

Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Лабораторная работа
№2

«Процессы и асинхронное взаимодействие»

Выполнил: студент 4 курса

ИВТ, гр. ИП-813

Бурдуковский И.А.

Выполнил с: Стояк Ю.К.

Проверил:

преподаватель кафедры ПМиК

Белевцова Екатерина Андреевна

Новосибирск, 2021 г.

Оглавление

Задание на лабораторную работу	3
Выполнение	4
Вывод	12

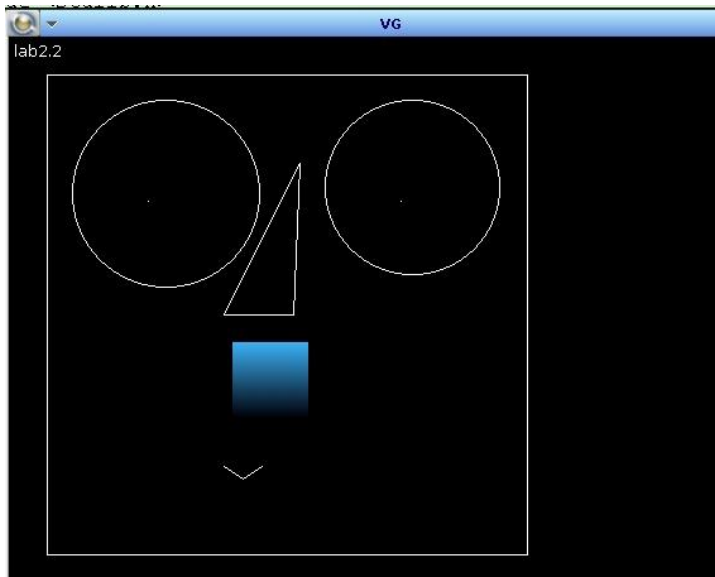
Задание на лабораторную работу

1. Тщательно изучить библиотеку VinGraph.
2. Используя функции библиотеки VinGraph, нарисовать абстрактную картину, которой представлены (почти) все доступные графические элементы.
3. Заставить нарисованные элементы двигаться независимо друг от друга с помощью параллельных процессов (можно изменять во времени положение, цвет, размеры, конфигурацию графических элементов). Предусмотреть завершение программы по нажатию на любую клавишу.
4. Нарисовать нечто, движущееся по замкнутой кривой. Организовать изменение траектории движения по нажатию на клавиши (организуя взаимодействие процессов через общую область памяти (shared memory)). В качестве фона можно использовать (оживленную) картину, созданную на предыдущих этапах работы.
5. Затем последнюю программу сделать с помощью нитей в одном процессе.

Выполнение

- 1) Изучена библиотека VinGraph на основе справочного материала с эиоса. Ознакомлены с основными фигурами в VinGraph и функциями.
- 2) Реализована программа, выводящая в терминал VinGraph почти все доступные графические элементы.

Выполнение программы:



Код программы:

```
#include <stdlib.h>
#include <unistd.h>
#include <vingraph.h>

int main()
{
    ConnectGraph();
    Text(2, 2, "lab2.2");
    Rect(30, 30, 380, 380);
    Ellipse(50, 50, 150, 150);
    Ellipse(250, 50, 140, 140);
    Pixel(110, 130);
    Pixel(310, 130);

    tPoint p1 [] = {{170,220}, {230, 100}, {225,220}};
    Polygon(p1,3);

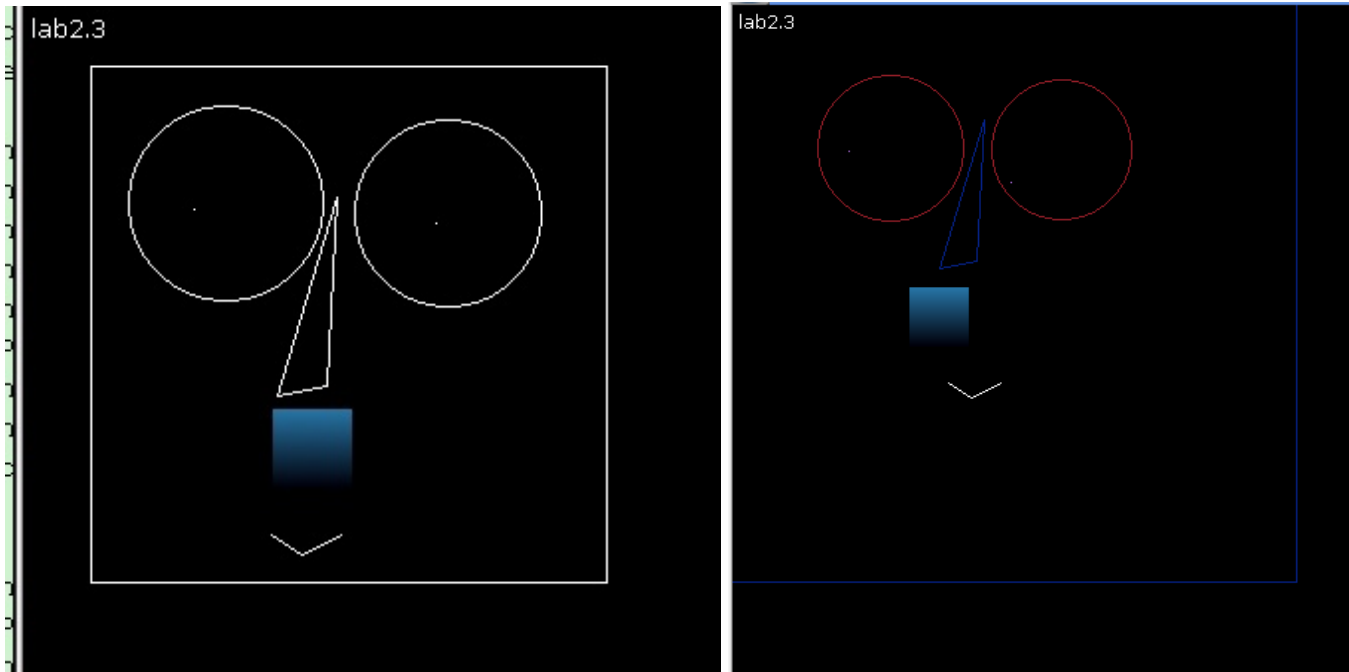
    int *im_buf = (int*)malloc(60*60*4);
    for (int i = 59, c = 10; i >= 0; i--, c += 0x010304)|
        for(int j = 0; j < 60; j++)
            im_buf[60*i + j] = c;
    Image32(175, 240, 60, 60, im_buf);

    tPoint p2 [] = {{170, 340}, {185, 350}, {200, 340}};
    Polyline(p2, 3);

    sleep(10);
    CloseGraph();
    return 0;
}
```

3) Для реализации этого задания мы брали все фигуры из прошлого. Они все двигаются независимо друг от друга в параллельных процессах. Для создания процессов используется функция `fork()`

Результат работы программы:



Код программы:

```
#include <stdlib.h>
#include <unistd.h>
#include <vingraph.h>
#include <iostream>
#include <stdio.h>
#include <sys/types.h>
#include <process.h>
#include <time.h>

int main()
{
    ConnectGraph();
    Text(2, 2, "lab2.3");

    int rect = Rect(30, 30, 260, 260);
    int elip1 = Ellipse(50, 50, 100, 100);
    int elip2 = Ellipse(170, 50, 95, 95);
    int pix1 = Pixel(75, 100);
    int pix2 = Pixel(210, 100);
    tPoint p1 [] = {{130,190}, {160, 90}, {155,185}};
    int polyg = Polygon(p1,3);
    int *im_buf = (int*)malloc(40*40*4);
    for (int i = 39, c = 10; i >= 0; i--, c += 0x010304)
        for(int j = 0; j < 40; j++)
            im_buf[40*i + j] = c;
    int img = Image32(120, 200, 40, 40, im_buf);
    tPoint p2 [] = {{130, 260}, {145, 270}, {165, 260}};
    int poly1 = Polyline(p2, 3);

    pid_t proc1, proc2, proc3, proc4;

    int a = getpid();
    printf("\ncurent process = %d",a);
```

```

if(proc1 = fork())
{
    int b = getpid();
    printf("\ncurent process 1 = %d", b);
    if(proc2 = fork())
    {
        int c = getpid();
        printf("\ncurent process 2 = %d", c);
        if(proc3 = fork())
        {
            int d = getpid();
            printf("\ncurent process 3 = %d", d);
            if(proc4 = fork())
            {
                int e = getpid();
                printf("\ncurent process 4 = %d\n", e);
                InputChar();
                CloseGraph();
            }
            else
            {
                srand(time(0));
                while(true)
                {
                    int c = RGB(rand()% 255, rand()% 255,rand()% 255);
                    int x = (rand()% 3)-1;
                    int y = (rand()% 3)-1;
                    Move(pix1,x,y);
                    SetColor(pix1,c);
                    x = (rand()% 3)-1;
                    y = (rand()% 3)-1;
                    Move(pix2,x,y);
                    SetColor(pix2,c);
                    delay(100);
                }
            }
        }
    }
}

```

```

    }
}
else
{
    srand(time(0));
    while(true)
    {
        int c = RGB(rand()% 255, rand()% 255,rand()% 255);
        int x = (rand()% 3)-1;
        int y = (rand()% 3)-1;
        Move(elip1,x,y);
        SetColor(elip1,c);
        x = (rand()% 3)-1;
        y = (rand()% 3)-1;
        Move(elip2,x,y);
        SetColor(elip2,c);
        delay(200);
    }
}
else
{
    srand(time(0));
    while(true)
    {
        int c = RGB(rand()% 255, rand()% 255,rand()% 255);
        int lr = (rand()%60)-30;
        int x = (rand()% 3)-1;
        int y = (rand()% 3)-1;
        Move(rect,x,y);
        SetColor(rect,c);
        Enlarge(rect,lr,lr);
        x = (rand()% 3)-1;
        y = (rand()% 3)-1;
        Move(polyg,x,y);
    }
}
}

```

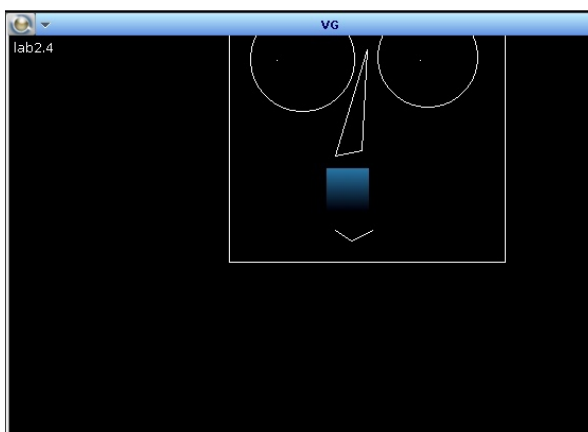
```

        SetColor(rect,c);
        Enlarge(rect,lr,lr);
        x = (rand()% 3)-1;
        y = (rand()% 3)-1;
        Move(polyg,x,y);
        SetColor(polyg,c);
        delay(300);
    }
}
else
{
    srand(time(0));
    while(true)
    {
        int x = (rand()% 3)-1;
        int y = (rand()% 3)-1;
        Move(img,x,y);
        x = (rand()% 3)-1;
        y = (rand()% 3)-1;
        Move(polyl,x,y);
        delay(400);
    }
}
CloseGraph();
return 0;
}

```

4) Был нарисован квадрат, движущийся по замкнутой кривой. На его движение можно влиять с помощью клавиш: W, A, S, D. Взаимодействие процессов реализовано через shared memory.

Результат работы программы:



Код программы:

```
#include <stdlib.h>
#include <unistd.h>
#include <vingraph.h>
#include <iostream>
#include <stdio.h>
#include <sys/types.h>
#include <process.h>
#include <time.h>
#include <sys/mman.h>
#include <math.h>

int main()
{
    ConnectGraph();
    int key = 0, x = 0, y = 0;
    pid_t proc;
    float phi = 0, rho = 0;
    float *a = static_cast<float*>(mmap(0, sizeof* a, PROT_READ + PROT_WRITE, MAP_SHARED + MAP_ANON, -1, 0));
    float *b = static_cast<float*>(mmap(0, sizeof* b, PROT_READ + PROT_WRITE, MAP_SHARED + MAP_ANON, -1, 0));

    Text(2, 2, "lab2.4");
    int pic = Picture(10, 10);
    int color = RGB(255, 255, 255);
    Rect(30, 30, 260, 260, 0, color, pic);
    Ellipse(50, 50, 100, 100, color, pic);
    Ellipse(170, 50, 95, 95, color, pic);
    Pixel(75, 100, color, pic);
    Pixel(210, 100, color, pic);
    tPoint p1 [] = {{130, 190}, {160, 90}, {155, 185}};
    Polygon(p1, 3, color, pic);
    int *im_buf = (int*)malloc(40*40*4);

    for (int i = 39, c = 10; i >= 0; i--, c += 0x010304)
        for(int j = 0; j < 40; j++)
            im_buf[40*i + j] = c;
    Image32(120, 200, 40, 40, im_buf, pic);
    tPoint p2 [] = {{130, 260}, {145, 270}, {165, 260}};
    Polyline(p2, 3, color, pic);

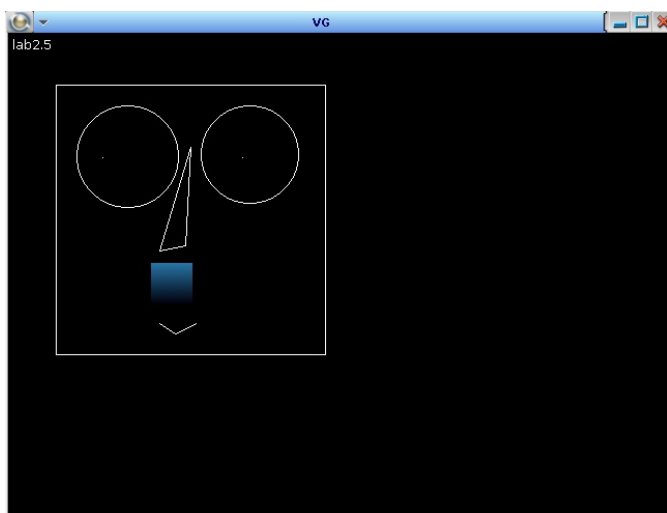
    *a = 100;
    *b = 100;
    if(proc = fork()){
        while(key != 13){
            key = InputChar();
            printf("%d\n", key);
            if (key == 'w')
                *a = *a + 10;
            if (key == 'a')
                *b = *b + 10;
            if (key == 's')
                *a = *a - 10;
            if (key == 'd')
                *b = *b - 10;;
        }
        CloseGraph();
    }
}
```



```
else{
    while(true){
        rho = *a * cos(phi) + *b;
        x = rho * cos(phi) + 20;
        y = rho * sin(phi) + 20;
        phi += 0.01;
        MoveTo(x, y, pic);
        delay(15);
    }
}
return 0;
```

5) Предыдущая программа была преобразована для работы с нитями из библиотеки `<pthread.h>`.

Результат работы программы:



Код программы:

```
#include <stdlib.h>
#include <unistd.h>
#include <vingraph.h>
#include <iostream>
#include <stdio.h>
#include <sys/types.h>
#include <pthread.h>
#include <time.h>
#include <sys/mman.h>
#include <math.h>

float *b;
float *a;
int x = 0, y = 0, pic;
float phi = 0, rho = 0;

void *pic_move(void *args)
{
    int key = 0;
    while(key != 13)
    {
        key = InputChar();
        printf("%d\n", key);
        if (key == 'w')
            *a = *a + 10;
        if (key == 'a')
            *b = *b + 10;
        if (key == 's')
            *a = *a - 10;
        if (key == 'd')
            *b = *b - 10;;
    }
}

*pic_draw(void *args)
while(true)
{
    rho = *a * cos(phi) + *b;
    x = rho * cos(phi) + 20;
    y = rho * sin(phi) + 20;
    phi += 0.01;
    MoveTo(x, y, pic);
    delay(15);
}

main()

ConnectGraph();
a = static_cast<float*>(mmap(0, sizeof* a, PROT_READ + PROT_WRITE, MAP_SHARED + MAP_ANON, -1,
b = static_cast<float*>(mmap(0, sizeof* b, PROT_READ + PROT_WRITE, MAP_SHARED + MAP_ANON, -1,

Text(2, 2, "lab2.5");
pic = Picture(10, 10);
int color = RGB(255, 255, 255);
Rect(30, 30, 260, 260, 0, color, pic);
Ellipse(50, 50, 100, 100, color, pic);
Ellipse(170, 50, 95, 95, color, pic);
Pixel(75, 100, color, pic);
Pixel(210, 100, color, pic);
Point p1 [] = {{130,190}, {160, 90}, {155,185}};
Polygon(p1,3, color, pic);
int *im_buf = (int*)malloc(40*40*4);
for (int i = 39, c = 10; i >= 0; i--, c += 0x010304)
    for(int j = 0; j < 40; j++)
        im_buf[40*i + j] = c;
```

```

    Image32(120, 200, 40, 40, im_buf, pic);
    tPoint p2 [] = {{130, 260}, {145, 270}, {165, 260}};
    Polyline(p2, 3, color, pic);

    *a = 100;
    *b = 100;

    pthread_t mover, drawer;
    pthread_create(&mover, 0, &pic_move, NULL);
    pthread_create(&drawer, 0, &pic_draw, NULL);
    if(!pthread_join(mover, 0))
        pthread_cancel(drawer);
I   CloseGraph();
    return 0;
}

```

Вывод

В этой лабораторной работе мы узнали о VinGraph, изучили рисование основных фигур в нем, их раскраску и перерисовку. Поработали с потоками и нитями.