

Wprowadzenie do grafiki maszynowej. Biblioteka Libjpeg

Aleksander Denisiuk
Uniwersytet Warmińsko-Mazurski
Olsztyn, ul. Słoneczna 54
denisjuk@matman.uwm.edu.pl

Biblioteka Libjpeg

Libjpeg

Najnowsza wersja tego dokumentu dostępna jest pod adresem

<http://wmii.uwm.edu.pl/~denisjuk/uwm>

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ Wczytywanie
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ `main`

Libjpeg

Wprowadzenie

Libjpeg

❖ Wprowadzenie

❖ Struktury

❖ Wczytywanie

❖ Przetwarzanie

❖ Zapisywanie

❖ main

- Rozwijana przez *Independent JPEG Group*
 - ✦ nie są związani z *ISO/IEC JTC1/SC29/WG1 standards committee* (aka *JPEG*)
- Ma zaimplementowane prawie wszystkie opcje formatu JPEG
- Zgodna z ANSI C
- Pierwsze wydanie było 7 października 1991
 - ✦ najnowsza stabilna wersja 9b
 - 17 stycznia 2016
- Wolna licencja

Wykorzystanie biblioteki

Libjpeg

❖ Wprowadzenie

❖ Struktury

❖ Wczytywanie

❖ Przetwarzanie

❖ Zapisywanie

❖ main

- W kodzie (C, C++):

```
#include <jpeglib.h>
```

- W opcjach konsolidatora (linkera):

```
-ljpeg
```

Struktury

Libjpeg

❖ Wprowadzenie

❖ Struktury

❖ Wczytywanie

❖ Przetwarzanie

❖ Zapisywanie

❖ main

- `jpeg_decompress_struct` *JPEG decompression object*
 - ◆ zawiera informacje o pliku
 - ◆ jest inicjalizowany po wczytaniu nagłówka
- `jpeg_compress_struct` *JPEG compression object*
 - ◆ zawiera informacje o parametrach kompresji
- `jpeg_error_mgr` *Error handler structure*
 - ◆ zawiera informacje o callbacku

Wczytywanie obrazu

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ **Wczytywanie**
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ main

● Inicjalizacja struktur

```
struct jpeg_decompress_struct cinfo;  
struct jpeg_error_mgr jerr;  
.  
.  
.  
.  
.  
.  
.  
.  
.  
.  
cinfo.err = jpeg_std_error(&jerr);  
jpeg_create_decompress(&cinfo);
```

● Otwieranie pliku

```
FILE *infile = fopen(file_name, "rb");  
if (!infile) return (ERROR);  
jpeg_stdio_src(&cinfo, infile);
```

Wczytywanie nagłówka

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ **Wczytywanie**
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ main

```
jpeg_read_header(&cinfo, TRUE);  
jpeg_start_decompress(&cinfo);
```


Otrzymanie parametrów obrazu

Libjpeg

❖ Wprowadzenie

❖ Struktury

❖ Wczytywanie

❖ Przetwarzanie

❖ Zapisywanie

❖ main

- Z pól struktury `jpeg_decompress_struct`

```
JDIMENSION width = cinfo.output_width;  
JDIMENSION height = cinfo.output_height;  
int num_components = cinfo.out_color_components;  
J_COLOR_SPACE color_space  
                = cinfo.out_color_space;
```

- `num_components`: 1 lub 3

- `J_COLOR_SPACE`

- ◆ `JCS_UNKNOWN`
- ◆ `JCS_GRAYSCALE`
- ◆ `JCS_RGB`
- ◆ `JCS_YCbCr`
- ◆ `JCS_CMYK`
- ◆ `JCS_YCCK`

Wczytywanie obrazka

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ **Wczytywanie**
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ main

```
JSAMPARRAY row_pointers = (JSAMPARRAY)
    malloc(sizeof(j_common_ptr) * height);
for (y=0; y<height; y++)
    row_pointers[y] = (JSAMPROW) malloc(rowbytes);

y=0;
JSAMPARRAY tmp = row_pointers;
while( cinfo.output_scanline < cinfo.image_height
    y = jpeg_read_scanlines( &cinfo, tmp, 1 );
    tmp +=y;
}
```

Końcówka Zwolnienie pamięci, zamykanie pliku

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ **Wczytywanie**
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ main

```
jpeg_finish_decompress (&cinfo);  
jpeg_destroy_decompress (&cinfo);  
free (row_pointer[0]);  
fclose (infile);
```

Przetwarzanie obrazka

Libjpeg

❖ Wprowadzenie

❖ Struktury

❖ Wczytywanie

❖ Przetwarzanie

❖ Zapisywanie

❖ main

```
if (color_space != JCS_RGB)    return;

for (y=0; y<height; y++) {
    JSAMPROW row = row_pointers[y];
    for (x=0; x<width; x++) {
        JSAMPROW ptr = &(row[x*3]);
        printf("Pixel at position [ %d - %d ] \
            has the following RGB values: \
            %d - %d - %d\n",
            x, y, ptr[0], ptr[1], ptr[2]);

        ptr[0] = 0;
        ptr[1] = ptr[2];
    }
}
```

Utworzenie pliku

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ Wczytywanie
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ main

```
FILE *outfile = fopen(filename, "wb");  
if (!outfile) return (ERROR);
```

Inicjalizacja struktur

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ Wczytywanie
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ main

```
struct jpeg_compress_struct cinfo;  
struct jpeg_error_mgr jerr;
```

```
cinfo.err = jpeg_std_error( &jerr );  
jpeg_create_compress(&cinfo);  
jpeg_stdio_dest(&cinfo, outfile);
```

Ustawianie parametrów obrazu

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ Wczytywanie
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ main

```
cinfo.image_width = width;  
cinfo.image_height = height;  
cinfo.input_components = num_components;  
cinfo.in_color_space = color_space;  
  
int quality = 75;  
jpeg_set_defaults( &cinfo );  
jpeg_set_quality (&cinfo, quality, TRUE);
```

Zapisywanie obrazka

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ Wczytywanie
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ main

```
tmp = row_pointers;
while( cinfo.next_scanline < cinfo.image_height) {
    y = jpeg_write_scanlines( &cinfo, tmp, 1);
    tmp +=y;
}
```

- `row_pointers` jest strukturą, utworzoną tak jak przy wczytywaniu obrazka.

Końcówka, zamykanie pliku, zwolnienie pamięci

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ Wczytywanie
- ❖ Przetwarzanie
- ❖ Zapisywanie
- ❖ main

```
jpeg_finish_compress( &cinfo );  
jpeg_destroy_compress( &cinfo );  
fclose( outfile );
```

```
for (y=0; y<height; y++)  
    free(row_pointers[y]);  
free(row_pointers);
```

Typowy *main*

Libjpeg

- ❖ Wprowadzenie
- ❖ Struktury
- ❖ Wczytywanie
- ❖ Przetwarzanie
- ❖ Zapisywanie

❖ *main*

```
int main(int argc, char **argv) {  
    if (argc != 3) return (ERROR);  
  
    read_jpeg_file(argv[1]);  
    process_file();  
    write_jpeg_file(argv[2]);  
    return 0;  
}
```