



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»  
(МТУСИ)**

ОТЧЕТ  
по курсовой работе  
дисциплина «Операционные системы»  
на тему «Исследование механизмов межпроцессного  
взаимодействия»

Выполнил:  
студент гр. БВТ2102  
Маширин Ф. С.

Москва, 2023 г.

Целью выполнения данной курсовой работы является закрепление и углубление теоретических знаний в области современных операционных систем, приобретение практических навыков разработки клиент-серверных приложений, использующих системные механизмы межпроцессного взаимодействия

### **Задачи курсовой работы:**

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.

Вариант	Сервер 1	Сервер 2
12	<ul style="list-style-type: none"><li>• количество жестких дисков и тип файловой системы каждого</li><li>• количество логических процессоров</li></ul>	<ul style="list-style-type: none"><li>• процент используемой физической памяти;</li><li>• процент используемой виртуальной памяти</li></ul>

### **Дополнительные задания**

1. Запуск клиентов в контейнерах.

## Введение

Межпроцессное взаимодействие — обмен данными между потоками одного или разных процессов. Реализуется посредством механизмов, предоставляемых ядром ОС или процессом, использующим механизмы ОС и реализующим новые возможности ИРС. Может осуществляться как на одном компьютере, так и между несколькими компьютерами сети.

Из механизмов, предоставляемых ОС и используемых для ИРС, можно выделить:

- механизмы обмена сообщениями;
- механизмы синхронизации;
- механизмы разделения памяти;
- механизмы удалённых вызовов (RPC).

Для оценки производительности различных механизмов ИРС используют следующие параметры:

- пропускная способность (количество сообщений в единицу времени, которое ядро ОС или процесс способно обработать);
- задержки (время между отправкой сообщения одним потоком и его получением другим потоком).

ИРС может называться терминами межпотокное взаимодействие (англ. inter-thread communication) и межпрограммное взаимодействие (англ. inter-application communication).

Межпроцессное взаимодействие, наряду с механизмами адресации памяти, является основой для разграничения адресного пространства между процессами.

Понимание принципов межпроцессного взаимодействия является очень важным, поэтому в ходе данной курсовой работы нами будет рассмотрен один из способов межпроцессного взаимодействия – сокеты.

# Основная часть

## Постановка задачи

Целью данной курсовой работы является закрепление и углубление теоретических знаний в области современных операционных систем, приобретение практических навыков разработки клиент-серверных приложений, использующих стандартные механизмы межпроцессного взаимодействия.

## Краткие теоретические основы разрабатываемой темы

«Клиент — сервер» (англ. client-server) — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.

Фактически клиент и сервер — это программное обеспечение. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов, но они могут быть расположены также и на одной машине. Программы-серверы ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных (например, загрузка файлов посредством HTTP, FTP, BitTorrent, потоковое мультимедиа или работа с базами данных) или в виде сервисных функций (например, работа с электронной почтой, общение посредством систем мгновенного обмена сообщениями или просмотр web-страниц во всемирной паутине).

Поскольку одна программа-сервер может выполнять запросы от множества программ-клиентов, её размещают на специально выделенной вычислительной машине, настроенной особым образом, как правило, совместно с другими программами-серверами, поэтому производительность этой машины должна быть высокой. Из-за особой роли такой машины в сети, специфики её оборудования и программного обеспечения, её также называют сервером, а машины, выполняющие клиентские программы, соответственно, клиентами.

## **Разработка и описание алгоритма работы приложения**

Сервера 1 и 2 запускаются на любых портах ожидают входящее соединение. После установления соединения передают данные клиенту, такие как “Ширину и высоту основного монитора”, “Количество мониторов” – с первого сервера и “размер файла подкачки в байтах”, “количество свободных байтов файла подкачки” – со второго. Клиент взаимодействует с сервером путем ввода в консоль номера требуемого сервера. Потом клиент вводит тип сообщения. После этого у клиента выбор: выводить сообщения каждую секунду, либо по вводу сообщения. При выборе определенного сервера, на самом сервере отображается информация о том, какой клиент подключился и когда получил данные.

## **Описание структуры программы**

В серверной части приложения определена функция, отвечающая за передачу данных клиенту, расчет информации. Используются функции `getDiskInfo`, `getCPUInfo`, `getMemoryValue` выполняющие задания.

В клиентской части определена функция, отвечающая за выбор сервера, за отправку запроса на сервер и вывод полученного ответа.

## **Руководство пользователя**

Прежде всего необходимо запустить оба сервера, иначе данные не будут получены клиентом. В открывшейся консоли клиента для соединения с первым сервером надо ввести цифру и нажать «Enter», на экран будет выведена информация от сервера.

```
PROBLEMS OUTPUT TERMINAL PORTS 4 DEBUG CONSOLE
```

```
gostik@DESKTOP-2THHGK:/home/mashirin_fedor/os_kurs$ ./server 8001
[Server] All setting are done
[Server] Server enabled
Сервер 1: Задание 1:
Количество жестких дисков и тип файловой системы каждого
Сервер 1: Задание 2:
количество логических процессоров
[Server] Waiting for connection...
[Server] Connection accepted

[Server] Client message accepted
[Server] Client message accepted
[Server] Client message: q
[Server] 2023-12-22 01:22:46 :: Количество логических процессоров = 1
2
NAME FSTYPE SIZE TYPE MOUNTPOINTS
loop0 488.6M loop /mnt/ws1/docker-desktop/cli-tools
loop1 180.2M loop
loop2 779.7M loop
sda ext4 389.8M disk
sdb swap 2G disk [SWAP]
sdc ext4 1T disk /mnt/ws1/docker-desktop-bind-mounts/Ubuntu/4bd3bc117e3f284b1976d3d93a109ac88848b8405adc3d64ab992a59ed1d9915
                        /fnap
                        /mnt/ws1/gdistro

sdd ext4 1T disk /mnt/ws1/docker-desktop/docker-desktop-user-
distro
sde ext4 1T disk /mnt/ws1/docker-desktop-data/isocache

Message sent to client

*****
[]
```

```
gostik@DESKTOP-2THHGK:/home/mashirin_fedor/os_kurs$ ./server2 8002
[Server] All setting are done
[Server] Server enabled
Сервер 2: Задание 1:
процент используемой физической памяти
Сервер 2: Задание 2:
процент используемой виртуальной памяти
[Server] Waiting for connection...
[]
```

```
gostik@DESKTOP-2THHGK:/home/mashirin_fedor/os_kurs$ ./client 8001
You connect to port: 8001
[Client] All setting are done
[Client] Successfully connected to server

-----
Сервер 1: Задание 1:
количество жестких дисков и тип файловой системы каждого
Сервер 1: Задание 2:
количество логических процессоров
Сервер 2: Задание 1:
процент используемой физической памяти
Сервер 2: Задание 2:
процент используемой виртуальной памяти
[Client] Enter message(0 Полный ответ, 1 Задание, 2 Задание, 3 Откликание): 0
[Client] Enter message(1 Каждую секунду, 2 Каждый ввод): 2
[Client] Message sent to server
[Client] Enter any message:
0
[Client] Message sent to server
[Client] Server message: 2023-12-22 01:22:46 :: Количество логических процессоров = 12
NAME FSTYPE SIZE TYPE MOUNTPOINTS
loop0 488.6M loop /mnt/ws1/docker-desktop/cli-tools
loop1 180.2M loop
loop2 779.7M loop
sda ext4 389.8M disk
sdb swap 2G disk [SWAP]
sdc ext4 1T disk /mnt/ws1/docker-desktop-bind-mounts/Ubuntu/4bd3bc117e3f284b1976d3d93a109ac88848b8405adc3d64ab992a59ed1d9915
                        /fnap
                        /mnt/ws1/gdistro

sdd ext4 1T disk /mnt/ws1/docker-desktop/docker-desktop-user-
-distro
sde ext4 1T disk /mnt/ws1/docker-desktop-data/isocache

[Client] Enter any message: []
```

Рис. 1: Подключение к первому серверу с запросом каждый ввод

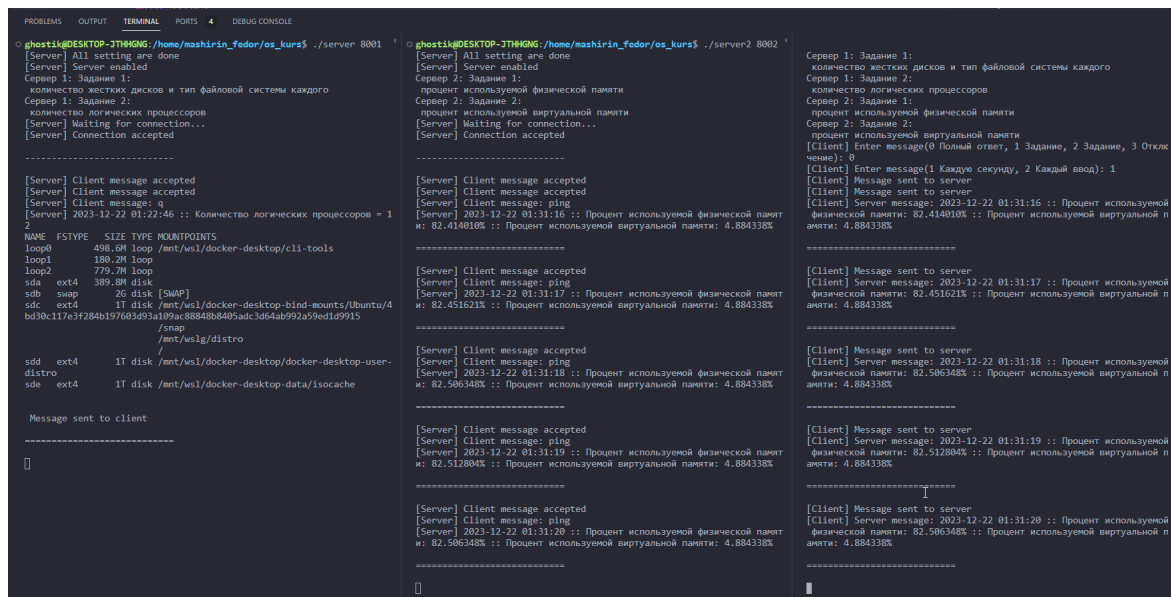


Рис. 2: Подключение к второму серверу с запросом каждую секунду

## **Заключение**

Код программы расположен на GitHub сервисе по адресу [https://github.com/Ghostik-gh/Kurovaya\\_po\\_OS](https://github.com/Ghostik-gh/Kurovaya_po_OS)

В ходе выполнения курсовой работы мы изучили методы разработки клиент-серверных приложений, а также механизмы обмена данными и средства синхронизации процессов в GNU/Linux с их особенностями применения на практике. Разобрали функции Linux для получения системной информации, информации о процессах и потоках, состоянии памяти, серверные приложения и приложение-клиент, обменивающиеся данными через заданный интерфейс. И оформили результаты выполнения курсовой работы в виде пояснительной записки в соответствии с предъявляемыми требованиями.

## Список литературы

1. Э.Таненбаум. Современные операционные системы. 3-изд. – СПб.: Питер, 2011
2. Э.Таненбаум, Х.Бос. Современные операционные системы. 4-изд. – СПб.: Питер, 2015
3. В.Г.Олифер, Н.А.Олифер. Сетевые операционные системы. – СПб.: Питер, 2009
4. С.В. Назаров, А.И. Широков. «Современные операционные системы». – М., Бином, 2013
5. Марк Руссинович, Дэвид Соломон, Алекс Ионеску, Павел Йосифович. Внутреннее устройство Windows. 7-е издание . - СПб.: Питер, 2018
6. Вдовикина Н.В., Машечкин И.В., Терехин А.Н., Томилин А.Н. «Операционные системы: взаимодействие процессов: учебно-методическое пособие». – Издательский отдел факультета ВМиК МГУ, 2008