ОС лаба 1

Задание 1

Парные сокеты, sockpair
Создаем child и он с parent обменивается сообщениями – дуплексная связь
Child посылает “Hello parent”, и parent посылает …

Задание 2

Взаимодействие через сокеты на отдельно стоящей машине AF_UNIX, Type – DGRAM
2 варианта

a) Клиент только посылает сообщение серверу, ничего не получает от сервера
b) Взаимодействие клиента и сервера. Клиент посылает сообщение-запрос, сервер посылает сообщение-ответ

Реализуем КАЛЬКУЛЯТОР
Клиент посылает 2 числа и операцию. Сервер должен посчитать

2 программа:
Клиент отправляет серверу 2 числа и операцию
Сервер выполняет работу калькулятора
На стороне клиент, т к клиент получает от сервера ответ, надо заполнить поля struct sockaddr. Как правило в качестве файла клиента указывается PID.cli
Надо вызывать bind, чтобы связать клиента с его адресом
Посылка – sendto с указанием адреса клиента. Но для получения ответа надо вызвать receivefrom. Никаких connect
На сервере: инициализируются поля адреса сервера, устанавливается связь, receivefrom(адрес клиента), sendto
Сист вызов bind связывает файл дескриптор сокета сервера с адресом сервера. Имя сервера задается #define
Вывод как в RPC, на стороне клиента: a + b = c
Когда демонстрируем запускаем много клиентов. Можно создать клиента с циклом повторения обращения к серверу. Можно fork
Нужно ли здесь взаимоисключение
SIGTERM

Задание 3

Листинг 5.9, стр 171
Один и тот же клиент работает как читатель и как писатель.
Как читатель – получает информацию от сервера (массив)
(элементы массива могут быть недоступны, они помечаются как пробел или -1)
Клиент на основе данных массива запрашивает свободный элемент и посылает индекс.

Клиент запросил соединение, получил и отправил признак что он читатель, сервер отправляет ему массив.
После этого клиент переходит в режим записи и указывает индекс массива который он хочет получить.
Тот же самый процесс, который обслуживал его как читателя будет обслуживать его как писателя
Запись в массив только в режиме монопольного доступа

Потом на потоках, у потоков нет собственного АП
Мультиплексирование

Сетевой сервер в 3х вариантах, кот работает по задаче читатель-писатель:
На сервере есть массив (или вектор)
Клиент может обращаться к серверу как читатель или как писатель
Если клиент обращается как читатель, т е он посылает буковку ‘r’, то сервер предоставляет ему возможность прочитать этот массив, то сервер должен его послать
Сервер посылает массив, в этом массиве какие-то элементы уже могут быть заняты, т е он ему посылает текущее состояние массива
Если ячейка пустая (элемент массив – ‘ ’ или -1), то это – элемент недоступен.
Клиент может выбрать элемент (его индекс), указать что он писатель и это уже должно выполняться в режиме монопольного доступа. Если окажется что др клиент оказался шустрее, то возвращается – ‘элемент занят’. Если никто не претендует, то в режиме монопольного доступа сервер определяет это элемент как занятый и заменяет значение элемента на пробел или -1 и отсылает что ОК.
3 варианта:

a) 1 accept и распараллелить fork() (в книге есть пример)
b) 1 accept и распараллелить pthread() (надо уточнить про id и про группу)
c) Мультиплексирование (на лек 2 будет 2 модели ввода-вывода – разговор про мультиплексирование)

Читатель читает индекс, выбирает тот который свободен, переходит в режим писателя
Один и тот же клиент