Task2. Patching

Способ 1.
Вывод функции bubble_sort:

```
incomprehensible@incomprehensible:~/OxygenSoftw$ vim task2.c
incomprehensible@incomprehensible:~/OxygenSoftw$ gcc -g -O0 task2.c -o task2
incomprehensible@incomprehensible:~/OxygenSoftw$ ./task2
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After sorting: -500 -100 -50 -5 0 0 1 1 90 234 700 12345
incomprehensible@incomprehensible:~/OxygenSoftw$ arm-linux-gnueabi-gcc -g task2.c -o task2
incomprehensible@incomprehensible:~/OxygenSoftw$ ls
exp  task1  task1.c  task2  task2.c  task2_nobubble.c  Tasks
```

Импортирую бинарник task2 в Ghidra в формате Raw binary.
Нахожу функцию bubble_sort и ее адрес.

```
                    FUN_000005a0
000005a0 00 48 2d e9    stmdb       sp!,{ r11 lr }
000005a4 04 b0 8d e2    add         r11,sp,#0x4
000005a8 18 d0 4d e2    sub         sp,sp,#0x18
000005ac 18 00 0b e5    str         r0,[r11,#local_1c]
000005b0 1c 10 0b e5    str         r1,[r11,#local_20]
000005b4 00 30 a0 e3    mov         r3,#0x0
000005b8 14 30 0b e5    str         r3,[r11,#local_18]
000005bc 34 00 00 ea    b           LAB_00000694

                    LAB_000005c0                            XREF[1]:     000006
000005c0 00 30 a0 e3    mov         r3,#0x0
000005c4 10 30 0b e5    str         r3,[r11,#local_14]
000005c8 27 00 00 ea    b           LAB_0000066c

                    LAB_000005cc                            XREF[1]:     000006
000005cc 10 30 1b e5    ldr         r3,[r11,#local_14]
```

А также функцию multiply_by8 на замену.

```
                    FUN_000004cc                            XREF[1]:     FUN_
000004cc 00 48 2d e9    stmdb       sp!,{ r11 lr }
000004d0 04 b0 8d e2    add         r11,sp,#0x4
000004d4 10 d0 4d e2    sub         sp,sp,#0x10
000004d8 10 00 0b e5    str         r0,[r11,#local_14]
000004dc 14 10 0b e5    str         r1,[r11,#local_18]
000004e0 00 30 a0 e3    mov         r3,#0x0
000004e4 0c 30 0b e5    str         r3,[r11,#local_10]
000004e8 0d 00 00 ea    b           LAB_00000524

                    LAB_000004ec                            XREF[1]:     0000
000004ec 0c 30 1b e5    ldr         r3,[r11,#local_10]
000004f0 03 31 a0 e1    mov         r3,r3, lsl #0x2
000004f4 10 20 1b e5    ldr         r2,[r11,#local_14]
000004f8 03 30 82 e0    add         r3,r2,r3
000004fc 00 20 93 e5    ldr         r2,[r3,#0x0]
00000500 0c 30 1b e5    ldr         r3,[r11,#local_10]
00000504 03 31 a0 e1    mov         r3,r3, lsl #0x2
00000508 10 10 1b e5    ldr         r1,[r11,#local_14]
0000050c 03 30 81 e0    add         r3,r1,r3
```

Нахожу функцию main.

Нахожу инструкцию, где вызывается функция bubble_sort и дизассемблирую.

```
eb      bl      FUN_000003c4
e2      sub     r3,r11,#0x38
e5      ldr     r1,[r11,#local_40]
e1      cpy     r0,r3
eb      bl      0x000005a0
e3      mov     r3,#0x0
e5      ldr     r2,[DAT_000007e8]
e5      ldr     r1,[r2,#0x0]=>DAT_000108d4
e5      ldr     r2,[r11,#local_c]
```

Вставляю в операнд инструкции перехода bl адрес первой инструкции функции multiply_by8. Готово.

Полученный бинарник сохраняю как task2.bin.

Копирую пропатченную программу на Raspberry pi и тестирую (new_task2).

```
pi@raspberrypi:~ $ ls
BigBro  abbro  bench  new_task2
pi@raspberrypi:~ $ chmod +x new_task2
pi@raspberrypi:~ $ ./new_task2
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After multiplying: 8 720 -4000 0 0 8 1872 98760 5600 -400 -40 -800
pi@raspberrypi:~ $
```

Отработала multiply_by8.

Способ 2.

Я выделила свои функции в обоих исходниках в отдельную секцию. Назвала ее .patch.

```c
#include <stdio.h>


void bubble_sort() __attribute__((section (".patch")));

void bubble_sort(int *arr, int sz)
{
        int tmp;

        for (int i = 0; i < sz; ++i)
        {
                for (int j = 0; j<sz-i-1; ++j)
                {
                        if (arr[j] > arr[j + 1])
                        {
                                tmp = arr[j];
                                arr[j] = arr[j+1];
                                arr[j+1] = tmp;
                        }
                }
        }
}

int main()
{
        int arr[] = { 1, 90, -500, 0, 0, 1, 234, 12345, 700, -50, -5, -100};
        int sz = sizeof(arr) / sizeof(int);

        printf("Before routine: ");
        for (int i = 0; i < sz; ++i)
                printf("%d ", arr[i]);

        putchar('\n');

        bubble_sort(&arr[0], sz);

        printf("After routine: ");
        for (int i = 0; i < sz; ++i)
                printf("%d ", arr[i]);

        putchar('\n');
        return 0;
}
~
```

```c
#include <stdio.h>


void multiply_by8() __attribute__((section (".patch")));

__attribute__ ((noinline))
void multiply_by8(int *arr, int sz)
{
        for (int i = 0; i < sz; ++i)
                arr[i] *= 8;
}

int main()
{
        int arr[] = { 1, 90, -500, 0, 0, 1, 234, 12345, 700, -50, -5, -100};
        int sz = sizeof(arr) / sizeof(int);

        printf("Before routine: ");
        for (int i = 0; i < sz; ++i)
                printf("%d ", arr[i]);

        putchar('\n');

        multiply_by8(&arr[0], sz);

        printf("After routine: ");
        for (int i = 0; i < sz; ++i)
                printf("%d ", arr[i]);

        putchar('\n');
        return 0;
}
~
```

У меня есть два executable файла.
Их вывод на Raspberry pi:

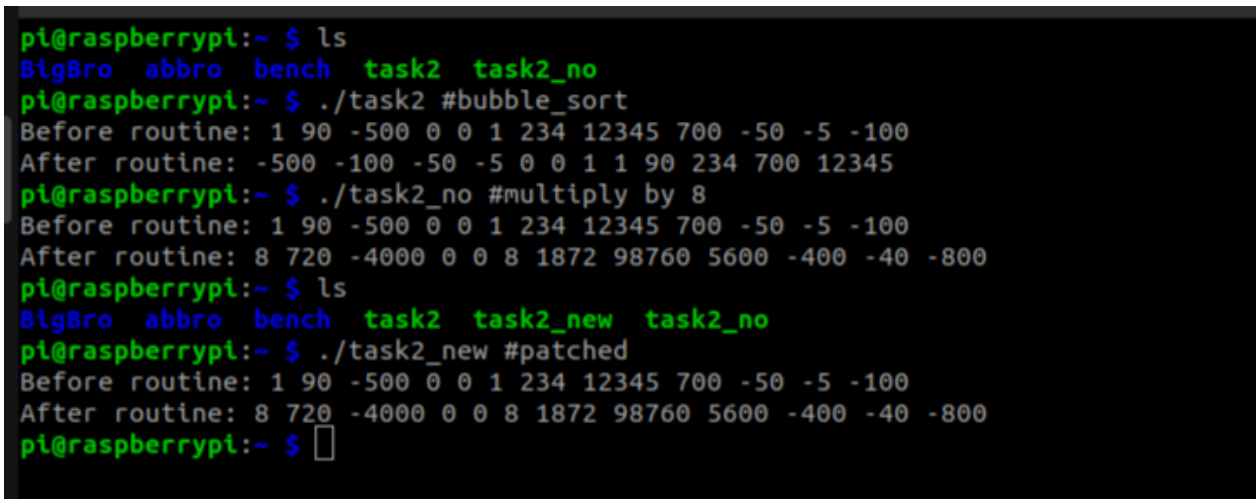
```
pi@raspberrypi:~ $ ls
BigBro  abbro  bench  task2  task2_no
pi@raspberrypi:~ $ ./task2 #bubble_sort
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After routine: -500 -100 -50 -5 0 0 1 1 90 234 700 12345
pi@raspberrypi:~ $ ./task2_no #multiply by 8
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After routine: 8 720 -4000 0 0 8 1872 98760 5600 -400 -40 -800
pi@raspberrypi:~ $
```

Теперь я могу вытащить секцию .patch из нужного мне бинарника (task2_no), который содержит multiply_by8. Сохраняю ее как patch.bin.

И заменяю секцию task2, содержащего пузырьковую сортировку, на patch.bin. При тестировании я не указывала новый файл в output и заменила сам executable task2. Но в готовой версии создала новый файл task2_new.

```
incomprehensible@incomprehensible: ~/Courses/IP...   incomprehensible@incomprehensible: ~/OxygenSoft...
incomprehensible@incomprehensible:~/OxygenSoftw/exp$ ls
task2  task2.c  task2_no  task2_nobubble.c  task2_no_sections  task2_sections
incomprehensible@incomprehensible:~/OxygenSoftw/exp$ arm-linux-gnueabi-objcopy -j .patch -Obinary task2_no patch.bin
incomprehensible@incomprehensible:~/OxygenSoftw/exp$ arm-linux-gnueabi-objcopy --update-section .patch=patch.bin task2
incomprehensible@incomprehensible:~/OxygenSoftw/exp$ scp task2 pi@192.168.1.40:/home/pi/task2_new
pi@192.168.1.40's password:
task2
incomprehensible@incomprehensible:~/OxygenSoftw/exp$
```

Переношу исполняемый файл на свою Raspberry pi для быстрой проверки (task2_new).



Видим, что замена произошла.