

Пояснения ко второму заданию

```
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <aio.h>
#include <fcntl.h>
#include <unistd.h>
#include <signal.h>
#include <string.h>
#include <errno.h>
```

```
#include <sys/types.h>
#include <sys/stat.h>
#include <signal.h>
```

```
#include <stdint.h>
#include <inttypes.h>
```

Рекомендуется использовать для каждой асинхронной операции следующую структуру данных

```
struct aio_operation {
    struct aiocb aio;
    char *buffer;
    int write_operation;
    void* next_operation;
};
```

открытие и создание файлов - open

```
open(read_filename, O_RDONLY | O_NONBLOCK, 0666)
open(write_filename, O_CREAT | O_WRONLY | O_TRUNC | O_NONBLOCK, 0666)
```

получение размера файла - через функцию fstat

вызов асинхронного чтения - aio_read

вызов асинхронной записи - aio_write

проверка прочитанных байтов - aio_return

ожидание асинхронных операций - aio_suspend

функция завершения (сами пишете, шаблон ниже)

```
void aio_completion_handler(signal_t signal) {
    struct aio_operation *aio_op = (struct aio_operation *)signal.sival_ptr;
```

```
if (aio_op->write_operation) {  
    // операция записи  
  
    } else {  
        //операция чтения  
  
    }  
}
```

Можно (лайфхак) для того, чтобы поймать конец копирования, использовать функции
raise / sleep