

Операционные системы
ДЗ 5. Отчёт
Работа на 10 баллов

Фролов-Буканов Виктор Дмитриевич БПИ-228

5 марта 2024

1 Коды программы (отправитель и получатель)

sender.cpp

```
#include <iostream>
#include <signal.h>
#include <unistd.h>

void custom_handler(int nsig) {
}

int main() {
    signal(SIGUSR1, custom_handler);
    pid_t pid = getpid();
    std::cout << "SENDER\n";
    std::cout << "My_pid_is_" << pid << '\n';
    pid_t receiver_pid;
    std::cout << "Enter_receiver_id: ";
    std::cin >> receiver_pid;
    std::cout << "Enter_number: ";
    int num;
    std::cin >> num;
    std::cout << "\n";

    for (int i = 0; i < 32; ++i) {
        int bit = num & 1;
        if (!bit) {
            kill(receiver_pid, SIGUSR1);
        } else {
            kill(receiver_pid, SIGUSR2);
        }
        num >>= 1;
        pause(); // waiting for signal (method from unistd.h)
    }

    kill(receiver_pid, SIGTERM);
    return 0;
}
```

receiver.cpp

```
#include <iostream>
#include <unistd.h>
#include <signal.h>
#include <vector>
#include <algorithm>

bool toBreak = false;
std::vector<int> bits;

void my_build_number_handler(int nsig) {
    if (nsig == SIGUSR2) {
        bits.push_back(1);
    } else {
        bits.push_back(0);
    }
}
```

```

}

void my_terminate_handler(int nsig) {
    toBreak = true;
}

int main() {
    bits.reserve(32);
    signal(SIGUSR1, my_build_number_handler);
    signal(SIGUSR2, my_build_number_handler);
    signal(SIGTERM, my_terminate_handler);
    pid_t pid = getpid();
    std::cout << "RECEIVER\n";
    std::cout << "My_pid_is_" << pid;
    pid_t sender_pid;
    std::cout << "\nEnter_sender_id: ";
    std::cin >> sender_pid;
    while (!toBreak) {
        kill(sender_pid, SIGUSR1);
        pause();
    }

    std::reverse(bits.begin(), bits.end());
    std::cout << "Number_has_been_received_";
    std::cout << "Bits_of_the_number_are_";
    for (auto i = 0; i < 32; ++i) {
        std::cout << bits[i];
    }
    std::cout << '\n';
    int multiplier = 1;
    int ans = 0;
    int mask = 0;
    if (bits[0] == 1) mask = 1;

    for (int i = 31; i > 0; --i) {
        ans += (bits[i] ^ mask) * multiplier;
        multiplier *= 2;
    }

    if (mask) ans = -ans - 1;
    std::cout << "Value_of_the_number_is_" << ans << '\n';

    return 0;
}

```

2 Результат работы программы

```
frolovbuk@LAPTOP-KØRFAB50:~/hw5$ g++ -o r.out receiver.cpp
frolovbuk@LAPTOP-KØRFAB50:~/hw5$ g++ -o s.out sender.cpp
frolovbuk@LAPTOP-KØRFAB50:~/hw5$ ./s.out
SENDER
My pid is 1310
Enter receiver id: 1311
Enter number: -1246781
```

Figure 1: 1 скрин

```
frolovbuk@LAPTOP-KØRFAB50:~/hw5$ ./r.out
RECEIVER
My pid is 1311
Enter sender id: 1310
Number has been received. Bits of the number are 1111111111011001111100111000011
Value of the number is -1246781
```

Figure 2: 2 скрин

```
frolovbuk@LAPTOP-KØRFAB50:~/hw5$ ./s.out
SENDER
My pid is 1328
Enter receiver id: 1329
Enter number: 12445
```

Figure 3: 3 скрин

```
frolovbuk@LAPTOP-KØRFAB50:~/hw5$ ./r.out
RECEIVER
My pid is 1329
Enter sender id: 1328
Number has been received. Bits of the number are 0000000000000000011000010011101
Value of the number is 12445
```

Figure 4: 4 скрин

3 Пару замечаний

1. Программа корректно работает как на отрицательных, так и на положительных числах
2. Порядок ввода данных в обе программы не важен, в любом случае передача числа будет осуществляться корректно с правильным выводом бит
3. Для индикации окончания передачи числа с отправителя я использую сигнал ***SIGTERM***, который обрабатывается в получателе, и в итоге программа выходит из бесконечного цикла, в котором она оповещала программу sender о том, что бит был получен