Лекция №1 13.02.17

2Дз – второе получаем после сдачи первого.

Второе дз до 13 недели.

3 модуля:

1-й: 8-я неделя, сдать 3ЛР+РК1, 30 баллов, 9-я неделя – ДЗ

2-Й: 10-я неделя – РК2

3-й: 15-я неделя – зачет, если есть 35 баллов и все сдано.

План

1. Функции сетевого ПО
2. Сетевые архитектуры
3. Компоненты сетевого ПО
4. Архитектура клиент-серверных приложений
5. Промежуточное программное обеспеченье

**Функции системного сетевого ПО**

* управление сетью

Есть соответствующие протоколы, собирающие информации о нагрузке и о трафиках. С помощью агентов фиксируются определенные аномалии сети.

* управление ресурсами сети

Управление ресурсами, распределенными в сети

* коммуникационные функции

Адресация, буферизация, маршрутизация, управление потоками данных

* управление каталогами и файлами сети

Управление каталогами и файлами по сети. Удаленные файловые системы базируются на локальных файловых системах.

* защита от несанкционированного доступа

Разрешение доступа к определенным данным только в оговоренное время, какое-то количество раз. У каждого пользователя, работающего в сети, имеются определенные привилегии.

* обеспечение отказоустойчивости

Отказоустойчивость – сохранение устойчивости при наличии внешних негативных факторов. Для серверов отказоустойчивость обеспечивается автономными источниками питания. Так же есть дублирование информации (либо аппаратное либо программное решение).

**Сетевые архитектуры**

Основная архитектура – ПО, которое обеспечивает независимые компьютеры и работоспособность таких приложений, как почта, передача фалов и т.д. Например, TCP/IP.

Архитектура на основе сетевой операционной системы. Есть рабочие станции и один или несколько серверов, которые обеспечивают управление этой сетью и функционирование сетевых приложений и доступ к ресурсам сети. В этом случае каждый пользователь имеет собственную ОС, а сетевая ОС является дополнением к локальной ОС, которая обеспечивает взаимодействие локальной и сетевой ОС.

Архитектура на основе распределенной ОС. В этом случае используется общая ОС. С точки зрения пользователя эта ОС видится, как локальная. Такая распределенная ОС обеспечивает прозрачность для всех пользователей.

**Компоненты СПО**

1. Сетевые ОС. Выделенные – это ОС, находящиеся на определенном узле (выделенный сервер с выделенной ОС). Невыделенные – могут работать как сервер ОС или как рабочая станция.

2. Сетевые службы ОС.

Так же как локальные службы существуют и сетевые. Первые операционные системы Microsoft не имели сетевых служб, т.к. они были ориентированы на работу с сетевым клиентом. Есть два типа ОС:

1)содержащие сетевые службы,

2) работающие с сетевым клиентом.

3. Сетевые приложения.

Два варианта приложений: отдельные программы, работающие на разных узлах сети, но решающие общую проблему, либо программы как части одной программы.

Задачи организации работы сетевых приложений:

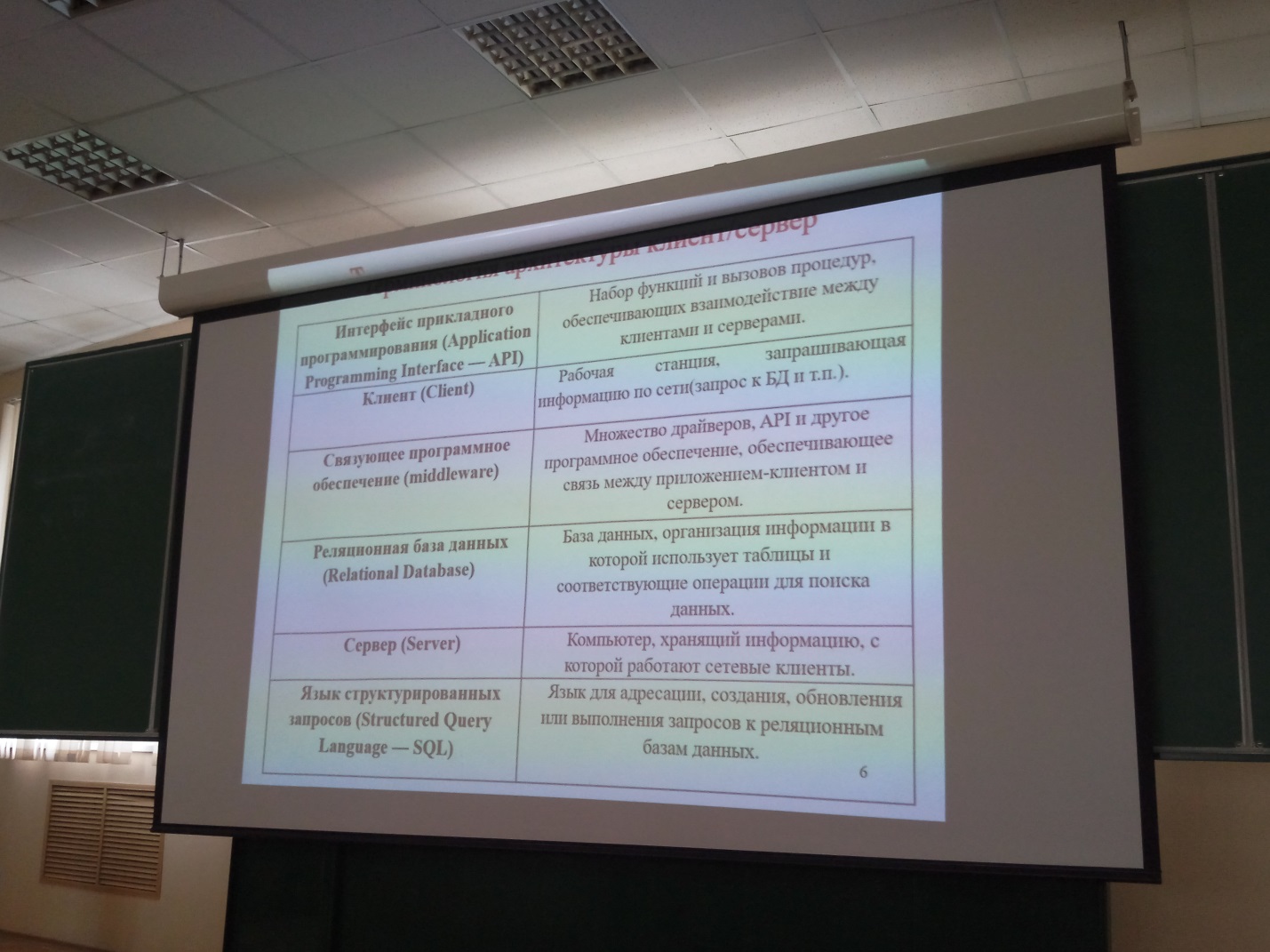
1. способ разделения приложений на части, выполняющихся на разных компьютерах сети,
2. выделение специализированных узлов в сети, на которых выполняются некоторые общие для всех приложений функции

4. Драйверы сетевых адаптеров.

Аппаратно связь осуществляется с помощью сетевых адаптеров.

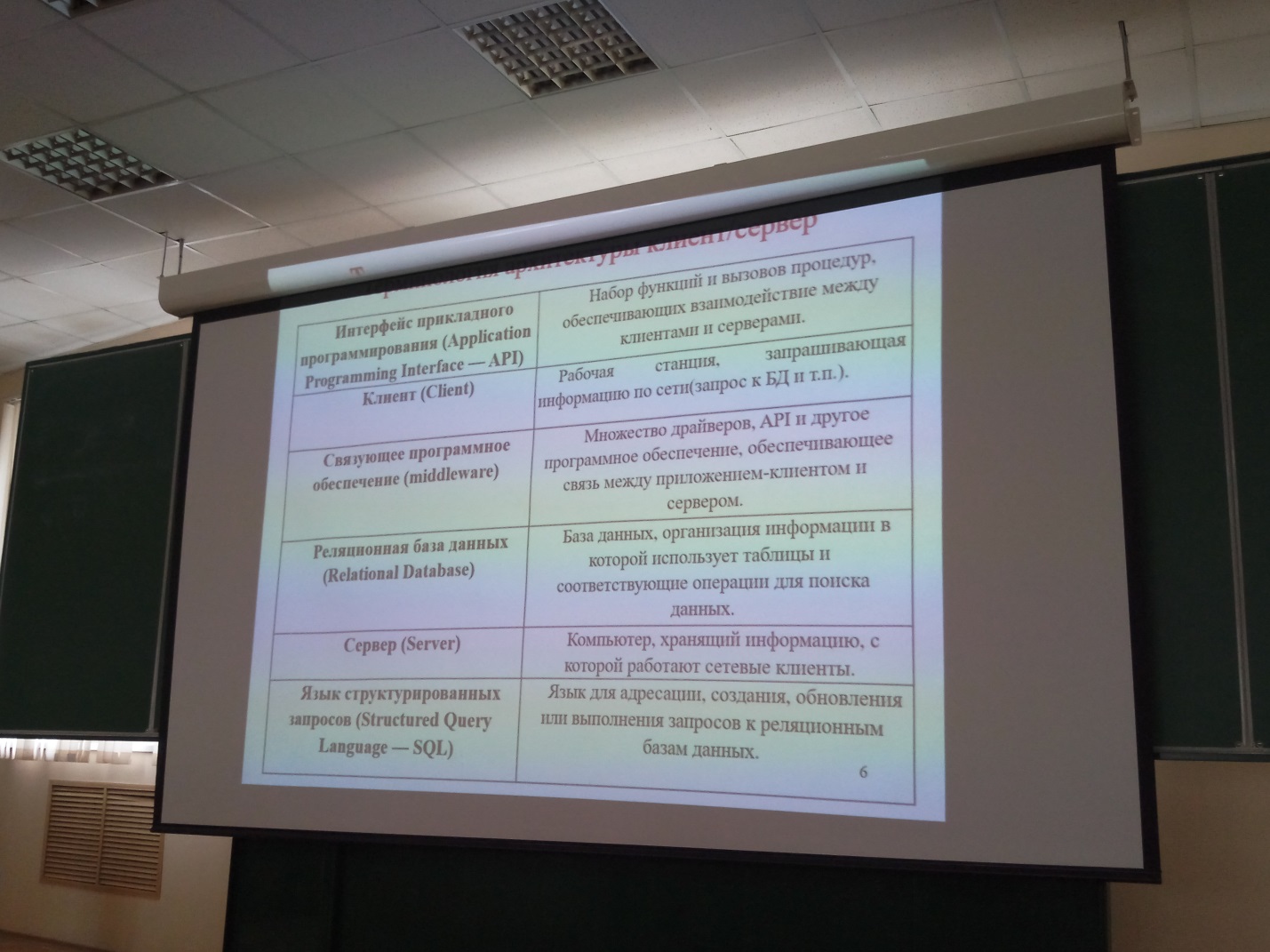
**Архитектура клиент-серверных приложений**

1. Терминология архитектуры клиент/сервер.

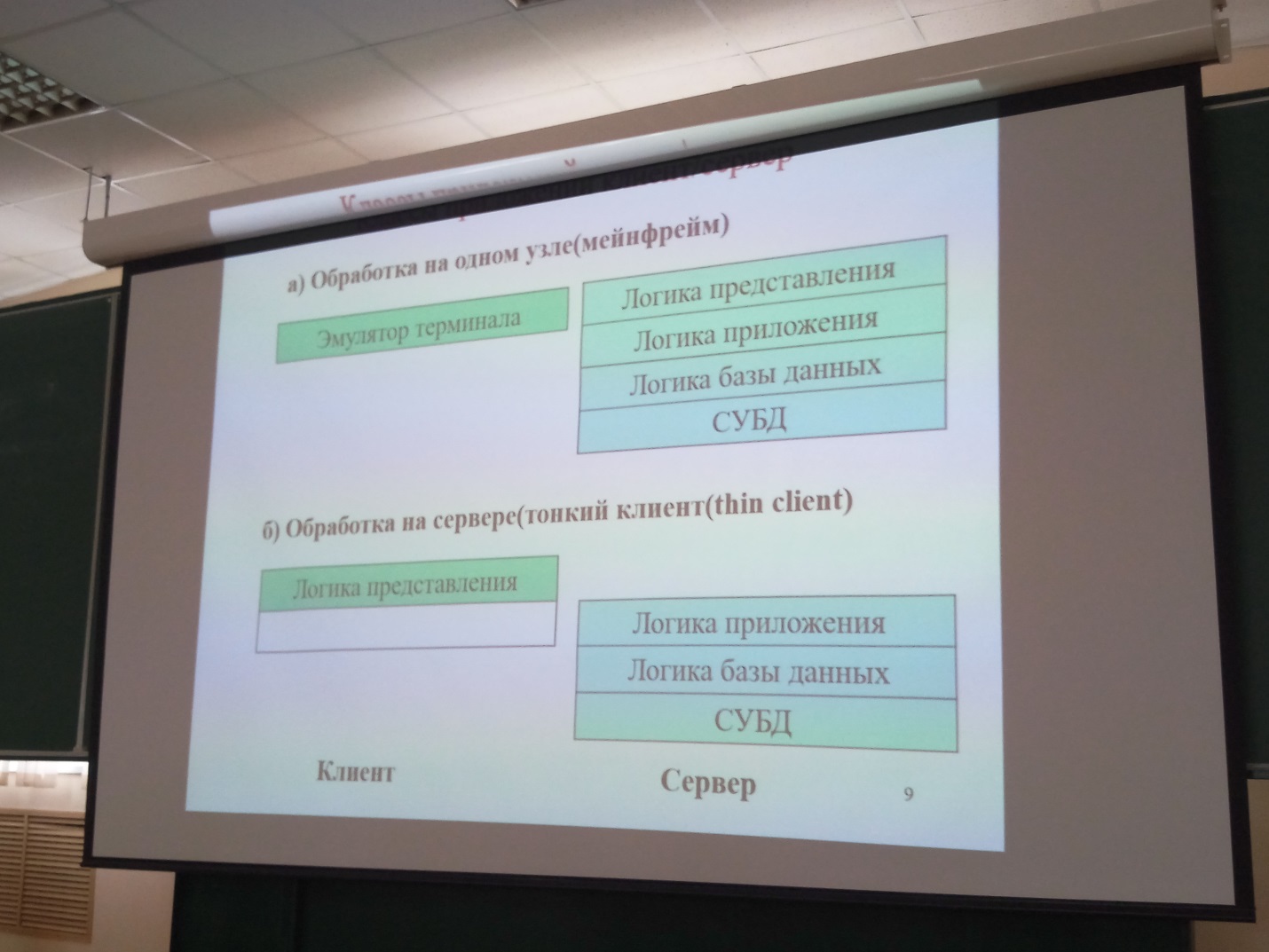


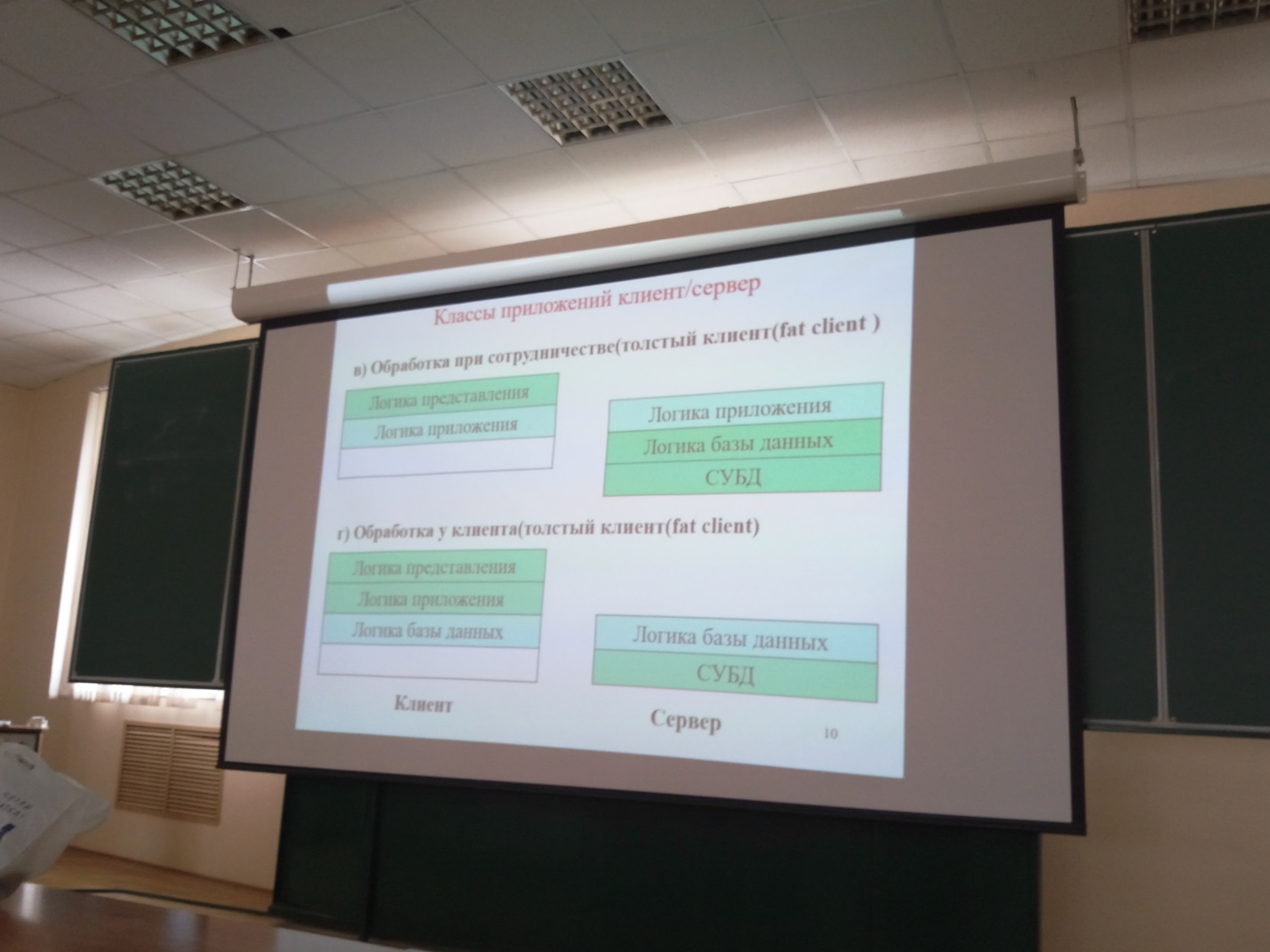
2. Обобщенная архитектура клиент/сервер.

Все взаимодействие между клиентом и сервером на уровне коммуникационного ПО. Клиент и сервер имеют собственные ОС и собственные платформы.



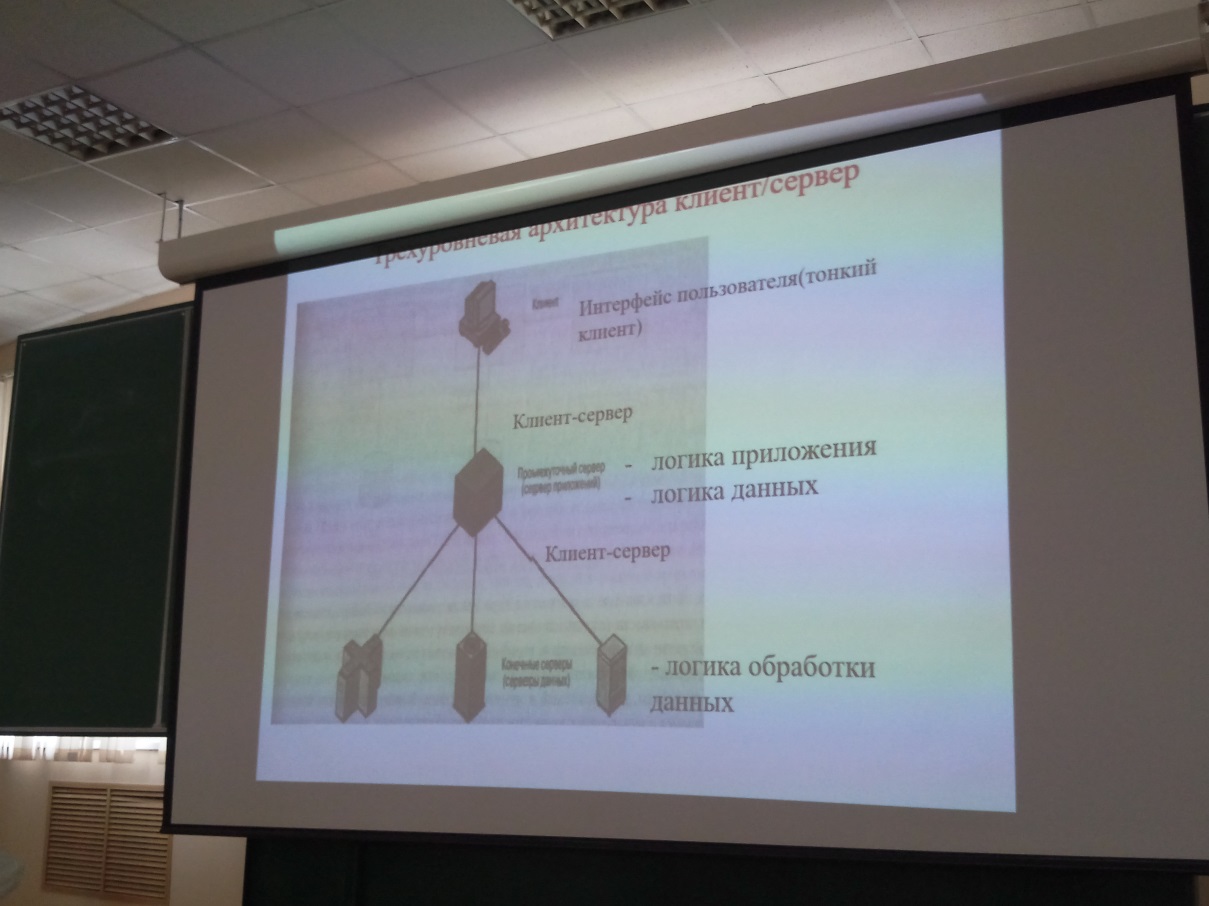
3. Классы приложений клиент/сервер.





4. Трехуровневая архитектура клиент/сервер.

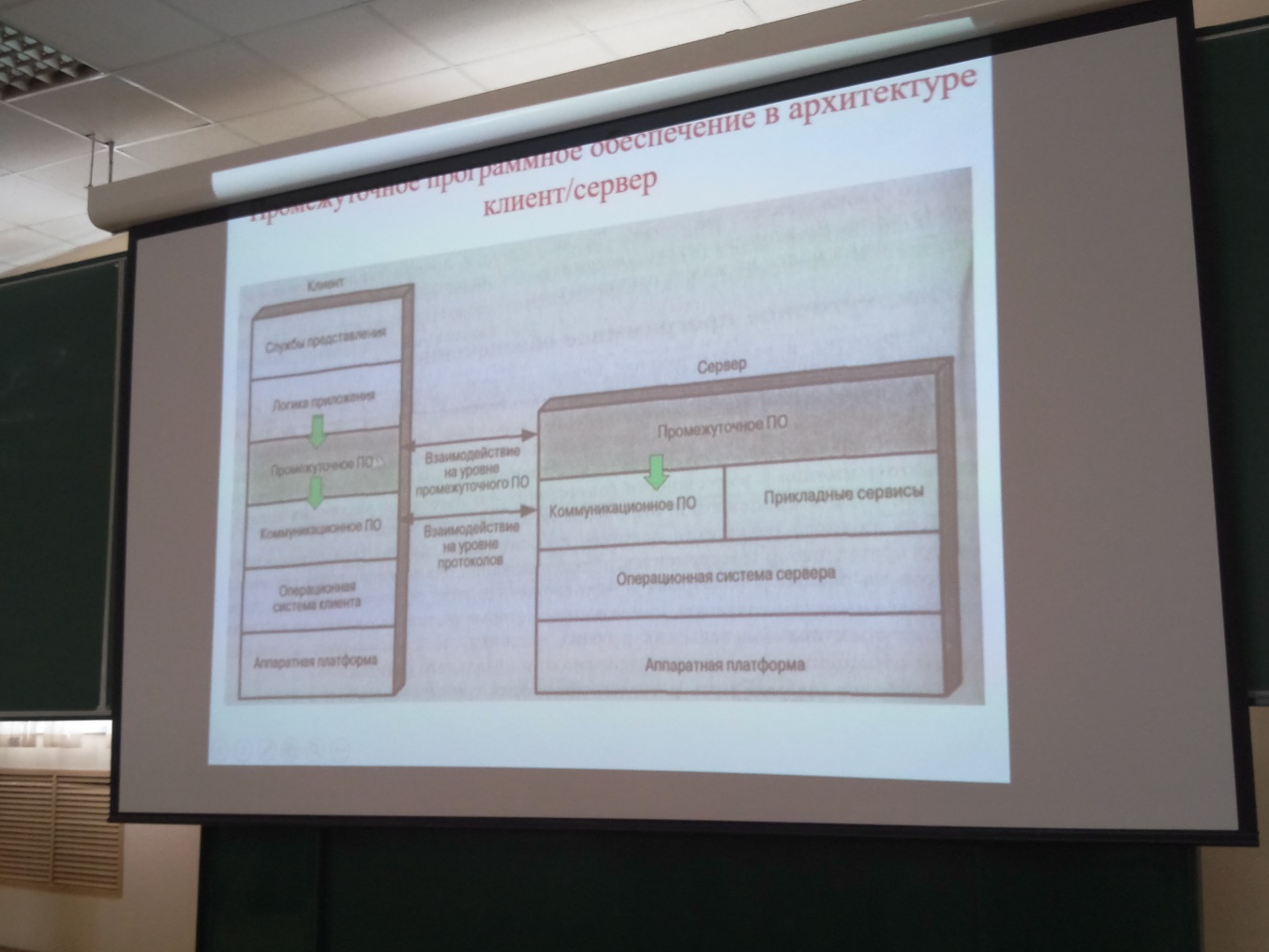
Такую архитектуру можно рассматривать как совокупность двух архитектур клиент/сервер. Промежуточный сервер взаимодействует с конечными серверами (серверами данных). Наиболее мошной должна быть часть сервера приложений (промежуточный сервер).



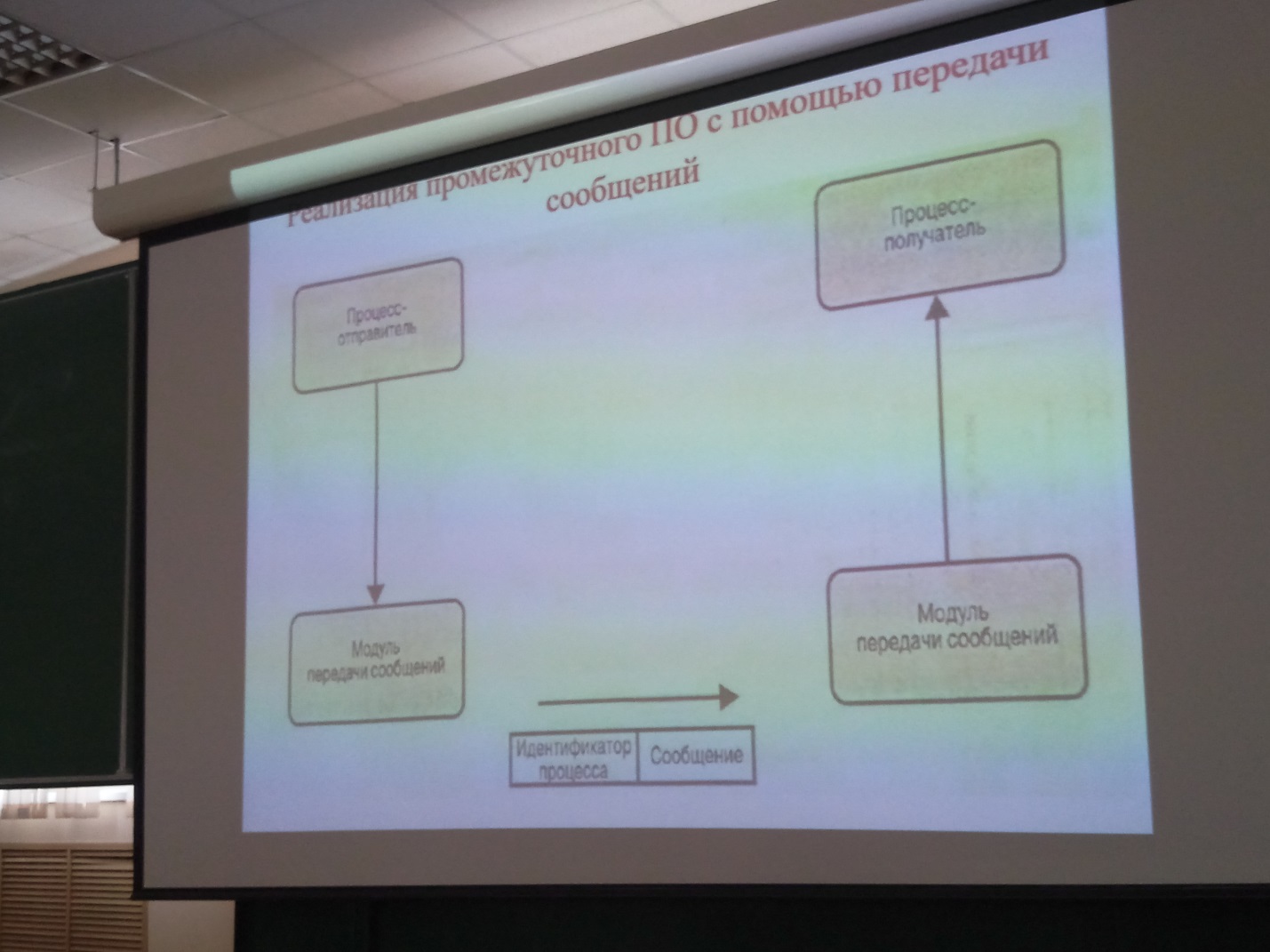
**Промежуточное программное обеспечение**

1. Базовые механизмы реализации промежуточного ПО.

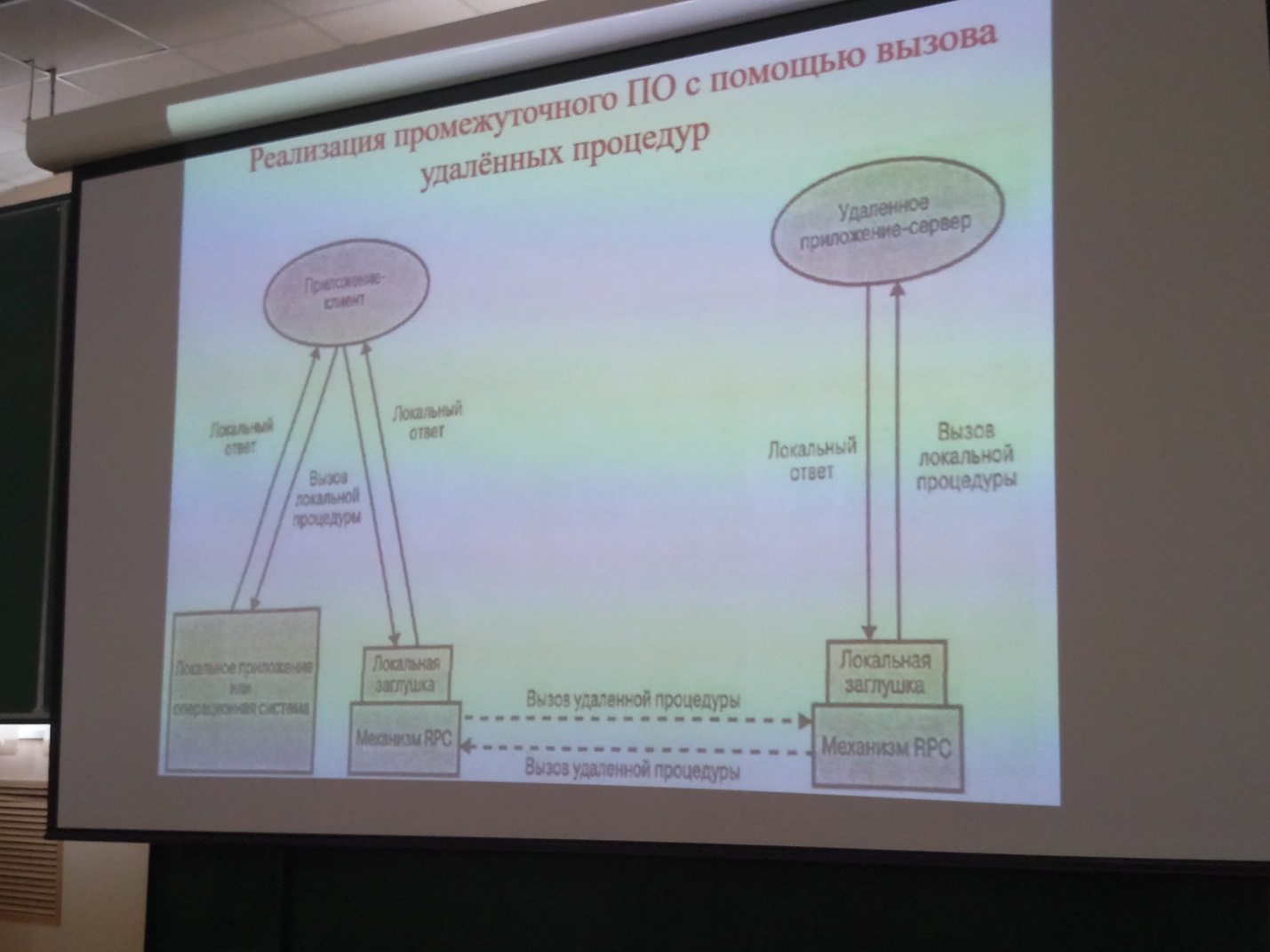
2. Промежуточное ПО в архитектуре клиент/сервер.



3. Реализация промежуточного ПО с помощью передачи сообщений.



4. Реализация промежуточного ПО с помощью вызова удаленных процедур.



5. Реализация промежуточного ПО с помощью объектно-ориентированного механизма.

