

## Лабораторная работа № 5 для группы К8-123

Создать клиент-серверное приложение, обеспечивающее хранение и выдачу файлов.

### Сервер

Обеспечивает прием, хранение и выдачу файлов по запросам клиентов. Каждому файлу присваивается числовой идентификатор (порядковый номер) типа `int`. Нумерация сквозная возрастающая. Исходное имя файла не сохраняется и не передается.

По запросам клиентов сервер должен:

- принять новый файл и сообщить клиенту присвоенный ему идентификатор;
- выдать файл по идентификатору по запросу клиента;
- выдать список имеющихся на сервере файлов с указанием их идентификаторов, размера и времени создания;
- удалить файл по запросу клиента.

Протокол взаимодействия сервера и клиента разрабатывается самостоятельно.

Сервер должен работать в виде процесса-демона с использованием системы `syslog` и записывать в журнал информацию обо всех транзакциях. Сервер при запуске создает временный каталог для хранения файлов. Этот каталог должен быть удален вместе с содержимым при завершении сервера после получения сигнала `SIGTERM`. Сервер должен отслеживать повторный запуск и сообщать об ошибке, если уже одна копия сервера запущена. Каталог хранения файлов задается в качестве параметра. При запуске без параметров используется каталог `/tmp/server-$pid`, где `$pid` – PID процесса-сервера.

### Клиент

Выполняется в виде утилиты командной строки. Должен проанализировать опции, выполнить одно из действий, поддерживаемых сервером и завершиться. Если сервер не запущен, то вывести сообщение об ошибке. Опции разрабатываются самостоятельно и должны включать: команду, идентификатор файла, имя файла для передачи или сохранения.

Взаимодействие клиента и сервера осуществляется при помощи:

#### Вариант 1

Очереди сообщений

В этом случае сервер отвечает за создание и удаление очереди сообщений. По наличию очереди сообщений сервер может отслеживать повторный запуск, а клиент наличие запущенного сервера. Необходимо предусмотреть передачу файлов произвольного размера, несмотря на то, что размер сообщений в очереди ограничен.

#### Вариант 2

Семафоров и разделяемой памяти

В этом случае сервер отвечает за создание и удаление семафоров и разделяемой памяти. По наличию этих объектов сервер может отслеживать повторный запуск, а клиент наличие запущенного сервера. Необходимо предусмотреть одновременную работу с несколькими клиентами (несколько областей памяти или последовательно при помощи семафоров).

#### Вариант 3

Дейтаграммных сокетов в домене UNIX

Наличие запущенного сервера можно проверять специальной `echo`-дейтаграммой, на которую сервер обязан ответить. Необходимо предусмотреть передачу файлов произвольного размера, несмотря на то, что размер дейтаграмм может быть ограничен.

#### **Вариант 4**

Дейтаграммных сокетов в домене Интернет

Наличие запущенного сервера можно проверять специальной echo-дейтаграммой, на которую сервер обязан ответить. Необходимо предусмотреть передачу файлов произвольного размера, несмотря на то, что размер дейтаграмм может быть ограничен. необходимо помнить, что протокол ненадежный и возможны дублирование и потери дейтаграмм.

#### **Вариант 5**

Потоковых сокетов в домене UNIX

Наличие запущенного сервера можно проверять по наличию файла типа сокет с заданным именем в каталоге

#### **Вариант 6**

Потоковых сокетов в домене Интернет

Разрабатывается классическое клиент-серверное Internet-приложение.