotate the image with a graphic primitive
- `-duplicate count,indexes` — duplicate an image one or more times
- `-edge radius` — apply a filter to detect edges in the image
- `-emboss radius` — emboss an image
- `-encipher filename` — convert plain pixels to cipher pixels
- `-encoding type` — text encoding type
- `-endian type` — endianness (MSB or LSB) of the image
- `-enhance` — apply a digital filter to enhance a noisy image
- `-equalize` — perform histogram equalization to an image
- `-evaluate operator value` — evaluate an arithmetic, relational, or logical expression
- `-evaluate-sequence operator` — evaluate an arithmetic, relational, or logical expression for an image sequence
- `-extent geometry` — set the image size
- `-extract geometry` — extract area from image
- `-family name` — render text with this font family



- `-fft` — implements the discrete Fourier transform (DFT)
- `-fill color` — color to use when filling a graphic primitive
- `-filter type` — use this filter when resizing an image
- `-flatten` — flatten a sequence of images
- `-flip` — flip image in the vertical direction
- `-floodfill geometry color` — floodfill the image with color
- `-flop` — flop image in the horizontal direction
- `-font name` — render text with this font
- `-format string` — output formatted image characteristics
- `-frame geometry` — surround image with an ornamental border
- `-function name` — apply a function to the image
- `-fuzz distance` — colors within this distance are considered equal
- `-fx expression` — apply mathematical expression to an image channel(s)
- `-gamma value` — level of gamma correction
- `-gaussian-blur geometry` — reduce image noise and reduce detail levels
- `-geometry geometry` — preferred size or location of the image
- `-gravity type` — horizontal and vertical text placement
- `-green-primary point` — chromaticity green primary point
- `-help` — print program options
- `-identify` — identify the format and characteristics of the image
- `-ift` — implements the inverse discrete Fourier transform (DFT)
- `-implode amount` — implode image pixels about the center
- `-insert index` — insert last image into the image sequence
- `-intent type` — type of rendering intent when managing the image color
- `-interlace type` — type of image interlacing scheme
- `-interline-spacing value` — the space between two text lines
- `-interpolate method` — pixel color interpolation method
- `-interword-spacing value` — the space between two words
- `-kerning value` — the space between two characters
- `-label string` — assign a label to an image
- `-lat geometry` — local adaptive thresholding
- `-layers method` — optimize or compare image layers
- `-level value` — adjust the level of image contrast
- `-limit type value` — pixel cache resource limit
- `-linear-stretch geometry` — linear with saturation histogram stretch
- `-liquid-rescale geometry` — rescale image with seam-carving
- `-log format` — format of debugging information
- `-loop iterations` — add Netscape loop extension to your GIF animation
- `-mask filename` — associate a mask with the image
- `-mattecolor color` — frame color
- `-median radius` — apply a median filter to the image
- `-mode radius` — make each pixel the 'predominant color' of the neighborhood
- `-modulate value` — vary the brightness, saturation, and hue
- `-monitor` — monitor progress
- `-monochrome` — transform image to black and white



- `-morph value` — morph an image sequence
- `-morphology method kernel` — apply a morphology method to the image
- `-motion-blur geometry` — simulate motion blur
- `-negate` — replace each pixel with its complementary color
- `-noise radius` — add or reduce noise in an image
- `-normalize` — transform image to span the full range of colors
- `-opaque color` — change this color to the fill color
- `-ordered-dither NxN` — ordered dither the image
- `-orient type` — image orientation
- `-page geometry` — size and location of an image canvas (setting)
- `-paint radius` — simulate an oil painting
- `-ping` — efficiently determine image attributes
- `-pointsize value` — font point size
- `-polaroid angle` — simulate a Polaroid picture
- `-posterize levels` — reduce the image to a limited number of color levels
- `-precision value` — set the maximum number of significant digits to be printed
- `-preview type` — image preview type
- `-print string` — interpret string and print to console
- `-process image-filter` — process the image with a custom image filter
- `-profile filename` — add, delete, or apply an image profile
- `-quality value` — JPEG/MIFF/PNG compression level
- `-quantize colorspace` — reduce image colors in this colorspace
- `-quiet` — suppress all warning messages
- `-radial-blur angle` — radial blur the image
- `-raise value` — lighten/darken image edges to create a 3-D effect
- `-random-threshold low,high` — random threshold the image
- `-red-primary point` — chromaticity red primary point
- `-regard-warnings` — pay attention to warning messages.
- `-region geometry` — apply options to a portion of the image
- `-remap filename` — transform image colors to match this set of colors
- `-render` — render vector graphics
- `-repage geometry` — size and location of an image canvas
- `-resample geometry` — change the resolution of an image
- `-resize geometry` — resize the image
- `-respect-parentheses` — settings remain in effect until parenthesis boundary.
- `-roll geometry` — roll an image vertically or horizontally
- `-rotate degrees` — apply Paeth rotation to the image
- `-sample geometry` — scale image with pixel sampling
- `-sampling-factor geometry` — horizontal and vertical sampling factor
- `-scale geometry` — scale the image
- `-scene value` — image scene number
- `-seed value` — seed a new sequence of pseudo-random numbers
- `-segment values` — segment an image
- `-selective-blur geometry` — selectively blur pixels within a contrast threshold
- `-separate` — separate an image channel into a grayscale image



- `-sepia-tone threshold` — simulate a sepia-toned photo
- `-set attribute value` — set an image attribute
- `-shade degrees` — shade the image using a distant light source
- `-shadow geometry` — simulate an image shadow
- `-sharpen geometry` — sharpen the image
- `-shave geometry` — shave pixels from the image edges
- `-shear geometry` — slide one edge of the image along the X or Y axis
- `-sigmoidal-contrast geometry` — increase the contrast without saturating highlights or shadows
- `-smush offset` — smush an image sequence together
- `-size geometry` — width and height of image
- `-sketch geometry` — simulate a pencil sketch
- `-solarize threshold` — negate all pixels above the threshold level
- `-splice geometry` — splice the background color into the image
- `-spread radius` — displace image pixels by a random amount
- `-statistic type geometry` — replace each pixel with corresponding statistic from the neighborhood
- `-strip` — strip image of all profiles and comments
- `-stroke color` — graphic primitive stroke color
- `-strokewidth value` — graphic primitive stroke width
- `-stretch type` — render text with this font stretch
- `-style type` — render text with this font style
- `-swap indexes` — swap two images in the image sequence
- `-swirl degrees` — swirl image pixels about the center
- `-synchronize` — synchronize image to storage device
- `-taint` — mark the image as modified
- `-texture filename` — name of texture to tile onto the image background
- `-threshold value` — threshold the image
- `-thumbnail geometry` — create a thumbnail of the image
- `-tile filename` — tile image when filling a graphic primitive
- `-tile-offset geometry` — set the image tile offset
- `-tint value` — tint the image with the fill color
- `-transform` — affine transform image
- `-transparent color` — make this color transparent within the image
- `-transparent-color color` — transparent color
- `-transpose` — flip image in the vertical direction and rotate 90 degrees
- `-transverse` — flop image in the horizontal direction and rotate 270 degrees
- `-treedepth value` — color tree depth
- `-trim` — trim image edges
- `-type type` — image type
- `-undercolor color` — annotation bounding box color
- `-unique-colors` — discard all but one of any pixel color.
- `-units type` — the units of image resolution
- `-unsharp geometry` — sharpen the image
- `-verbose` — print detailed information about the image
- `-version` — print version information
- `-view` — FlashPix viewing transforms
- `-vignette geometry` — soften the edges of the image in vignette style



- `-virtual-pixel method` — access method for pixels outside the boundaries of the image
- `-wave geometry` — alter an image along a sine wave
- `-weight type` — render text with this font weight
- `-white-point point` — chromaticity white point
- `-white-threshold value` — force all pixels above the threshold into white
- `-write filename` — write images to this file