# Лабораторная работа 8.

## Сигналы

## Задания

**Задание 1**. Создайте два обработчика сигнала SIGALRM.

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

using namespace std;


time_t curtime;
struct tm * loctime;


void sigalrmHandler_1(int sig)
{
    cout << "First Handler" << endl;
    curtime = time(0);
    loctime = localtime(&curtime);
    fputs(asctime(loctime), stdout);
    cout << "\n";
}


void sigalrmHandler_2(int sig)
{
    cout << "Second Handler" << endl;
    curtime = time(0);
    loctime = localtime(&curtime);
    fputs(asctime(loctime), stdout);
    cout << "\n";
}
```

```
int main()
{
    int i = 0;

    while (i++ < 10)
    {
        if (i % 2 == 0)
        {
            signal(SIGALRM, sigalrmHandler_2);
        }

        else
        {
            signal(SIGALRM, sigalrmHandler_1);
        }

        raise(SIGALRM);
        sleep(1);
    }
}
```

```
First Handler
Tue Sep 28 07:15:02 2021

Second Handler
Tue Sep 28 07:15:03 2021

First Handler
Tue Sep 28 07:15:04 2021

Second Handler
Tue Sep 28 07:15:05 2021

First Handler
Tue Sep 28 07:15:06 2021

Second Handler
Tue Sep 28 07:15:07 2021

First Handler
Tue Sep 28 07:15:08 2021

Second Handler
Tue Sep 28 07:15:09 2021

First Handler
Tue Sep 28 07:15:10 2021

Second Handler
Tue Sep 28 07:15:11 2021
```

**Задание** 2. Измените последнюю программу так, чтобы она выдавала не время и дату, а счет в игре двух человек. Первый обработчик играет за первого человека (условно говоря). Второй обработчик играет за второго человека. Первый обработчик добавляет случайное число к общей переменной, второй – отнимает. Играют до 10 раз. Если в конце игры на счету будет число больше нуля – то выиграл первый игрок, иначе выиграл второй игрок.

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

using namespace std;


time_t curtime;
struct tm * loctime;

int score;


void sigalrmHandler_1(int sig)
{
    srand(time(NULL)); // seed

    cout << "First Player" << endl;

    score += rand() % 100;

    cout << score << endl;
}
```

```
void sigalrmHandler_2(int sig)
{
    cout << "Second Player" << endl;

    srand(time(NULL)); // seed

    score -= rand() % 100;

    cout << score << endl;
}


int main()
{
    score = 0;

    for(int i = 0; i < 10; i++)
    {
        i % 2 == 0
        ? signal(SIGALRM, sigalrmHandler_2)
            : signal(SIGALRM, sigalrmHandler_1);

        raise(SIGALRM);
        sleep(1);
    }

    if (score > 0)
    {
        cout << "First player Win!" << endl;
    }

    else if (score < 0)
    {
        cout << "Second player Win!"<< endl;
    }
```

```
    else
    {
        cout << "Draw"<< endl;
    }
}
```

```
Second Player
-96
First Player
-31
Second Player
-97
First Player
-68
Second Player
-96
First Player
-67
Second Player
-137
First Player
-92
Second Player
-103
First Player
-67
Second player Win!
```

**Задание**. Представим, что есть клиент банка, который справляется о состоянии своего счета. Клиент посылает запрос (сигнал), банк возвращает ответ (сумма).

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

using namespace std;

double progress(double sum);

void user(int sig);
void bank(int sig);

double score;

int main()
{
    score = 50.0;

    (void) signal (SIGUSR1, user);
    raise(SIGUSR1);

    while(1)
        sleep(1);
}

void user(int sig)
{
    cout << "User send request! " << score << "\n" << endl;
    sleep(1);

    (void) signal(SIGUSR2, bank);
    raise(SIGUSR2);
}
```

```cpp
void bank(int sig)
{
    score = progress(score);
    cout << "Bank return response! " << score << "\n" << endl;
    sleep(1);

    (void) signal(SIGUSR1, user);
    raise(SIGUSR1);
}

double progress(double sum)
{
    return sum + 0.1 * sum;
}
```

```
User send request! 50

Bank return response! 55

User send request! 55

Bank return response! 60.5

User send request! 60.5

Bank return response! 66.55

User send request! 66.55

Bank return response! 73.205

User send request! 73.205

Bank return response! 80.5255

User send request! 80.5255

Bank return response! 88.578

User send request! 88.578

Bank return response! 97.4359

User send request! 97.4359
```