Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №8
«Сокеты. Взаимодействие процессов»

Выполнил:
студент группы 150501
Ковальчук Д.И.

Проверил:
старший преподаватель
_____ Поденок Л.П.

Минск
2023

1 ПОСТАНОВКА ЗАДАЧИ

Задача разработка многопоточного сервера и клиента, работающих по простому протоколу.

Изучаемые системные вызовы: `socket()`, `bind()`, `listen()`, `connect()`, `accept()` и прочих, связанных с адресацией в домене `AF_INET`.

Протокол должен содержать следующие запросы:

`ECHO` - эхо-запрос, возвращающий без изменений полученное от клиента;

`QUIT` - запрос на завершение сеанса;

`INFO` - запрос на получения общей информации о сервере;

`CD` - изменить текущий каталог на сервере;

`LIST` - вернуть список файловых объектов из текущего каталога.

Протокол может содержать дополнительные запросы по выбору студента, не выходящие за пределы корневого каталога сервера и не изменяющих файловую систему в его дереве.

Запросы клиенту отправляются на `stdin`.

Ответы сервера и ошибки протокола выводятся на `stdout`.

Ошибки системы выводятся на `stderr`.

Подсказка клиента для ввода запросов символ '>'.

Клиент помимо интерактивных запросов принимает запросы из файла. Файл с запросами указывается с использованием префикса '@':

```
$ myclient server.domen
Вас приветствует учебный сервер ‘myserver’
> @file
> ECHO какой-то текст
Какой-то текст
> LIST
dir1
dir2
file
> CD dir1
dir1> QUIT
BYE
$
```

`ECHO` - эхо-запрос, возвращающий без изменений полученное от клиента.

```
> ECHO “произвольный текст”
произвольный текст
>
```

```
QUIT - запрос на завершение сеанса

> QUIT
```

```
BYE
$
```

INFO - запрос на получения общей информации о сервере.
Сервер отправляет текстовый файл с соответсвующей информацией.

```
> INFO
Вас приветствует учебный сервер ‘myserver’
```

Этот же файл сервер отправляет клиенту при установлении сеанса.
LIST - вернуть список файловых объектов из текущего каталога.
Текущий каталог - каталог в дереве каталогов сервера. Корневой каталог сервера устанавливается из командной строки при старте сервера.

```
> LIST
dir1/
dir2/
file1
file2 --> dir2/file2
file3 -->> dir1/file
>
```

Каталоги выводятся с суффиксом '/' после имени, файлы как есть, симлинки на регулярные файлы разрешаются через '-->', симлинки на симлинки разрешаются через '-->>'. Корневой каталог сервера пр выводе указывается префиксом '/' перед именем.

CD - изменить текущий каталог на сервере.
Выход за пределы дерева корневого каталога сервера запрещается, команда безмолвно игнорируется.

```
dir2> LIST
file2
dir2> CD ../dir1
dir1> LIST
file --> /file1
dir1> CD ..
> CD ..
>
```

Соединения функционируют независимо, т. е. текущий каталог у каждого соединения свой.

2 АЛГОРИТМ

Алгоритм программы `server`:

1) Начало.
2) Установка обработчика прерываний.
3) Инициализация сокета, ожидание подключения от очередного клиента.
4) Создание нового потока и запуск в нём обработчика команд клиента.
4) Посылка информации о сервере клиенту.
5) Получение команды от клиента.
6) Анализ команды.
7) Команда “LIST” – список файловых объектов текущего каталога.
8) Команда “CD” – смена текущего каталога.
9) Команда “ECHO” – эхо-запрос.
10) Опция “INFO” – запрос на получение общей информации на сервере.
11) Опция “HELP” – вывод списка всех команд.
12) Опция “QUIT” – запрос на завершение сеанса.
13) Закрытие очередного потока и используемого в нём сокета, переход к шагу 3.
14) Конец.

Алгоритм программы `client`:

1) Начало.
2) Установка обработчика прерываний.
3) Ввод имени (IP-адреса) сервера.
4) Инициализация сокета, установление подключения с сервером.
5) Ввод команды
6) Посылка команды серверу.
7) Приём ответа сервера.
8) Если был введён запрос на выход, переход к шагу 9, иначе к шагу 5.
9) Закрытие сокета.
10) Конец.

3 ТЕСТ

Вывод программы server:

```
darya@:~/osisp/lab08$ ./server  ~/osisp/lab08/
SERVER INIT
FTP-SERVER WAITS FOR CLIENT CONNECTION
GOT ACCESS TO NEW CLIENT
cmd 250 ok
WELCOME TO MY_SERVER! SERVER IS READY WOR WORK!
ENTER HELP TO GET AVAILABLE COMMANDS.

help

CLIENT COMMAND IS help
ARGUMENT IS
AVAILABLE COMMANDS:
HELP – ВЫВОД ПОМОЩИ
ECHO – ЭХО-ЗАПРОС, КОТОРЫЙ ВОЗВРАЩАЕТ ПОЛУЧЕННОЕ БЕЗ ИЗМЕНЕНИЯ
INFO – ПОЛУЧЕНИЕ ИНФОРМАЦИИ О СЕРВЕРЕ
CD   – СМЕНА ТЕКУЩЕГО КАТАЛОГА
LIST – СПИСОК ОБЪЕКТОВ ИЗ ТЕКУЩЕГО КАТАЛОГА
QUIT – ЗАВРЕШЕНИЕ СЕАНСА
ВВОД МОЖЕТ ПРОИЗВОДИТЬСЯ КАК С ПОМОЩЬЮ CAPS-LOCK, ТАК И БЕЗ

info

CLIENT COMMAND IS info
ARGUMENT IS
cmd 250 ok
WELCOME TO MY_SERVER! SERVER IS READY WOR WORK!
ENTER HELP TO GET AVAILABLE COMMANDS.

list

CLIENT COMMAND IS list
ARGUMENT IS
header.h
client
script.txt
server
server.c
client.c
readme
makefile

echo hello

CLIENT COMMAND IS echo
ARGUMENT IS  hello

hello

quit
```

```
CLIENT COMMAND IS quit
ARGUMENT IS
CMD 231 OKAY, USER LOGGER OUT

CLOSING CLIENT SOCKET
```

Вывод программы `client`:

```
darya@:~/osisp/lab08$ ./client
CLIENT STARTED WORK

ENTER NAME OR IP-ADRESS OF SERVER (quit TO EXIT): localhost
CONNECTING TO SERVER
LOG: cmd 250 ok
WELCOME TO MY_SERVER! SERVER IS READY WOR WORK!
ENTER HELP TO GET AVAILABLE COMMANDS.

> help
AVAILABLE COMMANDS:
HELP — ВЫВОД ПОМОЩИ
ECHO — ЭХО-ЗАПРОС, КОТОРЫЙ ВОЗВРАЩАЕТ ПОЛУЧЕННОЕ БЕЗ ИЗМЕНЕНИЯ
INFO — ПОЛУЧЕНИЕ ИНФОРМАЦИИ О СЕРВЕРЕ
CD   — СМЕНА ТЕКУЩЕГО КАТАЛОГА
LIST — СПИСОК ОБЪЕКТОВ ИЗ ТЕКУЩЕГО КАТАЛОГА
QUIT — ЗАВРЕШЕНИЕ СЕАНСА
ВВОД МОЖЕТ ПРОИЗВОДИТЬСЯ КАК С ПОМОЩЬЮ CAPS-LOCK, ТАК И БЕЗ

> info
LOG: cmd 250 ok
WELCOME TO MY_SERVER! SERVER IS READY WOR WORK!
ENTER HELP TO GET AVAILABLE COMMANDS.

> list
header.h
client
script.txt
server
server.c
client.c
readme
makefile

> echo hello

hello

> quit
CMD 231 OKAY, USER LOGGER OUT

CLIENT FINISHED WORK.
```

4 ВЫВОД

В ходе выполнения лабораторной работы была разработана программа многопоточного сервера и клиента, работающих по простому протоколу, были предусмотрены тупиковые ситуации, если сервер или клиент разрывают подключение. Также были использованы такие функции как `socket()`, `bind()`, `listen()`, `conect()`, `accept()`.