

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина»

Кафедра ВПМ

Отчёт
о лабораторной работе №1
по дисциплине
«Распределённые системы обработки информации»
Тема:
«Сервис-ориентированная архитектура вычислений
на основе протокола XML-RPC»

Выполнила
студентка гр. 343М
Торжкова А.О.
Проверил
доц. Князьков П.А.

Рязань 2023

Цель работы

Получение практических знаний о сервис-ориентированной архитектуре вычислений на основе протокола XML-RPC.

Выполнение

Задание 1. Загрузить дистрибутив Anaconda3 2022/Python 3.9 version.

Задание 2. Изучить программы файлов тетрадей клиента `xmlrpc_client.ipynb` и сервера `xmlrpc_server.ipynb`.

Задание 3. Переписать код тетрадей клиента `xmlrpc_client.ipynb` и сервера `xmlrpc_server.ipynb` с Python 2 на Python 3. Для примера работы с Python 3 и изображением изучить тетрадь «`gdal_tiff_mask_2022_PIL.ipynb`».

Задание 4. Дополнить функционал сервера функцией проверки на наличие человека в черном списке с проверкой по ФИО и дате рождения.

Код тетради `xmlrpc_server.ipynb` (рис. 1):

```
# Задание 4. Проверка на наличие человека в черном списке с проверкой по ФИО и дате рождения
def black_list_check_by_name_and_birthday(full_name, birthday):
    s, n, p = full_name.split()
    frame = pd.read_csv('../resources/bad_boys2.csv', header=0, sep=',', encoding='utf8')
    exist = [f[0] == s and f[1] == n and f[2] == p and f[3] == birthday for f in frame.values]
    if any(exist):
        return full_name + " (" + birthday + "): " + "bad_boy"
    else:
        return full_name + " (" + birthday + "): " + "good_boy"
server.register_function(black_list_check_by_name_and_birthday, 'black_list_check_by_name_and_birthday')
```

Рисунок 1 – Задание 4, сервер

Код тетради `xmlrpc_client.ipynb` и результат выполнения (рис. 2):

```
# Задание 4. Проверка на наличие человека в черном списке с проверкой по ФИО и дате рождения

print(server.black_list_check_by_name_and_birthday('Иванов Иван Иванович', '22.03.1989'))
print(server.black_list_check_by_name_and_birthday('Иванов Иван Иванович', '23.03.1989'))

Иванов Иван Иванович (22.03.1989): bad_boy
Иванов Иван Иванович (23.03.1989): good_boy
```

Рисунок 2 – Задание 4, клиент

Задание 5. Модифицировать функцию сервера `send_back_inversion` на корректную работу также с монохромными изображениями, т.е. входными данными могут быть размерности (M, N, 3) и (M, N, 1). Для инверсии изображения запрещено использовать библиотечные функции.

Код тетради xmlrpc_server.ipynb (рис. 3):

```
# Задание 4. Проверка на наличие человека в черном списке с проверкой по ФИО и дате рождения
def black_list_check_by_name_and_birthday(full_name, birthday):
    s, n, p = full_name.split()
    frame = pd.read_csv('../resources/bad_boys2.csv', header=0, sep=',', encoding='utf8')
    exist = [f[0] == s and f[1] == n and f[2] == p and f[3] == birthday for f in frame.values]
    if any(exist):
        return full_name + " (" + birthday + "): " + "bad_boy"
    else:
        return full_name + " (" + birthday + "): " + "good_boy"
server.register_function(black_list_check_by_name_and_birthday, 'black_list_check_by_name_and_birthday')
```

Рисунок 3 – Задание 5, сервер

Код тетради xmlrpc_client.ipynb и результат выполнения (рис. 4-5):

```
def show_modified_image(path, modify_func, param=None):
    img = Image.open(path)
    img_arr = array(img)
    pimg = pickle.dumps(img_arr)
    img_bin = rpc.Binary(pimg)
    if param is None:
        img_bin_out = modify_func(img_bin)
    else:
        img_bin_out = modify_func(img_bin, param)
    img_arr_out = pickle.loads(img_bin_out.data)
    gray()
    imshow(img_arr_out)

# Задание 5. Модифицировать функцию сервера send_back_inversion на корректную работу также с монохромными изображениями,
# т.е. входными данными могут быть размерности (M, N, 3) и (M, N, 1).

show_modified_image('../resources/Jellyfish.jpg', server.color_inversion)
```

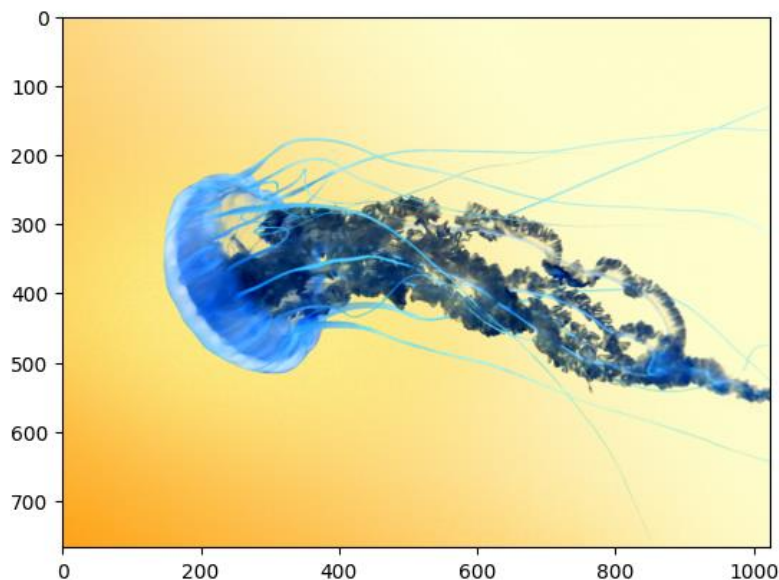


Рисунок 4 – Задание 5, клиент (цветное изображение)

```
show_modified_image('../resources/11.bmp', server.color_inversion)
```

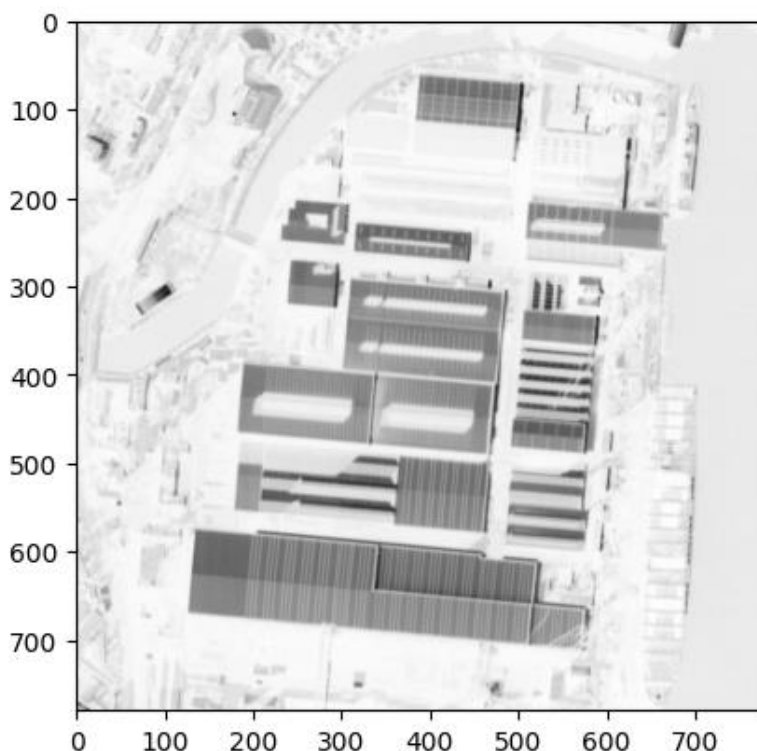


Рисунок 5 – Задание 5, клиент (монохромное изображение)

Задание 6. Дополнить функционал сервера функцией бинаризации изображения по задаваемому порогу 1-255 для цветных и монохромных изображений. Для бинаризации изображений запрещено использовать библиотечные функции.

Код тетради `xmIrpc_server.ipynb` (рис. 6):

```
# Задание 6. Функция бинаризации изображения по задаваемому порогу 1-255 для цветных и монохромных изображений
def binarize_by_threshold(bin_data, threshold):
    img_arr = pickle.loads(bin_data.data)
    height = img_arr.shape[0]
    width = img_arr.shape[1]
    for i in range(height):
        for j in range(width):
            if type(img_arr[i][j]) == numpy.ndarray:
                for p in range(3):
                    img_arr[i][j][p] = 255 if img_arr[i][j][p] >= threshold else 0
            else:
                img_arr[i][j] = 255 if img_arr[i][j] >= threshold else 0
    pimg = pickle.dumps(img_arr)
    return client.Binary(pimg)
server.register_function(binarize_by_threshold, 'binarize_by_threshold')
```

Рисунок 6 – Задание 6, сервер

Код тетради xmlrpc_client.ipynb и результат выполнения (рис. 7-9):

```
# Задание 6. Функция бинаризации изображения по задаваемому порогу 1-255 для цветных и монохромных изображений
```

```
show_modified_image('../resources/Jellyfish.jpg', server.binarize_by_threshold, 100)
```

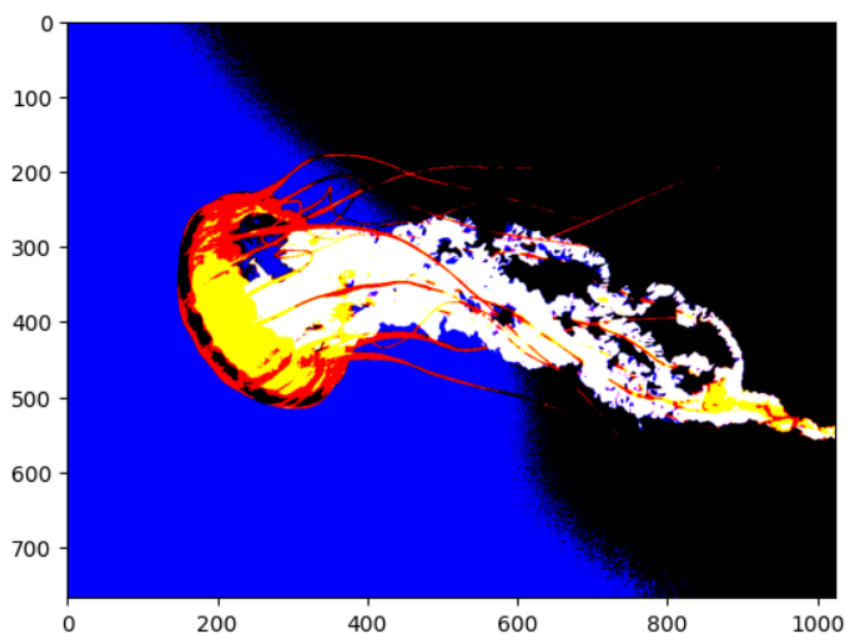


Рисунок 7 – Задание 6, клиент (цветное изображение, порог 100)

```
show_modified_image('../resources/Jellyfish.jpg', server.binarize_by_threshold, 10)
```

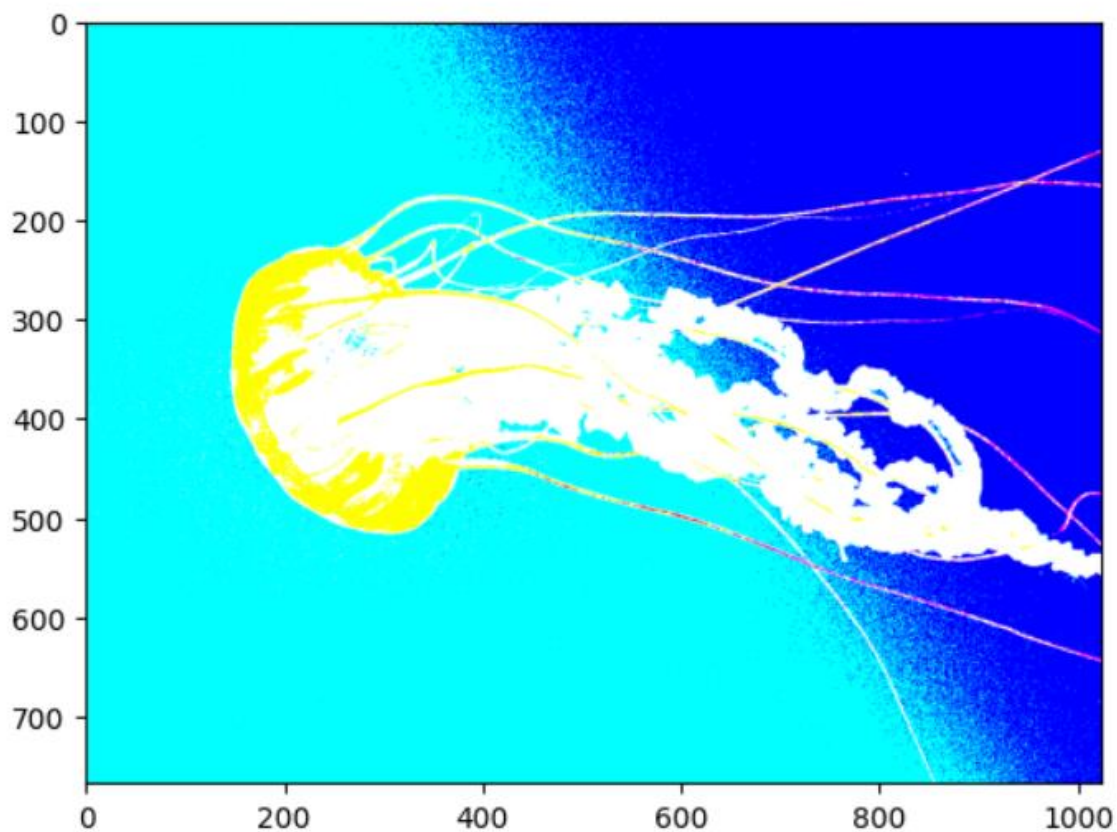


Рисунок 8 – Задание 6, клиент (цветное изображение, порог 10)

```
show_modified_image('../resources/11.bmp', server.binarize_by_threshold, 100)
```

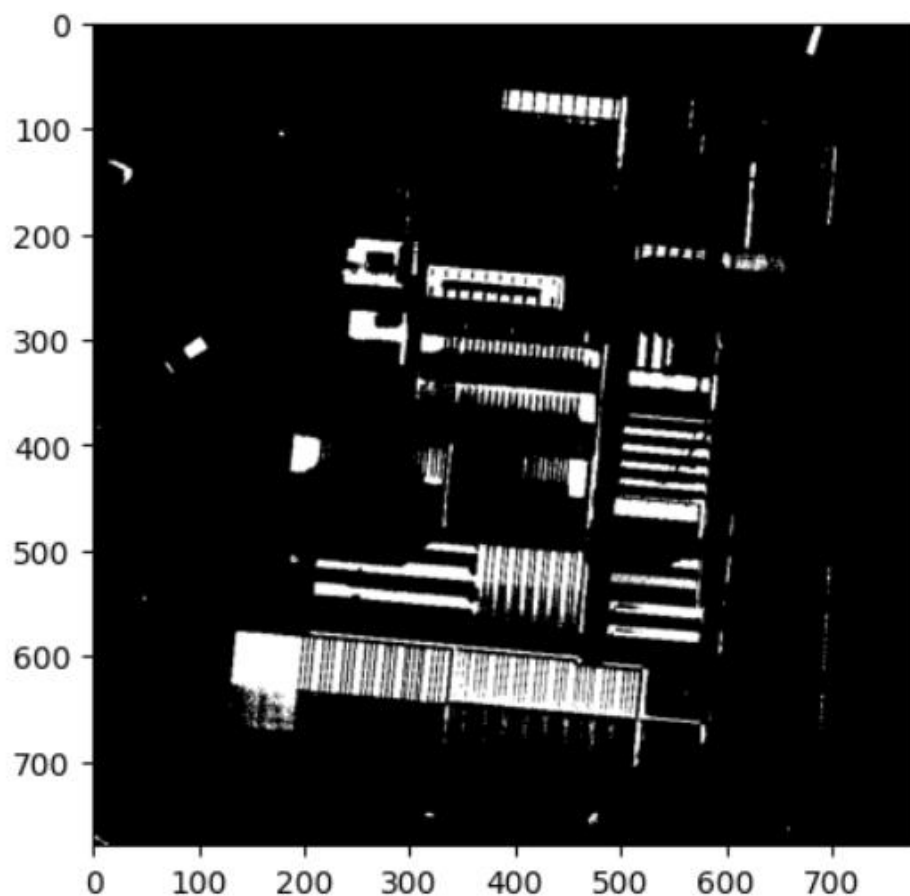


Рисунок 9 – Задание 6, клиент (монохромное изображение, порог 100)

Задание 7. Дополнить функционал сервера функцией разворота изображения относительно вертикали для цветных и монохромных изображений. Для поворота изображения запрещено использовать библиотечные функции.

Код тетради xmlrpc_server.ipynb (рис. 10):

```
# Задание 7. Функция разворота изображения относительно вертикали для цветных и монохромных изображений
def mirror_vertically(bin_data):
    img_arr = pickle.loads(bin_data.data)
    height = img_arr.shape[0]
    width = img_arr.shape[1]
    for i in range(height):
        for j in range(width // 2):
            if type(img_arr[i][j]) == numpy.ndarray:
                t1, t2 = [p for p in img_arr[i][j]], [p for p in img_arr[i][-j - 1]]
                img_arr[i][j], img_arr[i][-j - 1] = t2, t1
            else:
                img_arr[i][j], img_arr[i][-j - 1] = img_arr[i][-j - 1], img_arr[i][j]
    pimg = pickle.dumps(img_arr)
    return client.Binary(pimg)
server.register_function(mirror_vertically, 'mirror_vertically')
```

Рисунок 10 – Задание 7, сервер

Код тетради xmlrpc_client.ipynb и результат выполнения (рис. 11-12):

```
# Задание 7. Функция разворота изображения относительно вертикали для цветных и монохромных изображений  
show_modified_image('../resources/Jellyfish.jpg', server.mirror_vertically)
```

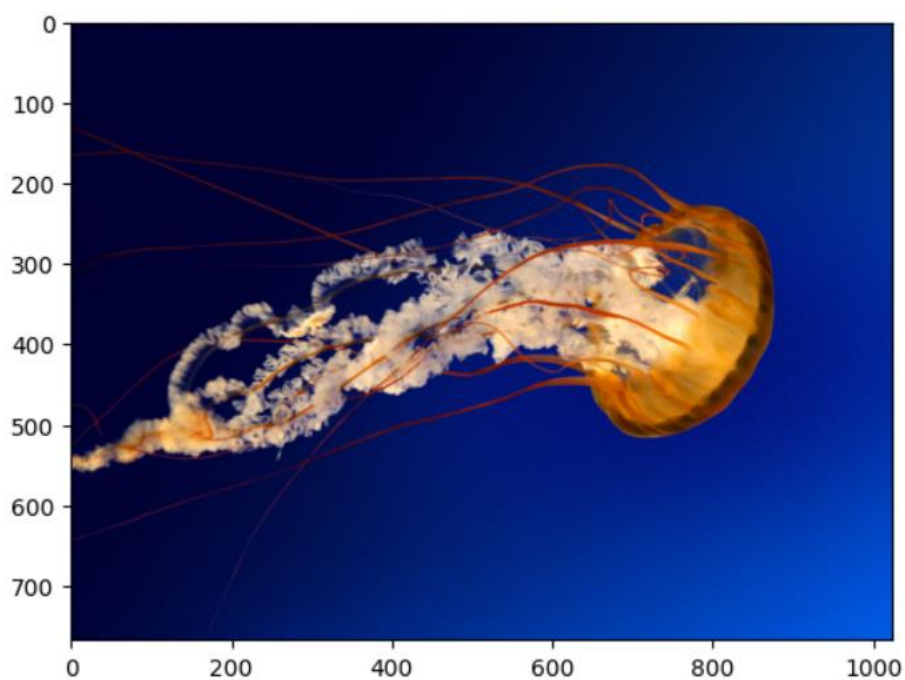


Рисунок 11 – Задание 7, клиент (цветное изображение)

```
show_modified_image('../resources/11.bmp', server.mirror_vertically)
```

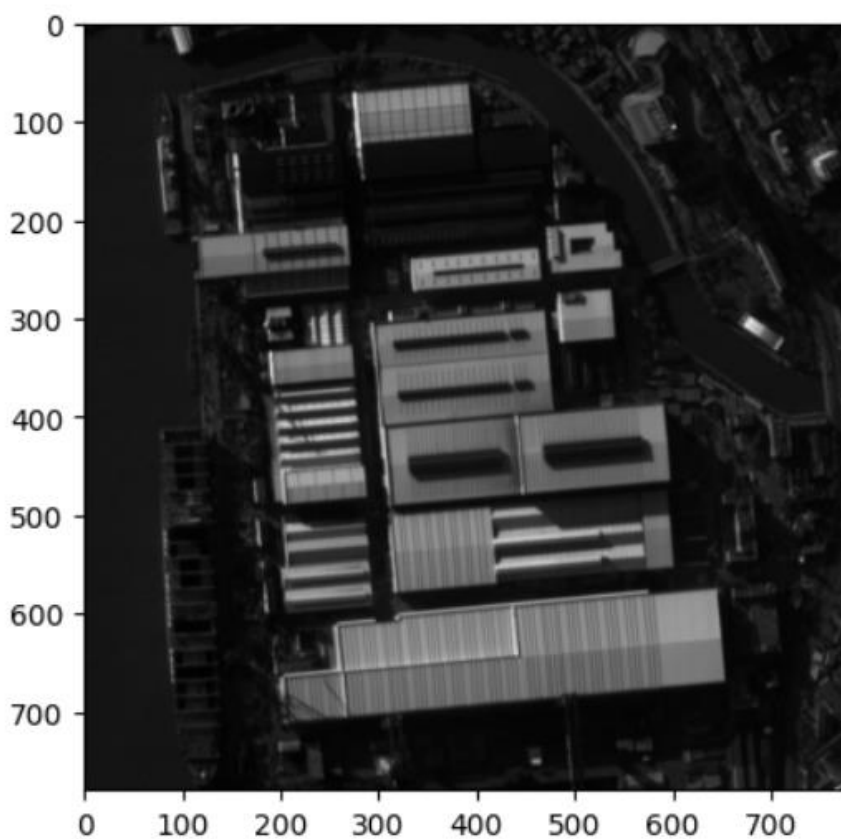


Рисунок 12 – Задание 7, клиент (монохромное изображение)

Задание 8. Дополнить функционал клиента проверкой функций сервера по пунктам 4-7 задания.

Задание 9. Оформить отчет по результатам выполнения ЛР.

Описание полученных навыков

В ходе лабораторной работы были получены практические знания о сервис-ориентированной архитектуре вычислений на основе протокола XML-RPC, навыки работы с некоторыми функциями библиотек `xmlrpc.server`, `xmlrpc.client`, `pickle`, `pandas`, а также чтения файлов `.csv` и преобразования цветных и монохромных изображений.

Вывод

Сервис-ориентированную архитектуру вычислений целесообразно использовать, когда необходимо разделить приложение на несколько независимых сервисов для большей автономности каждого из них. При этом сервисы можно переиспользовать для других приложений.