**RPC** – класс технологий, позв комп. прогам вызывать ф-ции или процедуры в другом адресном пр-ве (обычно на удаленных пк)

**Реализация RPC вкл:**\* сет. протокол для обмена в режиме К-С  
\* язык сериализации объектов

На трансп уровне RPC используют в основном протоколы TCP и UDP, но нек построены на основе HTTP (это нарушает архит ISO/OSI, т.к. HTTP изначально не транспортный протокол)

**Хар. черты вызова лок. процедур:**  
\* асимметричность – 1 из взаимод. сторон явл инициатором

\* синхронность – вып вызыв-щей процедуры приостан с момента выдачи запроса и возобн только после возврата из вызываемой процедуры

**Отличия RPC вызова от локального:**  
\* параметры не д сод ук-ли на ячейки нестек. памяти

\* обяз использ нижележащую систему связи

#include “stdafx.h”

int func (int n, char c[], int k);

int main()

int n = 50, k = 0;

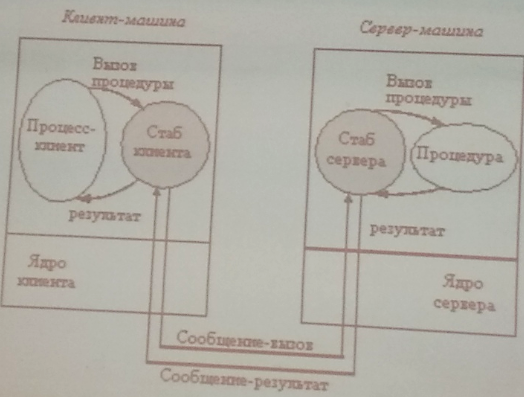
char buf[50];

k = func (n, buf, k);

return 0;

Указатель стека:  
--> лок. переменные функции main  
--> лок. переменные функции main | k | buf | n | лок. переменные функции func

**Remote Procedure Call:**



**Посл-сть выполнения операций при вызове удал. процедуры:**

\* процедура клиента вызыв. клиентскую заглушку

\* клиентская заглушка созд. сообщение и вызыв. службу RPC локальной ОС

\* Служба RPC пересыл сообщ службе RPC удаленной ОС

\* Служба RPC удаленной ОС вызыв серверную заглушку и передает ей сообщ

\* Серверная заглушка извлек из сообщ параметры и вызыв удал процедуру

\* Удал процедура исполн код и возвр параметры и знач возрвта серверной заглушке

\* Серв. заглушка формир сообщ и вызыв службу RPC своей локальной ОС  
\* Служба RPC своей локальной ОС сервера пересыл сообщ службе RPC локальной ОС клиента

\* Служба RPC локальной ОС клиента возвр сообщ клиентской заглушке

\* Заглушка извлек результаты и возвр их

Клиент -- - -- - - ------------------ ожид результата - -- ------- -- - - -- - -- - - -

вызов удал процедуры/ \ / \завершение вызова удал процедуры

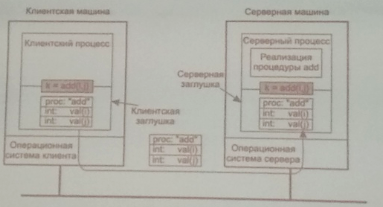
\ /

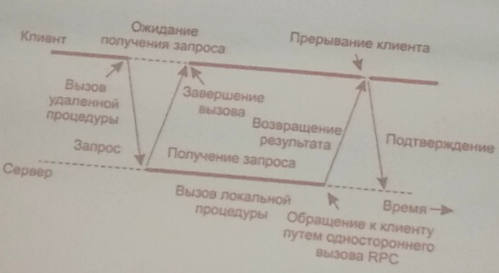
запрос \ /ответ

\ /

Север - -- - -- - --------------------------------------------------------------------------

вызов лок. серв. проц / \заверш работы серв проц





Процесс передачи параметров при вызове удал. процедуры наз. *Маршалингом параметров*

count = read (fd, buf, nbytes)

Вызывающая процедура заталк параметры в стек в обр. порядке

после того, как вызов read выполнен, он помещ возвращаемое значение в регистр, перемещает адрес возврата и возвр управление вызывающей процедуре, кот. выбирает параметры из стека, возвращая его в исх. состояние

в С м. вызыв по ссылке или по значению

по отн к выз-мой процедуре параметры-значения явл инициализируемыми лок. пер-ными

**Определение интерфейса между вызывающей программой и клиентской и серверной заглушками:**

**IDL (Interface Definition Language)** – кроссплатформаенный язык (спецификаций) определения интерфейсов

Разновидности IDL:

\* OSF DCE RPC IDL

\* CORHA IDL – разработан консорциумом OMG

\* COM IDL – разработан Microsoft (подмн-во CORBA IDL)

\* MIDL – Microsoft IDL (встроен в VisualStudio)

\* Java IDL – Sun Microsystems (совпад со спецификацией OMG)

**Семантика RPC в случае отказов:**

1. Клиент не может опред местонахождение сервера

2. Потерян запрос от К к С

3. Потеряно ответное сообщ от С к К

4. Сервер потерпел аварию после получения запроса. Сущ. 3 подхода к этой проблеме:

\* ждать до тех пор, пока сервер не перезагрузиться и пытаться выполнить операцию снова. Этот подход гарантирует, что RPC был выполнен до конца по крайней мере один раз, а возм. и более

\* сразу сообщить приложению об ошибке. этот подход гарант, что RPC Был выполнен не более 1 раза

\* не гарант ничего. Когда С отказывает, клиенту не оказ. никакой поддержки. RPC мб или не выполнен вообще, или выполнен много раз. Во всяком случае этот способ очень легко реализовать