

# Operating Systems. Homework 5

[Operating Systems](#)

Александр Васюков | БПИ235

## Задание

Необходимо написать две программы – передатчик и приемник, которые будут запускаться с разных терминалов. Каждая программа вначале печатает свой PID и запрашивает ввод с клавиатуры PID программы-собеседника (вводится пользователем). Порядок ввода PID не имеет значения.

Программа-передатчик запрашивает с клавиатуры ввод целого десятичного числа (число может быть и отрицательным!). Введенное число побитово передается программе-приемнику с использованием пользовательских сигналов SIGUSR1 и SIGUSR2. Программа-приемник после завершения приема печатает принятое число в десятичной системе счисления.

## Решение

Программа на Github: <https://github.com/vasyukov1/HSE-FCS-SE-2-year/tree/main/Operating%20Systems/Homeworks/05>

### Sender

```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>

pid_t receiver_pid;
volatile sig_atomic_t ready = 1;

void handler(int sig) {
    ready = 1;
}

void send_bit(int bit) {
    int signal = (bit == 0) ? SIGUSR1 : SIGUSR2;
    kill(receiver_pid, signal);
    ready = 0;
    while (!ready);
}

void send_number(int num) {
    for (int i = 0; i < 32; i++) {
        send_bit((num >> i) & 1);
    }
}
```

```

        kill(receiver_pid, SIGINT);
    }

    int main() {
        printf("Sender PID: %d\n", getpid());

        printf("Enter receiver PID: ");
        scanf("%d", &receiver_pid);

        struct sigaction sa;
        sa.sa_handler = handler;
        sa.sa_flags = SA_RESTART;
        sigaction(SIGUSR1, &sa, NULL);
        sigaction(SIGUSR2, &sa, NULL);

        int num;
        printf("Enter an integer: ");
        scanf("%d", &num);

        send_number(num);
        printf("Transmission complete.\n");
        return 0;
    }

```

Функция `send_bit` отправляет 1 бит, ожидая сигнал готовности.

Функция `send_number` отправляет всё число, используя предыдущую функцию.

Основная программа выводит PID и просит ввести PID получателя. После этого устанавливает сигналы и просит ввести число, которое надо передать. После передачи сообщает, что она завершена.

## Receiver

```

#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <math.h>

pid_t sender_pid;
volatile sig_atomic_t received_bits[32];
volatile sig_atomic_t counter = 0;

void handler(int sig) {
    received_bits[counter] = (sig == SIGUSR1) ? 0 : 1;
    ++counter;
    kill(sender_pid, SIGUSR1);
}

void end_handler(int sig) {

```

```

    int number = 0;
    for (int i = 0; i < counter; i++) {
        number |= (received_bits[i] << i);
    }
    printf("\nReceived number: %d\n", number);
    exit(0);
}

int main() {
    printf("Receiver PID: %d\n", getpid());

    printf("Enter sender PID: ");
    scanf("%d", &sender_pid);

    struct sigaction sa;
    sa.sa_flags = SA_RESTART;
    sa.sa_handler = handler;
    sigaction(SIGUSR1, &sa, NULL);
    sigaction(SIGUSR2, &sa, NULL);

    struct sigaction sa_end;
    sa_end.sa_flags = SA_RESTART;
    sa_end.sa_handler = end_handler;
    sigaction(SIGINT, &sa_end, NULL);

    printf("Waiting...\n");
    while (1) {
        pause();
    }
    return 0;
}

```

Функция `handler` получает биты передаваемого числа.

Функция `end_handler` восстанавливает число из битов и выводит его.

В основной программе печатается PID и просится ввести PID отправителя. После этого устанавливаются сигналы и ожидается число от отправителя.

## Пример работы

1. Запускаем `receiver` и `sender`.
2. Вводим PID отправителя для `receiver`.
3. Вводим PID получателя для `sender`.
4. Вводим передаваемое число.
5. В терминале `receiver` появляется число.

## Пример передачи числа 10 и -124:

```
alexvasyukov@Alexanders-MacBook-Air 05 % gcc receiver.c -o receiver
alexvasyukov@Alexanders-MacBook-Air 05 % ./receiver
Receiver PID: 91514
Enter sender PID: 91510
Waiting...

Received number: 10
alexvasyukov@Alexanders-MacBook-Air 05 % ./receiver
Receiver PID: 91554
Enter sender PID: 91550
Waiting...

Received number: -124
alexvasyukov@Alexanders-MacBook-Air 05 %
```

```
alexvasyukov@Alexanders-MacBook-Air 05 % gcc sender.c -o sender
alexvasyukov@Alexanders-MacBook-Air 05 % ./sender
Sender PID: 91510
Enter receiver PID: 91514
Enter an integer: 10
Transmission complete.
alexvasyukov@Alexanders-MacBook-Air 05 % ./sender
Sender PID: 91550
Enter receiver PID: 91554
Enter an integer: -124
Transmission complete.
alexvasyukov@Alexanders-MacBook-Air 05 %
```