**Логическая организация механизма передачи информации:**

1. Как устанавливается связь
2. Можно ли использовать средства связи сразу после создания процесса

Для использования общей памяти разными процессами потребуется специальное обращение к ОС, которая выделит необходимую область адресного пространства.

Но для передачи сигнала от одного процесса к другому никакая инициализация не нужна.

В то же время передача информации по линиям связи может потребовать первоначального резервирования такой линии для процессов, желающих обменяться информацией.

Различают 2 способа адресации:

1. Прямую

Взаимодействующие процессы непосредственно обращаются друг с другом, при каждой операции обмена данными явно указывая имя или номер процесса, которому информация предназначена или от которого она должна быть получена.

Если и процесс, от которого данные исходят, и процесс, принимающий данные, указывают имена своих партнеров по взаимодействию, то такая схема адресации называется симметричной прямой адресацией.

Ни один другой процесс не может вмешаться в процедуру симметричного прямого общения двух процессов, перехватить посланные или подменить ожидаемые данные.

Если только один из взаимодействующих процессов, например передающий, указывает имя своего партнера по кооперации, а второй процесс в качестве возможного партнера рассматривает любой процесс в системе, например ожидает получения информации от произвольного источника, то такая схема адресации называется асимметричной прямой адресацией

1. Непрямую

При непрямой адресации данные помещаются передающим процессом в некоторый промежуточный объект для хранения данных, имеющий свой адрес, откуда они могут быть затем изъяты каким-либо другим процессом.

При этом передающий процесс не знает, как именно идентифицируется процесс, который получит информацию, а принимающий процесс не имеет представления об идентификаторе процесса, от которого он должен ее получить.

При использовании непрямой адресации инициализация средства связи может и не требоваться.

**Информационная валентность процессов и средств связи**

Под однонаправленной связью мы будем понимать связь, при которой каждый процесс, ассоциированный с ней, может использовать средство связи либо только для приема информации, либо только для ее передачи.

При двунаправленной связи каждый процесс, участвующий в общении, может использовать связь и для приема, и для передачи данных. В коммуникационных системах принято называть однонаправленную связь симплексной, двунаправленную связь с поочередной передачей информации в разных направлениях – полудуплексной, а двунаправленную связь с возможностью одновременной передачи информации в разных направлениях – дуплексной.

**Особенности передачи информации с помощью линий связи**

Типы буфера канала связи:

1. Буфер нулевой ёмкости или отсутствует

Никакая информация не может сохраняться на линии связи.

В этом случае процесс, посылающий информацию, должен ожидать, пока процесс, принимающий информацию, не соблаговолит ее получить, прежде чем заниматься своими дальнейшими делами.

1. Буфер ограниченной ёмкости

Размер буфера равен n, то есть линия связи не может хранить до момента получения более чем n единиц информации. Если в момент передачи данных в буфере хватает места, то передающий процесс не должен ничего ожидать. Информация просто копируется в буфер. Если же в момент передачи данных буфер заполнен или места недостаточно, то необходимо задержать работу процесса отправителя до появления в буфере свободного пространства.

1. Буфер неограниченной ёмкости

Теоретически это возможно, но практически вряд ли реализуемо. Процесс, посылающий информацию, никогда не ждет окончания ее передачи и приема другим процессом.

Поток ввода-вывода и сообщения.  
Существует 2 модели передачи информации по каналу связи:

1. Поток ввода-вывода
2. Сообщения

При передаче с помощью потоковой модели операции передачи/приёма не интересуется содержимым данным.

**Надежность средств связи**

Мы будем называть способ коммуникации надежным, если при обмене данными выполняются четыре условия.

1. Не происходит потери информации.

2. Не происходит повреждения информации.

3. Не появляется лишней информации.

4. Не нарушается порядок данных в процессе обмена.

Очевидно, что передача данных через разделяемую память является надежным способом связи.

То, что мы сохранили в разделяемой памяти, будет считано другими процессами в первозданном виде, если, конечно, не произойдет сбоя в питании компьютера.

Для других средств коммуникации, как видно из приведенных выше примеров, это не всегда верно.

Для обнаружения повреждения информации будем снабжать каждое передаваемое сообщение некоторой контрольной суммой, вычисленной по посланной информации.

При приеме сообщения контрольную сумму будем вычислять заново и проверять ее соответствие пришедшему значению.

Если данные не повреждены (контрольные суммы совпадают), то подтвердим правильность их получения.

Если данные повреждены (контрольные суммы не совпадают), то сделаем вид, что сообщение к нам не поступило.

Вместо контрольной суммы можно использовать специальное кодирование передаваемых данных с помощью кодов, исправляющих ошибки.

Такое кодирование позволяет при числе искажений информации, не превышающем некоторого значения, восстановить первоначальные неискаженные данные.

Если по прошествии некоторого интервала времени подтверждение правильности полученной информации не придет на передающий конец линии связи, будем считать информацию утерянной и пошлем ее повторно.

Для того чтобы избежать двойного получения одной и той же информации, на приемном конце линии связи должен осуществляться соответствующий контроль.

Для гарантии правильного порядка получения сообщений будем их нумеровать.

При приеме сообщения с номером, не соответствующим ожидаемому, поступаем с ним как с утерянным и ждем сообщения с правильным номером.

Подобные действия могут быть возложены:

* на операционную систему;
* на процессы, обменивающиеся данными;
* совместно на систему и процессы, разделяя их ответственность.

**Нити исполнения**

Если алгоритм решения задачи обладает определенным внутренним параллелизмом, мы могли бы ускорить его работу, организовав взаимодействие нескольких процессов.