

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»
(НИЯУ МИФИ)

Институт Интеллектуальных Кибернетических Систем
Кафедра Кибернетики

Лабораторная работа 2(В)

Вариант 12

Работу выполнил:
студент группы Б14-501
Белозёров Александр Александрович
Проверил: Ктитров Сергей Викторович

МОСКВА 2017 г.

Задание

Разработать программу, передающую с использованием неименованного канала коэффициенты квадратного уравнения дочернему процессу. Дочерний процесс решает уравнение и возвращает решение через неименованный канал.

ЛИСТИНГ

```
qolver.c Makefile
all: qolver

qolver: qolver.o
| g++ qolver.o -o qolver

qolver.o: qolver.c
| g++ -c qolver.c

clean:
| rm -rf *.o qolver

qolver.c Makefile
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <math.h>

int solver(float* a, float* b, float* c)
{
    float x1, x2;

    float d=(*b**b)-(4**a**c);
    if(d>=0){
        x1=(-*b+sqrt(d))/(2**a);
        x2=(-*b-sqrt(d))/(2**a);
        *a=x1;
        *b=x2;
        *c=0;
    }else{
        x1=(-*b/(2**a));
        x2=(sqrt(-d)/(2**a));
        *a=x1;
        *b=x2;
        *c=1;
    }
    return 0;
}

int main(int argc, char *argv[])
{
    int fd[2];
    int state = pipe(fd);
    if (state == -1) printf("Piping error\n");

    pid_t p;
    switch ( p = fork() )
    {
        {
            case -1:
                printf("error\n");
                break;

            case 0:
                float buffer[3];
                read(fd[0], buffer, sizeof(buffer));
                close(fd[0]);

                solver(&buffer[0], &buffer[1], &buffer[2]);
        }
    }
}
```

```

write(fd[1], &buffer[0], sizeof(buffer));
close(fd[1]);

exit(EXIT_SUCCESS);
break;

default:
    printf("child process' PID = %d\n", p);

    float coeff[3];
    for (int i = 0; i < 3; i++) coeff[i] = atof(argv[i+1]);
    write(fd[1], &coeff[0], sizeof(coeff));
    close(fd[1]);

    int status;
    while(wait(&status) == -1);

    read(fd[0], coeff, sizeof(coeff));
    if (coeff[2]) printf("Imaginary roots are %f - %f and %f + %f\n", coeff[0], coeff[1], coeff[0], coeff[1]);
    else printf("Roots are %f and %f\n", coeff[0], coeff[1]);
    close(fd[0]);

    exit(EXIT_SUCCESS);
}
return 0;

```

Концепция

В данной программе:

- с использованием функциональности библиотеки `unistd.h`:
 - создается дубликат текущего процесса
 - создается неименованный канал для последующей двусторонней передачи данных
- родительский процесс:
 - записывает данные из параметров вызова в канал
 - ждет завершения работы клонированного процесса (`sys/wait.h`)
- клонированный процесс
 - считывает данные из канала
 - используя их как коэффициенты квадратного уравнения, ищет его корни (`math.h`)
 - записывает их в канал и завершает выполнение
- после чего родительский процесс
 - читает данные из канала
 - демонстрирует решение уравнения