

## Очереди сообщений

Очереди сообщений представляют собой связный список в адресном пространстве ядра. Сообщения могут посылаться в очередь по порядку и доставаться из очереди несколькими разными путями. Каждая очередь сообщений однозначно определена идентификатором IPC.

Очереди сообщений как средство межпроцессной связи позволяют процессам взаимодействовать, обмениваясь данными. Данные передаются между процессами дискретными порциями, называемыми **сообщениями**. Процессы, использующие этот тип межпроцессной связи, могут выполнять две операции: *послать или принять сообщение*.

Процесс, прежде чем послать или принять какое-либо сообщение, должен запросить систему породить программные механизмы, необходимые для обработки данных операций. Процесс делает это при помощи системного вызова `msgget`. Обратившись к нему, процесс становится владельцем или создателем некоторого средства обмена сообщениями; кроме того, процесс специфицирует первоначальные права на выполнение операций для всех процессов, включая себя. Впоследствии владелец может уступить право собственности или изменить права на операции при помощи системного вызова `msgctl`, однако на протяжении всего времени существования определенного средства обмена сообщениями его создатель остается создателем. Другие процессы, обладающие соответствующими правами для выполнения различных управляющих действий, также могут использовать системный вызов `msgctl`.

Процессы, имеющие права на операции и пытающиеся послать или принять сообщение, *могут приостанавливаться*, если выполнение операции не было успешным. В частности это означает, что процесс, пытающийся послать сообщение, может ожидать, пока процесс-получатель не будет готов; и наоборот, получатель может ждать отправителя. Если указано, что процесс в таких ситуациях должен приостанавливаться, говорят о выполнении над сообщением *``операции с блокировкой``*. Если приостанавливать процесс нельзя, говорят, что над сообщением выполняется *"операция без блокировки"*.

Процесс, выполняющий операцию с блокировкой, может быть приостановлен до тех пор, пока не будет удовлетворено одно из условий:

- операция завершилась успешно;
- процесс получил сигнал;
- очередь сообщений ликвидирована.

Системные вызовы позволяют процессам пользоваться этими возможностями обмена сообщениями. Вызывающий процесс передает системному вызову аргументы, а системный вызов выполняет (успешно или нет) свою функцию. Если системный вызов завершается успешно, он выполняет то, что от него требуется, и возвращает некоторую содержательную информацию. В противном случае процессу возвращается значение -1, известное как признак ошибки, а внешней переменной `errno` присваивается код ошибки.