**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ**

**И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

## Кафедра "Системное программирование"

**Отчет по курсовой работе**

по дисциплине «Операционные системы»

на тему:

«Исследование механизмов межпроцессного взаимодействия»

Выполнила:

студентка группы БВТ2102

Никифорова Олеся Ильинична

Проверила: Королькова Т. В.

Москва 2023

**Содержание**

1. Цель работы
2. Задачи
3. Задание
4. Общие указания к выполнению
5. Дополнительные задания
6. Основная часть
7. Вывод

**Цель работы**

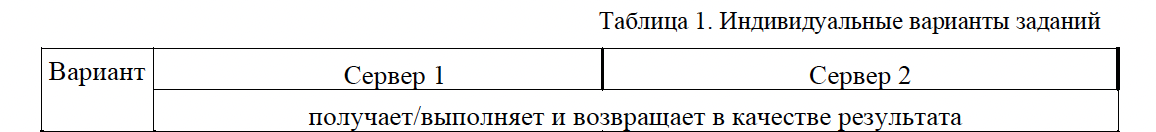
Закрепление и углубление теоретических знаний в области современных операционных систем, приобретение практических навыков разработки клиент-серверных приложений, использующих системные механизмы межпроцессного взаимодействия.

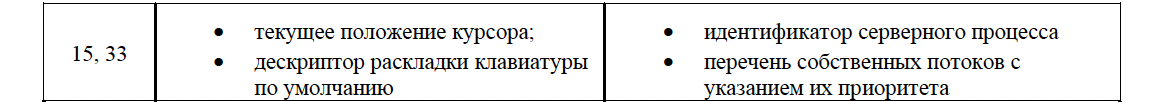
**Задачи**

1. Изучить методы разработки клиент-серверных приложений.
2. Изучить механизмы обмена данными в Windows NT и GNU/Linux и особенности их применения на практике.
3. Изучить средства синхронизации процессов в Windows NT и GNU/Linux и особенности их применения на практике.
4. Изучить функции WinAPI и возможности Linux для получения системной информации, информации о процессах и потоках, состоянии памяти.
5. Разработать серверные приложения и приложение-клиент, обменивающиеся данными через заданный интерфейс.
6. Оформить результаты выполнения курсовой работы в виде пояснительной записки в соответствии с предъявляемыми требованиями.

**Задание**

Разработать многозадачное приложение, выполняющее получение, сбор и вывод системной информации в соответствии с вариантом задания (таблица 1). Номер варианта задания на курсовую работу совпадает с номером, под которым записана фамилия студента в журнале группы.





**Общие указания к выполнению работы**

Многозадачное приложение реализуется на основе технологии «клиент-сервер» средствами интегрированной среды разработки Microsoft Visual Studio/Visual C++ или редактора Visual Studio Code, по согласованию с преподавателем возможно использование других компиляторов и языков программирования. Процессы 1 и 2 выступают как серверы приложений, третий процесс является их клиентом. Обмен данными между клиентами и серверами осуществляется при помощи сокетов.

И клиентские, и серверные процессы оформляются как отдельные приложения, запускаются и завершают работу независимо друг от друга. Серверы должны быть многопоточными, то есть допускать одновременное подключение нескольких клиентов. Серверные процессы по запросу получают системную информацию в соответствии с вариантом задания и передают ее клиенту с указанием текущего времени. На стороне клиента необходимо реализовать интерфейс, позволяющий пользователю подключиться по выбору к первому или второму серверу, а также возможность раздельного получения и отображения данных по каждому из пунктов индивидуального задания.

Необходимо предусмотреть проверку на повторный запуск серверов, т.е. исключить ситуацию, когда запускается более чем один экземпляр каждого сервера. В течение всего времени подключения клиента к серверу информация в клиентской части должна обновляться либо автоматически, либо по команде от пользователя.

**Дополнительные задания**

1. При обновлении информации на клиенте по таймеру данные передаются только в том случае, если что-либо изменилось с момента предыдущего обновления.
2. Обновление информации на клиенте происходит по инициативе сервера как реакция на изменения на сервере.
3. Существует отдельный сервер, который ведет лог операций (отдельныйдля каждого сервера, с сохранением в файл), в котором фиксируются подключения клиентов, переданные запросы, ошибки и т.д. Взаимодействие между серверами осуществляется с помощью каналов.
4. Реализация графического интерфейса на стороне клиента.

**Выполнение**

# Вывод

В результате выполненной работы были изучены организация виртуальной памяти в Linux, основные системные вызовы и функции в ОС GNU/LINUX для работы с файлами и каталогами, построена карта виртуальной памяти процесса.