

# ICO2OBJ

Преобразователь картинок в формате ICO в объектные файлы  
(Версия 0.1)

Yellow Rabbit

Позволяет конвертировать графические файлы формата ICO в объектные файлы для последующей линковки в исполняемые образы для БК11М<sup>1</sup>.

Входными параметрами являются: перечень имен графических файлов, имя выходного файла и несколько управляющих ключей. На выходе создается объектный файл для последующей линковки.

Конвертор накладывает следующие ограничения на формат входных файлов изображений:

- 4 бита на точку.

---

<sup>1</sup> GIT-репозиторий ассемблера, линковщика и утилит <https://github.com/yrabbit>

27 Декабря 2013 года в 15:36

## 1. Общая схема программы.

```

⟨ Включение заголовочных файлов 15 ⟩
⟨ Директивы препроцессора ⟩
⟨ Константы 8 ⟩
⟨ Собственные типы данных 11 ⟩
⟨ Прототипы 6 ⟩
⟨ Глобальные переменные 2 ⟩
int main(int argc, char *argv[])
{
    ⟨ Данные программы 3 ⟩
    const char *picname;
    ⟨ Разобрать командную строку 14 ⟩
    /* Поочередно обрабатываем все заданные файлы картинок */
    cur_input = 0;
    while ((picname = config.picnames[cur_input]) ≠ ~) {
        ⟨ Открыть файл картинки 4 ⟩
        handleOneFile(fpic);
        fclose(fpic);
        ++ cur_input;
    }
    return (0);
}

```

## 2. Номер текущего обрабатываемого файла картинки.

⟨ Глобальные переменные 2 ⟩ ≡  
**static int** *cur\_input*;

Смотри также секции 9, 10, 12, и 16.

Этот код используется в секции 1.

## 3. ⟨ Данные программы 3 ⟩ ≡ **FILE** \**fpic*;

Этот код используется в секции 1.

## 4. ⟨ Открыть файл картинки 4 ⟩ ≡ *fpic* = *fopen*(*picname*, "r"); **if** (*fpic* ≡ ~) { PRINTERR("Can't open %s\n", *picname*); **return** (ERR\_CANTOPEN); }

Этот код используется в секции 1.

## 5. Обработать один файл картинки.

```

static void handleOneFile(FILE *fpic)
{ }

```

## 6. ⟨ Прототипы 6 ⟩ ≡ **static void** *handleOneFile*(**FILE** \*);

Этот код используется в секции 1.

## 7. Разбор параметров командной строки.

Для этой цели используется достаточно удобная свободная библиотека *argp*.

```
#define VERSION "0.1"
```

## 8. <Константы 8> ≡

```
const char *argp_program_version = "ico2obj, "VERSION;
const char *argp_program_bug_address = "<yellowrabbit@bk.ru>";
```

Этот код используется в секции 1.

## 9. <Глобальные переменные 2> +≡

```
static char argp_program_doc[] = "Convert ICO images to object file";
static char args_doc[] = "file [...]";
```

## 10. Распознаются следующие опции:

- o — имя выходного файла.
- v — вывод дополнительной информации (возможно указание дважды);
- l LABEL — шаблон метки для изображения (6 символов RADIX50);
- [0123] — номера цветов для битов.

## <Глобальные переменные 2> +≡

```
static struct argp_option options[] = {
    {"output", 'o', "FILENAME", 0, "Output filename"},
    {"verbose", 'v', ~, 0, "Verbose output"},
    {"label", 'l', "LABEL", 0, "Label for images"},
    {"color0", '0', "COLOR", 0, "Color number for bits 00"},
    {"color1", '1', "COLOR", 0, "Color number for bits 01"},
    {"color2", '2', "COLOR", 0, "Color number for bits 10"},
    {"color3", '3', "COLOR", 0, "Color number for bits 11"},
    {0}
};
static error_t parse_opt(int, char *, struct argp_state *);
static struct argp argp = {options, parse_opt, args_doc, argp_program_doc};
```

## 11. Эта структура используется для получения результатов разбора параметров командной строки.

## <Собственные типы данных 11> ≡

```
typedef struct _Arguments {
    int verbosity;
    char output_filename[FILENAME_MAX]; /* Имя файла с текстом */
    char label[7]; /* Метка для картинок в объектном файле */
    char **picnames; /* Имена файлов картинок picnames[?] == NULL -> конец имен */
    int colors[4]; /* Номера цветов для битов */
} Arguments;
```

Этот код используется в секции 1.

## 12. <Глобальные переменные 2> +≡

```
static Arguments config = {0, {0}, {0}, ~, {0},};
```

13. Задачей данного простого парсера является заполнение структуры **Arguments** из указанных параметров командной строки.

```
static error_t parse_opt(int key, char *arg, struct argp_state *state)
{
    Arguments *arguments;
    arguments = (Arguments *) state->input;
    switch (key) {
    case 'l':
        if (strlen(arg) == 0 || strlen(arg) > 6) return (ARGP_ERR_UNKNOWN);
        strcpy(arguments->label, arg);
        break;
    case 'v': ++arguments->verbosity;
        break;
    case 'o':
        if (strlen(arg) == 0) return (ARGP_ERR_UNKNOWN);
        strncpy(arguments->output_filename, arg, FILENAME_MAX - 1);
        break;
    case '0': arguments->colors[0] = atoi(arg);
        break;
    case '1': arguments->colors[1] = atoi(arg);
        break;
    case '2': arguments->colors[2] = atoi(arg);
        break;
    case '3': arguments->colors[3] = atoi(arg);
        break;
    case ARGP_KEY_ARG: /* Имена объектных файлов */
        arguments->picnames = &state->argv[state->next - 1]; /* Останавливаем разбор параметров */
        state->next = state->argc;
        break;
    default: break;
        return (ARGP_ERR_UNKNOWN);
    }
    return (0);
}
```

14.

```
#define ERR_SYNTAX 1
#define ERR_CANTOPEN 2
#define ERR_CANTCREATE 3
<Разобрать командную строку 14> ≡
    argp_parse(&argp, argc, argv, 0, 0, &config); /* Проверка параметров */
    if (strlen(config->output_filename) == 0) {
        PRINTERR("No output filename specified\n");
        return (ERR_SYNTAX);
    }
    if (config->picnames == ~) {
        PRINTERR("No input filenames specified\n");
        return (ERR_SYNTAX);
    }
```

Этот код используется в секции 1.

**15.**     $\langle$  Включение заголовочных файлов 15  $\rangle \equiv$

```
#include <string.h>
#include <stdlib.h>
#ifdef _linux_
#include <stdint.h>
#endif
#include <argp.h>
```

Этот код используется в секции 1.

**16.**

$\langle$  Глобальные переменные 2  $\rangle + \equiv$

```
#define PRINTVERB (level,fmt,a... ) (((config.verbosity)  $\geq$  level) ? printf((fmt),##a) : 0)
#define PRINTERR (fmt,a... ) fprintf(stderr,(fmt),##a)
```

**17. Индекс.**

linux: 15.  
Arguments: 11.  
*a*: 16.  
*arg*: 13.  
*argc*: 1, 13, 14.  
*argp*: 10, 14.  
 ARGP\_ERR\_UNKNOWN: 13.  
 ARGP\_KEY\_ARG: 13.  
*argp\_option*: 10.  
*argp\_parse*: 14.  
*argp\_program\_bug\_address*: 8.  
*argp\_program\_doc*: 9, 10.  
*argp\_program\_version*: 8.  
*argp\_state*: 10, 13.  
*args\_doc*: 9, 10.  
Arguments: 11, 12, 13.  
*arguments*: 13.  
*argv*: 1, 13, 14.  
*atoi*: 13.  
*colors*: 11, 13.  
*config*: 1, 12, 14, 16.  
*cur\_input*: 1, 2.  
 ERR\_CANTCREATE: 14.  
 ERR\_CANTOPEN: 4, 14.  
 ERR\_SYNTAX: 14.  
*error\_t*: 10, 13.  
*fclose*: 1.  
 FILENAME\_MAX: 11, 13.  
*fmt*: 16.  
*fopen*: 4.  
*fpic*: 1, 3, 4, 5.  
*fprintf*: 16.  
*handleOneFile*: 1, 5, 6.  
*input*: 13.  
*key*: 13.  
*label*: 11, 13.  
*level*: 16.  
*main*: 1.  
*next*: 13.  
*options*: 10.  
*output\_filename*: 11, 13, 14.  
*parse\_opt*: 10, 13.  
*picname*: 1, 4.  
*picnames*: 1, 11, 13, 14.  
 PRINTERR: 4, 14, 16.  
*printf*: 16.  
 PRINTVERB: 16.  
*state*: 13.  
**static**: 10.  
*stderr*: 16.  
*strcpy*: 13.

*strlen*: 13, 14.  
*strncpy*: 13.  
*verbosity*: 11, 13, 16.  
 VERSION: 7, 8.

- 〈 Включение заголовочных файлов 15 〉 Используется в секции 1.
- 〈 Глобальные переменные 2, 9, 10, 12, 16 〉 Используется в секции 1.
- 〈 Данные программы 3 〉 Используется в секции 1.
- 〈 Константы 8 〉 Используется в секции 1.
- 〈 Открыть файл картинки 4 〉 Используется в секции 1.
- 〈 Прототипы 6 〉 Используется в секции 1.
- 〈 Разобрать командную строку 14 〉 Используется в секции 1.
- 〈 Собственные типы данных 11 〉 Используется в секции 1.

ICO2OBJ

	Секция	Страница
Общая схема программы .....	1	2
Разбор параметров командной строки .....	7	3
Индекс .....	17	6