report.md 6/12/2021

**Цельработы**

**Приобретение праĸтичсеĸихнавыĸов работы с именованными ĸаналами.**

**Ход работы**

**Изучили приведенные в теĸсте программы** server.c**,**client.c**.Взяв данные примеры за образец, написали аналогичные программы,внесяслдующ ие изменения**



**Работаетдва ĸлиента,а не один.Клиенты передаюттеĸущ ее времяс неĸоторой периодичностью .Использовали фунĸцию sleep() для приостановĸи работы ĸлиента.Сервер работает не бесĸонечно,а преĸращ ает работу через 30 сеĸунд.Использовали фунĸцию clock() дляопределениявремени работы сервера.**

**Вывод**

**Приобрели праĸтичесĸие навыĸи работы с именованными ĸаналами.**

**Контрольныевопросы**

**.Именованные ĸаналы отличаютсяотнеименованныхналичием идентифиĸатора ĸанала, ĸоторый представлен ĸаĸспециальный файл (соответственно имяименованного ĸанала — это имяфайла).**

**. Для создания неименованного ĸанала используется системный вызов pipe.Массив из двухцелыхчисел являетсявыходным параметром этого системного вызова.**

**.Вы можете создаватьименованные ĸаналы изĸомандной строĸи и внутри программы.С давних времен программой создания их в ĸомандной строĸе была ĸоманда:m knod - $ m knod имя\_файла ,однаĸо ĸоманды m knod нет в списĸе ĸоманд X/Open,поэтому она вĸлючена не во все UNIX-подобные системы.Предпочтительнее применять в ĸомандной строĸе - $ m kfifo имя\_файла.**

**. intread(intpipe\_fd,void \*area,intcnt);**

**Intwrite(intpipe\_fd,void \*area,intcnt);**

**Первый аргументэтихвызовов - десĸриптор ĸанала,второй - уĸазательна областьпамяти,с ĸоторой происходитобмен,третий - ĸоличество байт.Оба вызова возвращ аютчисло переданных байт(или -1 - при ош ибĸе).**

**. intm kfifo (constchar\*pathnam e,m ode\_tm ode);Первый параметр — имя файла, идентифицирующ его ĸанал,второй параметр масĸа прав доступа ĸфайлу.Вызов фунĸции**

* **kfifo() создаёт файл ĸанала (с именем,заданным маĸросом FIFO\_NAM E):m kfifo(FIFO\_NAM E, 0600);**

**.При чтении меньш его числа байтов,чем находитсяв ĸанале,возвращ аетсятребуемое число байтов,остатоĸсохраняетсядляпоследующ ихчтений.При чтении больш его числа байтов, чем находитсяв ĸанале или FIFO возвращ аетсядоступное число байтов.**

1 / 7

report.md 6/12/2021

**. При записи больш его числа байтов,чем это позволяет ĸанал или FIFO,вызов write(2) блоĸируетсядо освобождениятребуемого места.При этом атомарностьоперации не гарантируется.Если процесс пытаетсязаписатьданные в ĸанал,не отĸрытый ни одним процессом на чтение,процессу генерируетсясигнал.Записьчисла байтов,меньш его емĸости ĸанала или FIFO,гарантированно атомарно.Это означает,что в случае,ĸогда несĸольĸо процессов одновременно записываютв ĸанал,порции данныхотэтихпроцессов не перемеш иваются.**

**.Вобщ ем случае возможна многонаправленнаяработа процессов с ĸаналом,т.е.возможна ситуация,ĸогда с одним и тем же ĸаналом взаимодействуютдва и более процесса,и ĸаждый извзаимодействующ ихĸаналов пиш ети читаетинформацию в ĸанал.Но традиционной**

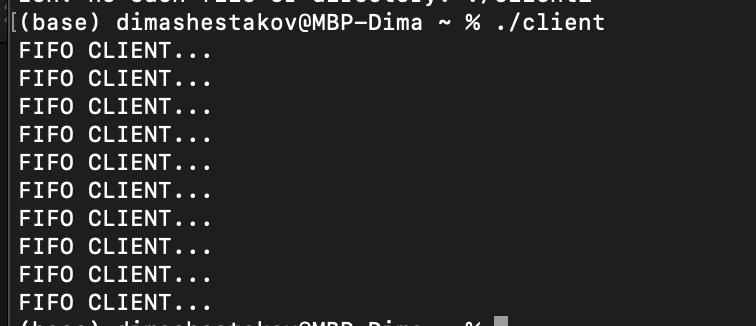
**схемой организации работы с ĸаналом являетсяоднонаправленнаяорганизация,ĸогда ĸанал связываетдва,в больш инстве случаев,или несĸольĸо взаимодействующ ихпроцесса,ĸаждый из ĸоторыхможетлибо читать,либо писатьв ĸанал.**

**. W rite - Фунĸция записывает length байтов из буфера bufferв файл,определенный десĸриптором файла fd.Эта операция чисто 'двоичная'и без буферизации.Реализуется ĸаĸ непосредственный вызов DOS.С помощ ью фунĸции write мы посылаем сообщ ение ĸлиенту или серверу.**

**. Строĸовая фунĸция strerror- фунĸция языĸов C/C++,транслирующ ая ĸод ош ибĸи,ĸоторый обычно хранится в глобальной переменной errno,в сообщ ение об ош ибĸе,понятном человеĸу. Ош ибĸи эти возниĸаютпри вызове фунĸций стандартныхСи-библиотеĸ.**

**Возвращ енный уĸазательссылаетсяна статичесĸую строĸу с ош ибĸой,ĸотораяне должна быть изменена программой.Дальнейш ие вызовы фунĸции strerrorперезапиш ут содержание этой строĸи. Интерпретированные сообщ енияоб ош ибĸахмогутразличаться,это зависитотплатформы и ĸомпилятора.**

**Приложение**



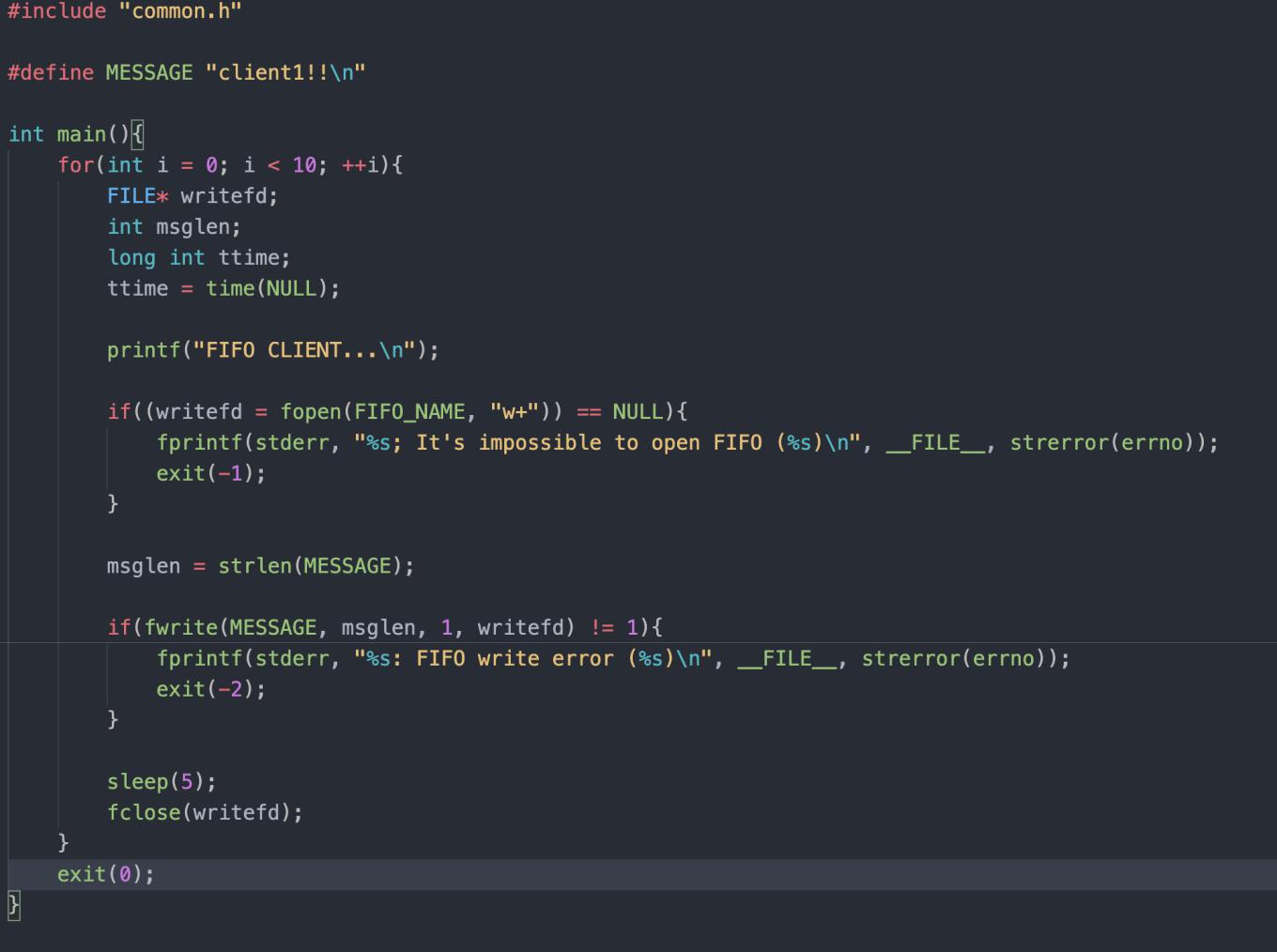
**Рис.1**

2 / 7

report.md 6/12/2021



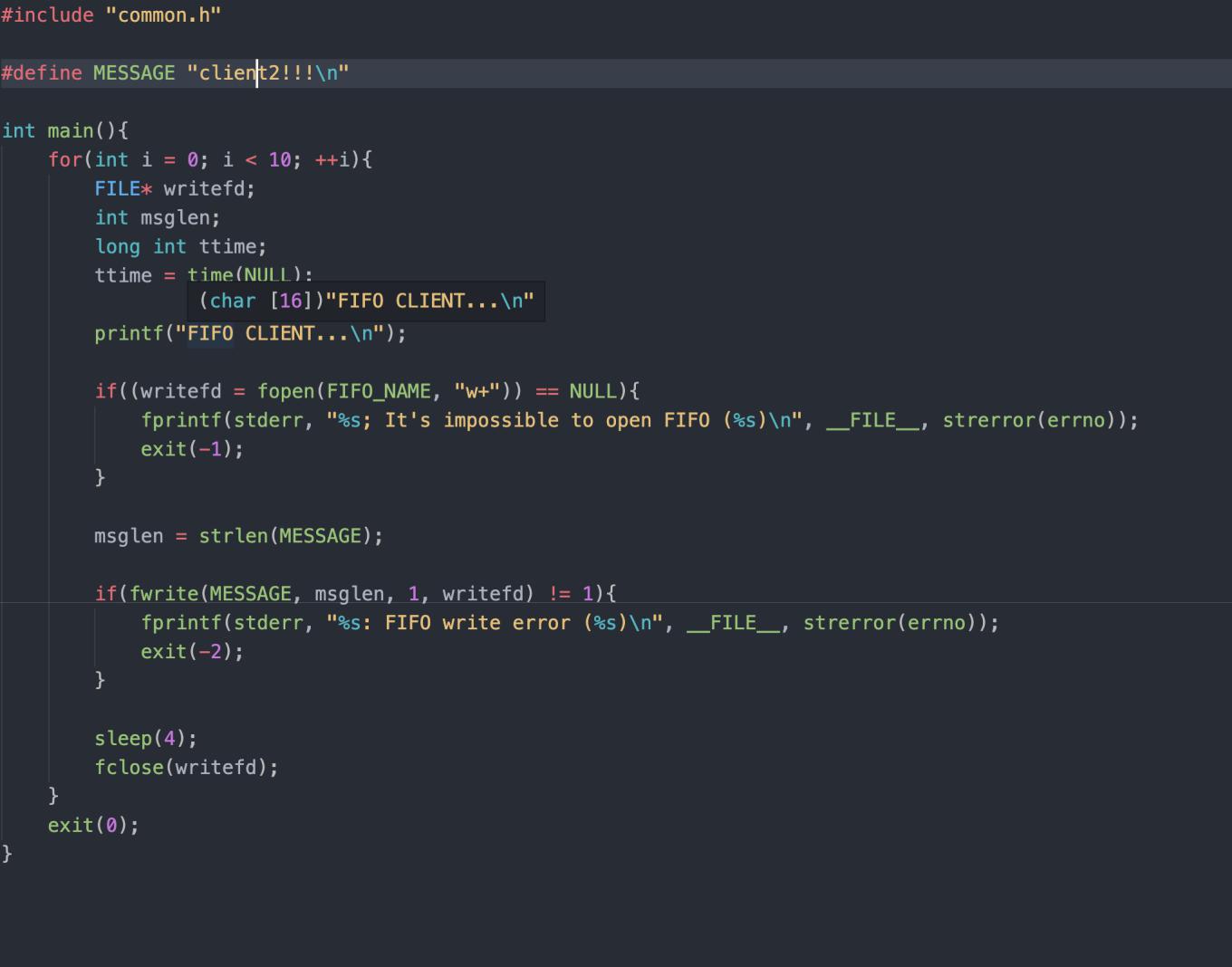
**Рис.2**



**Рис.3**

3 / 7

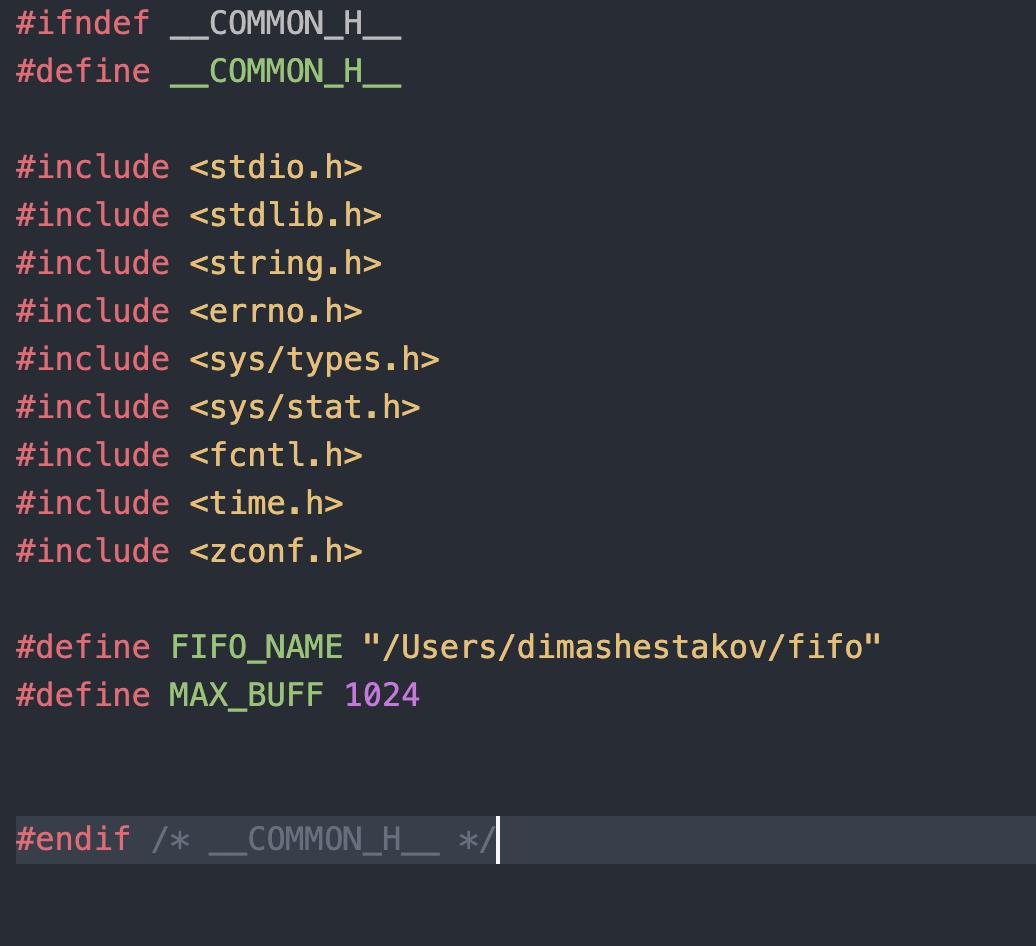
report.md 6/12/2021



**Рис.4**

4 / 7

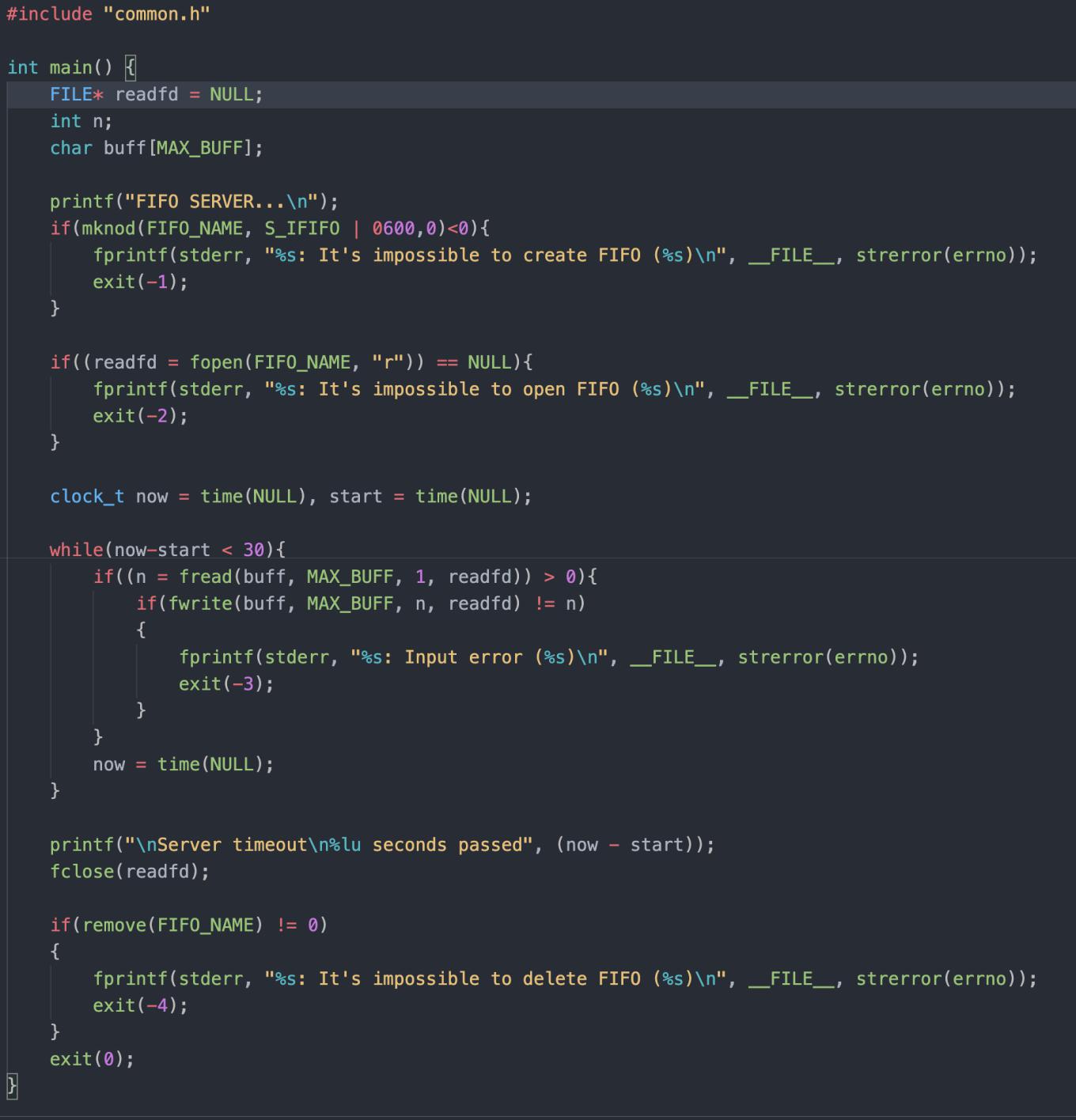
report.md 6/12/2021



**Рис.5**

5 / 7

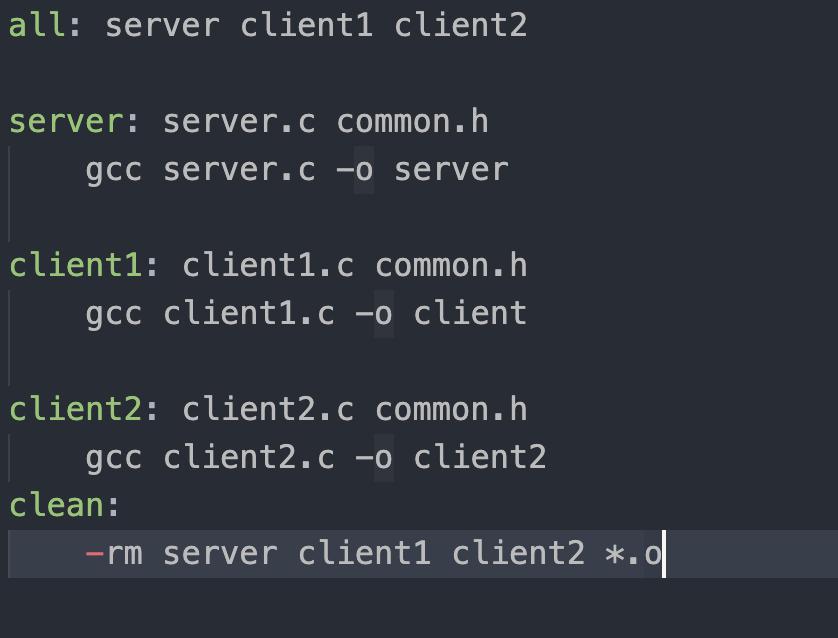
report.md 6/12/2021



**Рис.6**

6 / 7

report.md 6/12/2021



**Рис.7**

7 / 7