Цель работы. Создание клиент-серверных приложений, взаимодействующих друг с другом по сети на основе технологии соединения на сокетах L4.

Последовательность выполнения работы:

1. Проанализируйте код программы server_game.cpp , иллюстрирующей обмен данными с клиентскими приложениями по итеративной схеме.
2. Скомпилируйте и запустите server_game.
3. Запустите другой терминал и проверьте с него наличие в системе созданного сервером сокета и то, что он находится в состоянии LISTEN. Для этого выполните команду netstat -a | grep 1066 . Проанализируйте вывод данной команды и объясните ее смысл.
4. Запустите в качестве клиентского процесса утилиту telnet с параметрами: telnet localhost 1066 . При организации коммуникации по сети на разных компьютерах вместо localhost при запуске клиента указывается IP-адрес компьютера, на котором был запущен сервер.
5. Диалог с сервером заключается в угадывании слова. Оно вводится по буквам с клиентского терминала. При этом сервер вместо неугаданных букв выдает символы ”-” , а также считает число оставшихся неудачных попыток (всего их предусмотрено 12).
6. Завершите серверное приложение с помощью сигнала kill , и затем определите командой netstat -a | grep 1066 , когда исчезает из системы соединение на сокетах. Во время сеанса обмена также примените команду netstat -a | grep 1066 , чтобы исследовать состояние соединения.
7. Проделайте все заново, но запускайте не одно клиентское приложение (в виде telnet), а несколько экземпляров с разных терминалов, и попытайтесь работать с них одновременно. Проанализируйте, как сервер будет обслуживать запросы в этом случае.
8. Модифицируйте программу server_game.cpp так, чтобы запросы от каждого из клиентов могли обслуживаться конкурентно, путем запуска для каждого нового соединения собственного нового процесса на сервере или потока. Проанализируйте, как обслуживаются запросы в случае конкурентной схемы работы сервера. Возможно также улучшить качество самой игровой функции guess_word() сервера.