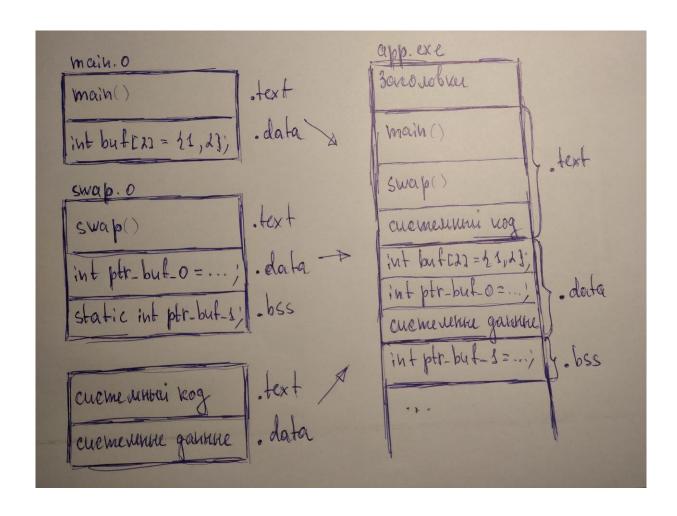
Схема распределения памяти в программе на Си

Компоновщик

```
D (.data)
#include "swap.h"
                                          #include <stdlib.h>
                                          #include "swap.h"
                        T (.text)
                                                                           D (.data)
int buf[2] = {1, 2};
                                          extern int buf[];
int main(void)
                                          int *ptr buf 0 = &buf[0];
                                          static int *ptr_buf_1;
    swap();
                                                                     b (.bss)
                                          void swap(void)
    return 0;
                  U
                                                                   T (.text)
                                               int *tmp = NULL;
                                               ptr buf 1 = &buf[1];
                          Отсутствует в
                                               tmp = ptr buf 0;
                         таблице символов
                                               ptr buf 0 = ptr buf 1;
                                               ptr buf 1 = tmp;
```

Компоновщик



Компоновщик

Процесс компоновки состоит из нескольких шагов:

• Перемещение (relocation)

«Склеивание» нескольких секций в одну.

• Разрешение ссылок (reference resolving)

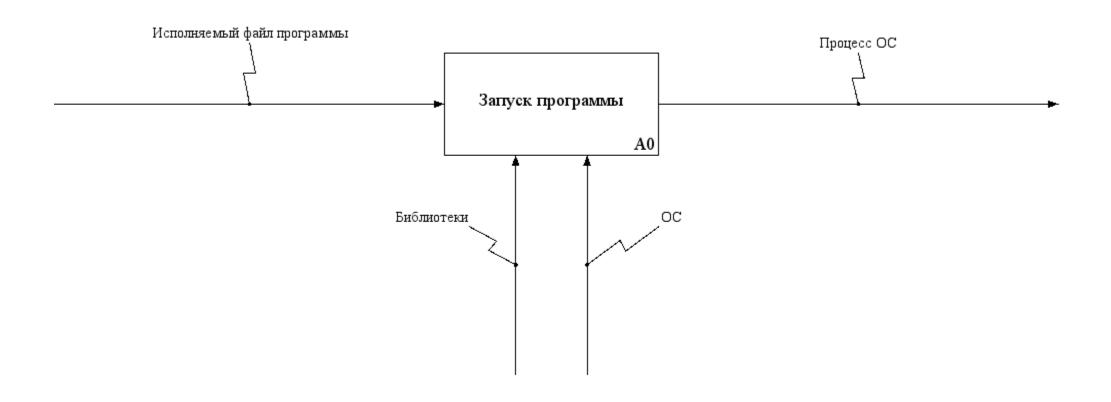
Ссылки внутри каждого файла уже разрешены.

- Поиск кода, который вызывает что-то за пределами своей исходной секции.
- Поиск места, где теперь располагается вызываемые код.
- Замена «поддельного» адреса на настоящий.

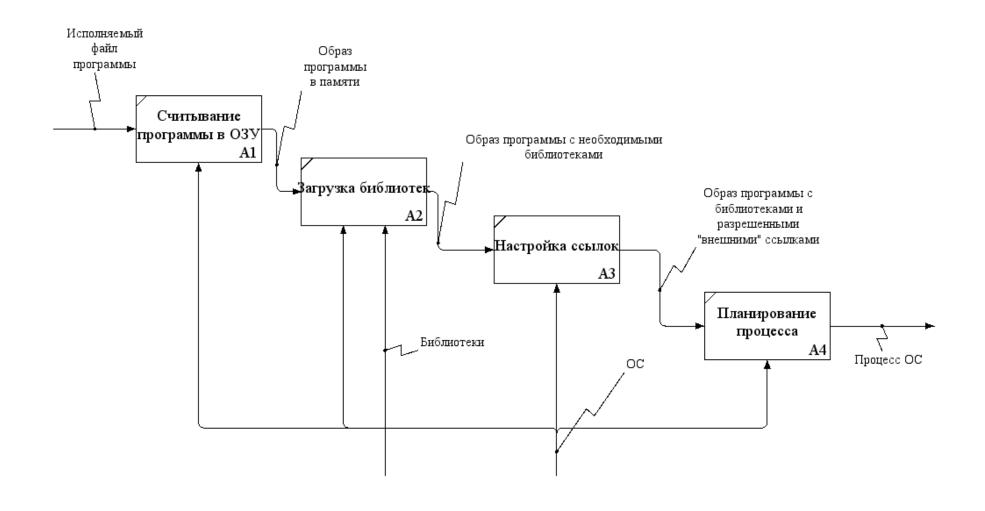
Представление о формате исполняемого файла

Заголовок 1
•••
Заголовок N
Секция .text
Секция .data
Секция .rodata
Секция .bss
Таблица импорта
• • •

Запуск программы



Запуск программы



Абстрактная память и процесс.

