

ЦЕЛЬ РАБОТЫ

Следует реализовать приложение на C99/C++11, которое должно уметь:

1. Получать данные из socket.
2. Передавать данные в socket.
3. Идентифицировать отправителя/получателя.
4. Вести историю передаваемых данных.

ОСНОВНАЯ ЧАСТЬ

1.1 Теоретическая часть

Сокеты — название программного интерфейса для обеспечения обмена данными между процессами. Процессы при таком обмене могут исполняться как на одной ЭВМ, так и на различных ЭВМ, связанных между собой сетью. Сокет — абстрактный объект, представляющий конечную точку соединения.

Существует два типа сокетов — UNIX-сокеты (или сокеты домена UNIX — Unix domain sockets) и INET-сокеты (IP-сокеты, network sockets).

UNIX-сокеты являются частью механизма IPC и позволяют обмен данными в обоих направлениях между процессами, работающими на одной машине.

INET-сокеты в свою очередь представляют собой механизм, позволяющий выполнять коммуникацию между процессами по сети.

Inet - сокеты именуются с помощью IP адресов и номеров портов Unix - сокетам даются имена объектов типа socket в файловой системе.

1.2 Практическая часть

Было создано приложение на C99/C++11, которое действует по сценарию:

1. Сценарий сервиса по подсчету символов в слове, представленный в приложении А

1. Создается ip-сокеты, связывается с IP-адресом и портом и начинает ждать подключений.
2. При подключении пользователя сервис просит его авторизоваться.
3. После авторизации пользователя сервис выдает правила своей работы и ждет ввода.

2. Сценарий пользователя, представленный в приложении Б

1. Обращается к созданному сервисом IP-сокеты.
2. Вводит свой login по просьбе сервиса.
3. Производит ввод слов.

Есть три вида ввода пользователя:

- Слово - подсчет символов в данном слове. При вводе несколько слов (последовательность символов через пробел) сервис подсчитает символы в каждом слове отдельно.
- `logout` - отключение текущего пользователя и ожидание нового подключения.
- `end` - завершение работы сервиса.

Ввод пользователя и ответ сервиса выводятся в консоль в виде:

Service -> user : сообщение

user -> Service : сообщение

```
hruleva@DebianHruleva:~/lab6$ ./client
Service -> unknown: Здравствуйте, как можно к вам обращаться?
sally
sally -> Service : sally
Service -> sally :
1. Сервис выдает количество букв в слове. При вводе нескольких слов, подсчитает для каждого.
2. Закончить работу пользователю "logout"
3. Закончить работу с сервисом - "end"
Вы были авторизованы как sally.
Введите слово
banana
sally -> Service : banana
Service -> sally :
В слове banana 6 символов.
Введите новое слово
(^...^)
sally -> Service : (^...^)
Service -> sally :
В слове (^...^) 7 символов.
Введите новое слово
logout
sally -> Service : logout
hruleva@DebianHruleva:~/lab6$ ./client
Service -> unknown: Здравствуйте, как можно к вам обращаться?
mika
mika -> Service : mika
```

Рисунок 1 – Пример работы сервиса с пользователем

```
hruleva@DebianHruleva:~/lab6$ ./client
Service -> unknown: Здравствуйте, как можно к вам обращаться?
mika
mika -> Service : mika
Service -> mika :
1. Сервис выдает количество букв в слове. При вводе нескольких слов, подсчитает для каждого.
2. Закончить работу пользователю "logout"
3. Закончить работу с сервисом - "end"
Вы были авторизованы как mika.
Введите слово
#_#
mika -> Service : #_#
Service -> mika :
В слове #_# 3 символов.
Введите новое слово
!!!game!!!
mika -> Service : !!!game!!!
Service -> mika :
В слове !!!game!!! 10 символов.
Введите новое слово
end
mika -> Service : end
hruleva@DebianHruleva:~/lab6$ █
```

Рисунок 2 – Пример работы сервиса с пользователем

```

hruleva@DebianHruleva:~/lab6$ ./server
Подключен к 2004
Есть новое подключение
Service -> unknown: Здравствуйте, как можно к вам обратиться?
sally -> Service : sally
Service -> sally :
1. Сервис выдает количество букв в слове. При вводе нескольких слов, подсчитает для каждого.
2. Закончить работу пользователю "logout"
3. Закончить работу с сервисом - "end"
Вы были авторизованы как sally.
Введите слово
sally - > Service : banana
Service -> sally :
В слове banana 6 символов.
Введите новое слово
sally - > Service : (^...^)
Service -> sally :
В слове (^...^) 7 символов.
Введите новое слово
sally - > Service : logout
Service - > sally : Подключение пользователя sally остановлено
Есть новое подключение
Service -> unknown: Здравствуйте, как можно к вам обратиться?
mika -> Service : mika

```

Рисунок 3 – Пример работы сервиса с пользователем

```

Есть новое подключение
Service -> unknown: Здравствуйте, как можно к вам обратиться?
mika -> Service : mika
Service -> mika :
1. Сервис выдает количество букв в слове. При вводе нескольких слов, подсчитает для каждого.
2. Закончить работу пользователю "logout"
3. Закончить работу с сервисом - "end"
Вы были авторизованы как mika.
Введите слово
mika - > Service : #_#
Service -> mika :
В слове #_# 3 символов.
Введите новое слово
mika - > Service : !!!game!!!
Service -> mika :
В слове !!!game!!! 10 символов.
Введите новое слово
mika - > Service : end
Service - > mika : Закрытие сервиса
hruleva@DebianHruleva:~/lab6$ █

```

Рисунок 4 – Пример работы сервиса с пользователем

ВЫВОДЫ

Таким образом, в ходе лабораторной работы были изучены ip-sockets и реализовано приложение на C99/C++11 , осуществляющее работу с ip-socket.