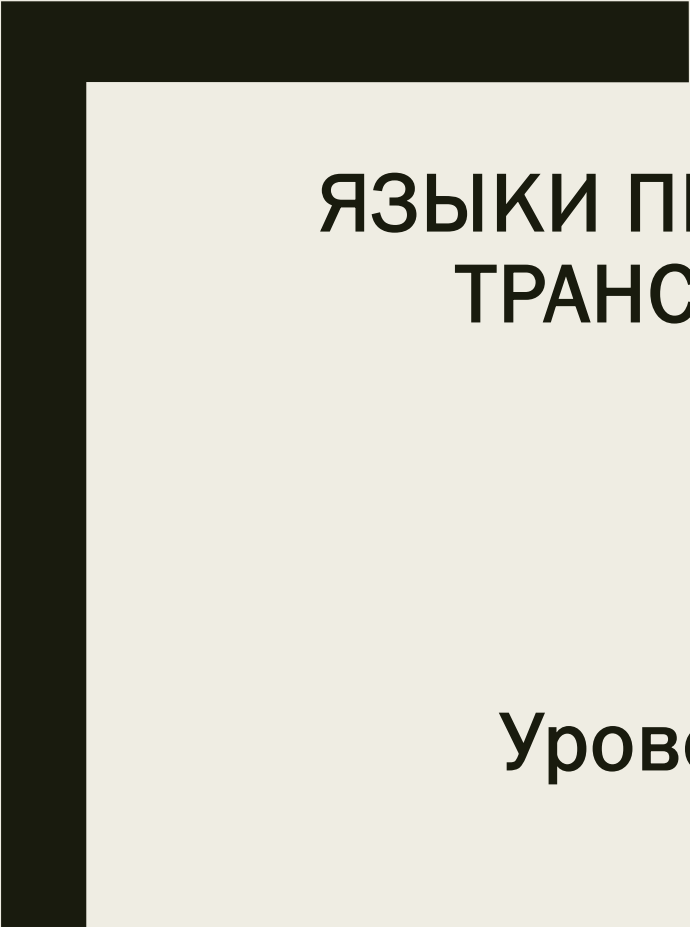
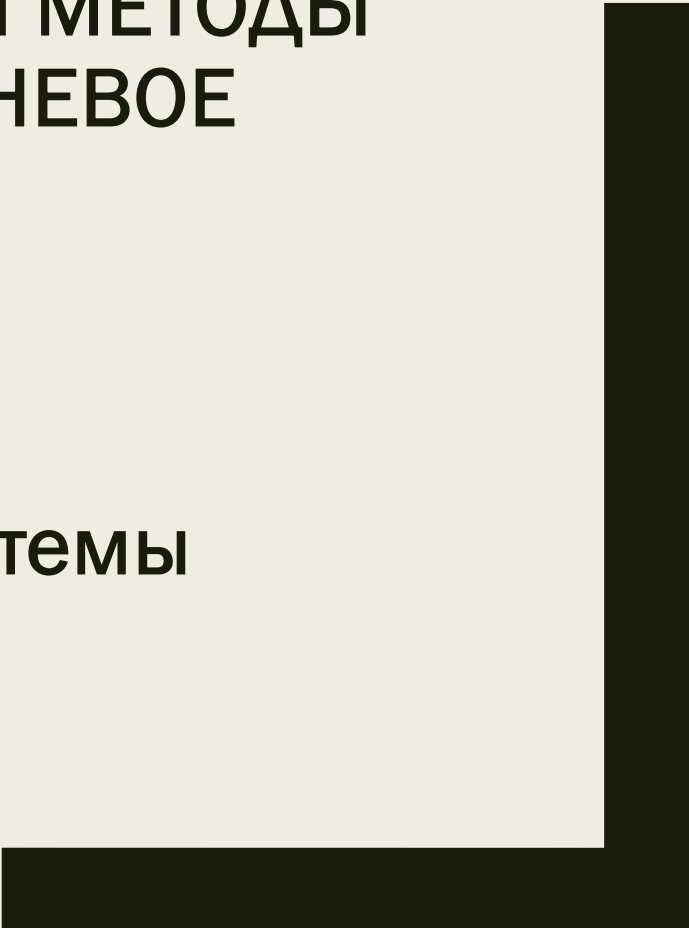




ЯЗЫКИ ПРОГРАММИРОВАНИЯ И МЕТОДЫ ТРАНСЛЯЦИИ / НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ

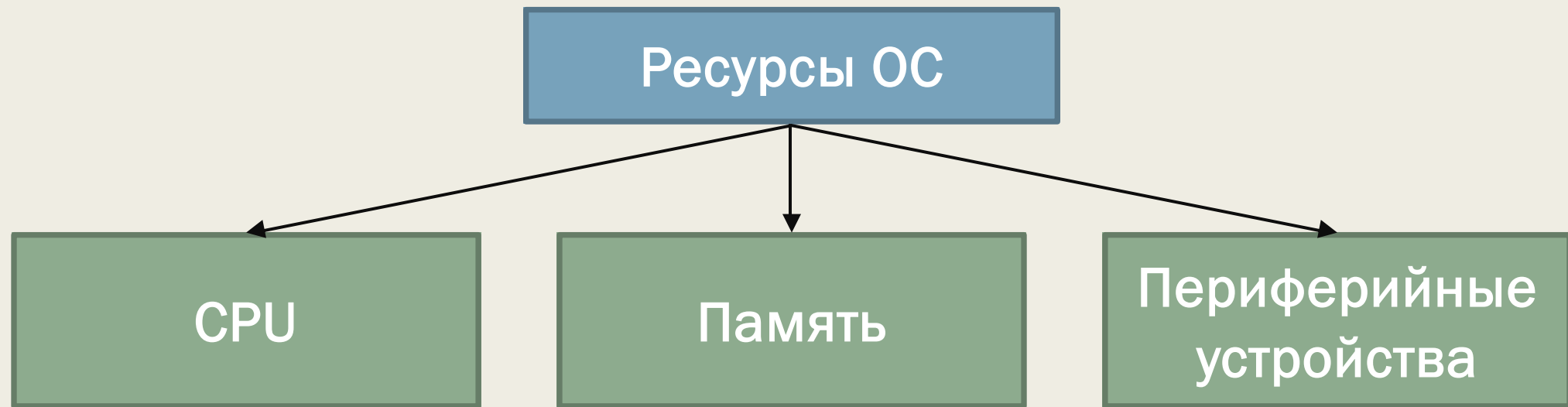
ЛЕКЦИЯ 4

Уровень операционной системы



Общая характеристика ОС

- Основное назначение ОС – управление ресурсами, абстракция оборудования и унификация доступа к нему, предоставление расширенных возможностей управления процессором и периферией



Задача разделения времени



Управление периферийными устройствами

- Периферийные устройства: любые устройства ввода-вывода или передачи данных (например, сетевая плата)
- Управление п.у. – упорядочение во времени или приоритезация исполнения тех или иных обработчиков на внешние запросы

Виды операционных систем

- **Общего назначения** (Windows, Linux, MacOS) – все ОС, которые обычно используются в повседневной жизни
- **Сетевые** – используются в маршрутизаторах или серверах
- **Системы хранения** – оптимизированы с точки зрения дисковых операций
- **Системы реального времени** – гарантируют время на обработку прерываний

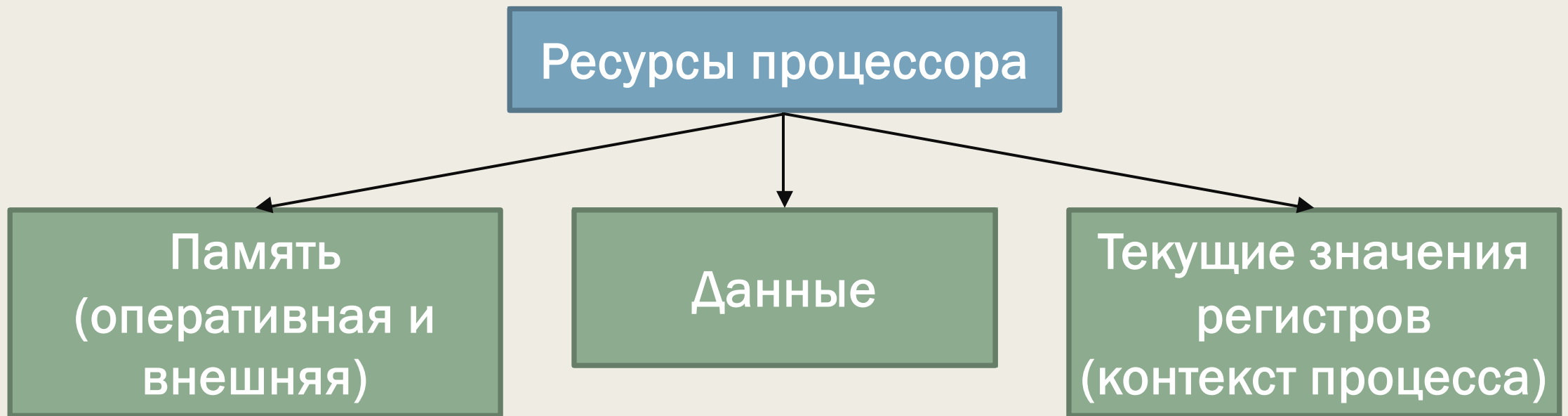
Адресное пространство

- В узком смысле: множество адресов, которые может принимать тот или иной указатель
- В широком смысле: диапазон всевозможных адресов или диапазон доступных имен

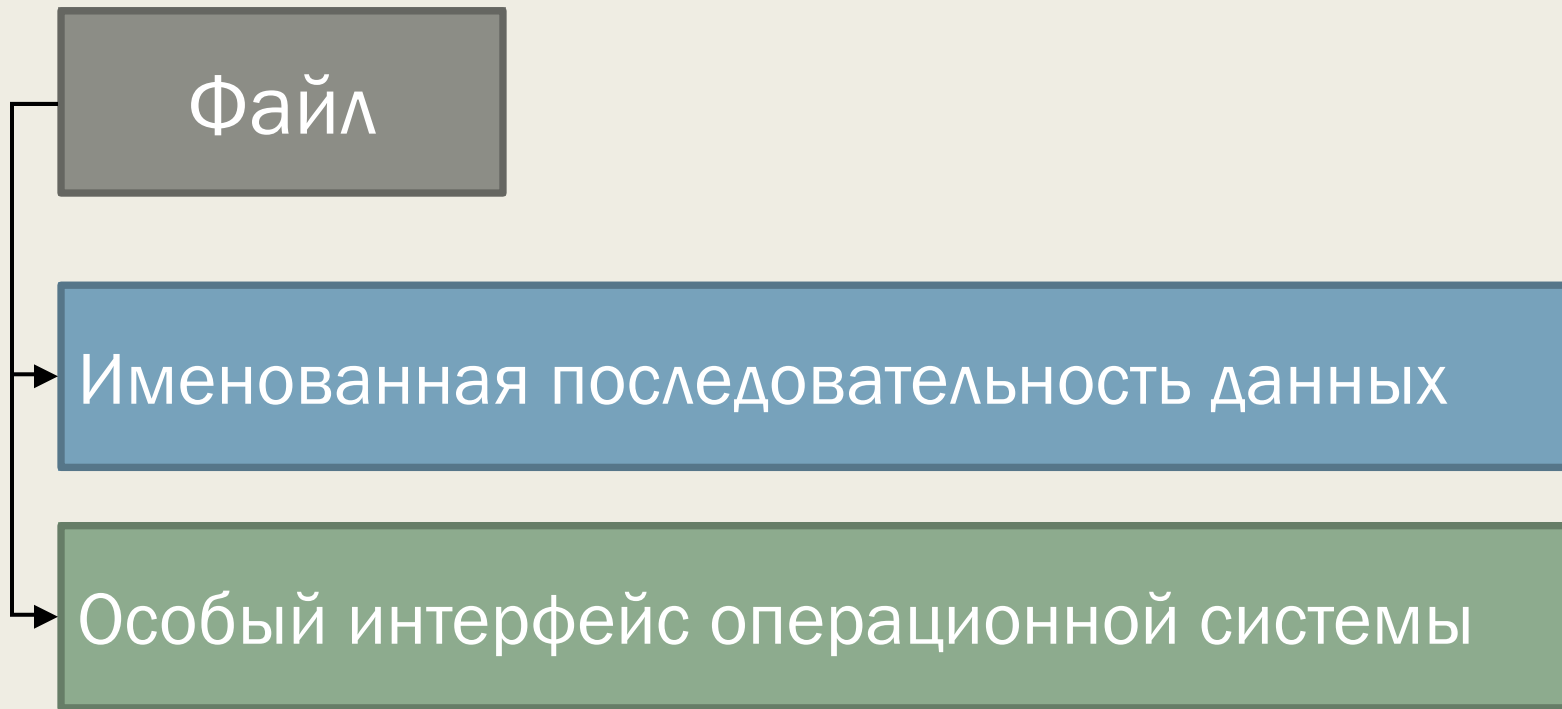
```
void main()  
{  
    int a, b;  
    float c;  
    ...  
}
```

Процесс в операционной системе

- Процесс (в общем смысле) – программа + все связанные с ней ресурсы



Файл в операционной системе



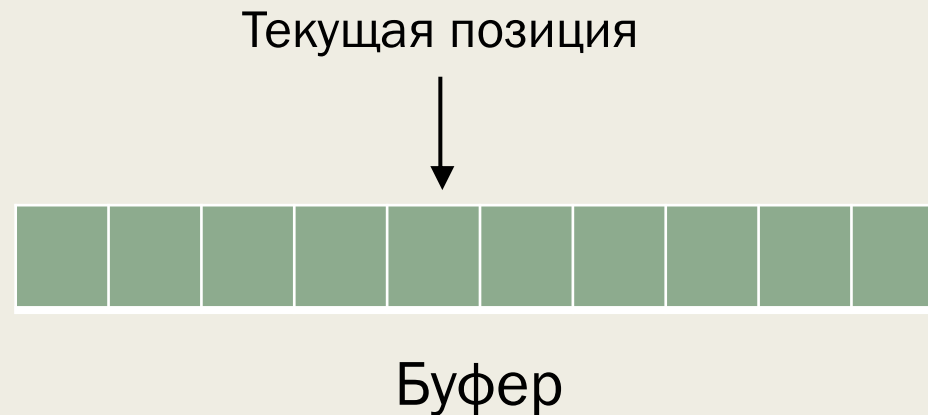
Файл – универсальный интерфейс доступа к данным и к устройствам

Атрибуты файла

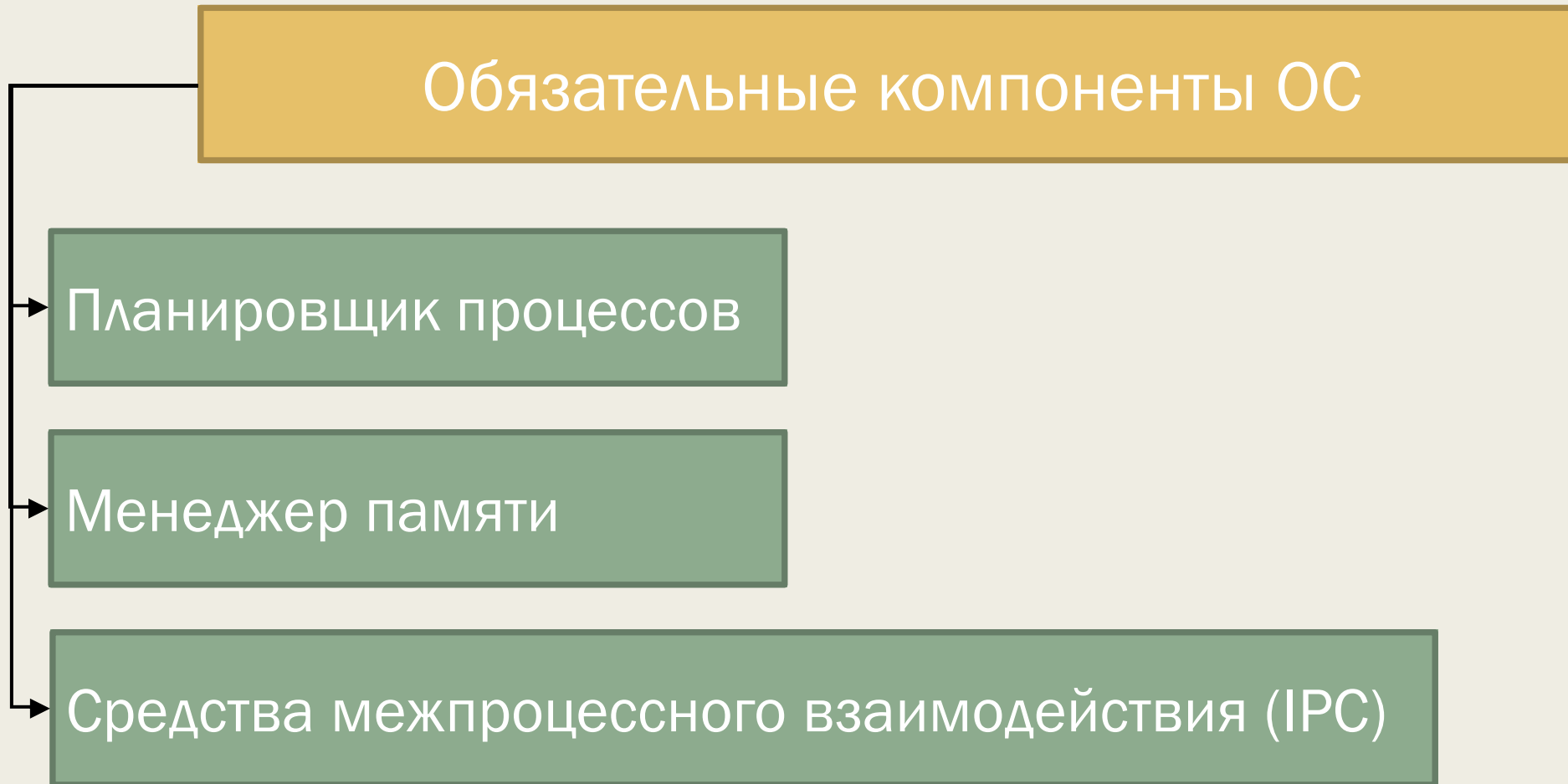
- Имя
- Флаги доступа (режим доступа – байтовый/блоковый)
- Тип
- Время (создания, последнего изменения, последнего доступа и т.д.)
- Атрибут исполняемого файла (x)
- Атрибут возможности архивации (a)
- Текущая позиция

Интерфейс управления файлами

- `ioctl` (I/O Control) – системный интерфейс, позволяющий определить режим использования файла
- Один из вариантов использования `ioctl` – управление размером буфера



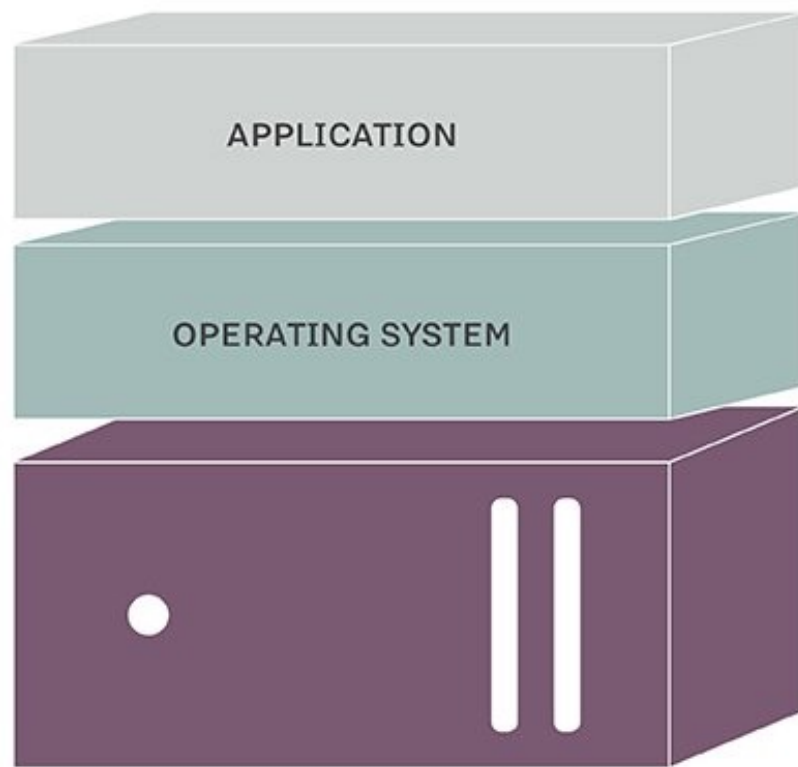
Базовая архитектура ОС



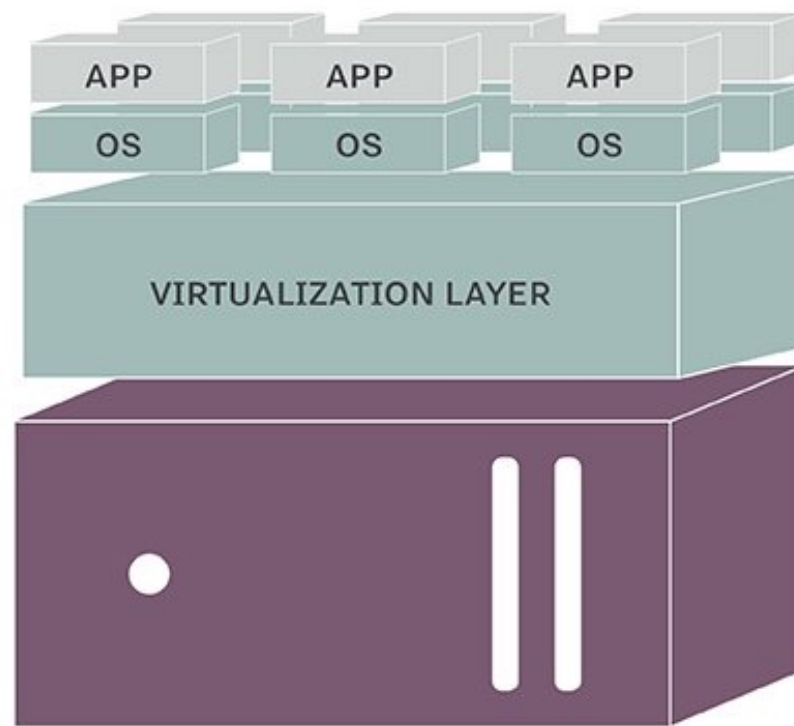
Варианты архитектур ОС

- Монолитные ОС
- Многоуровневые ОС
- ОС для виртуализации
- ОС с микроядром
- Клиент-серверные ОС

ОС для виртуализации

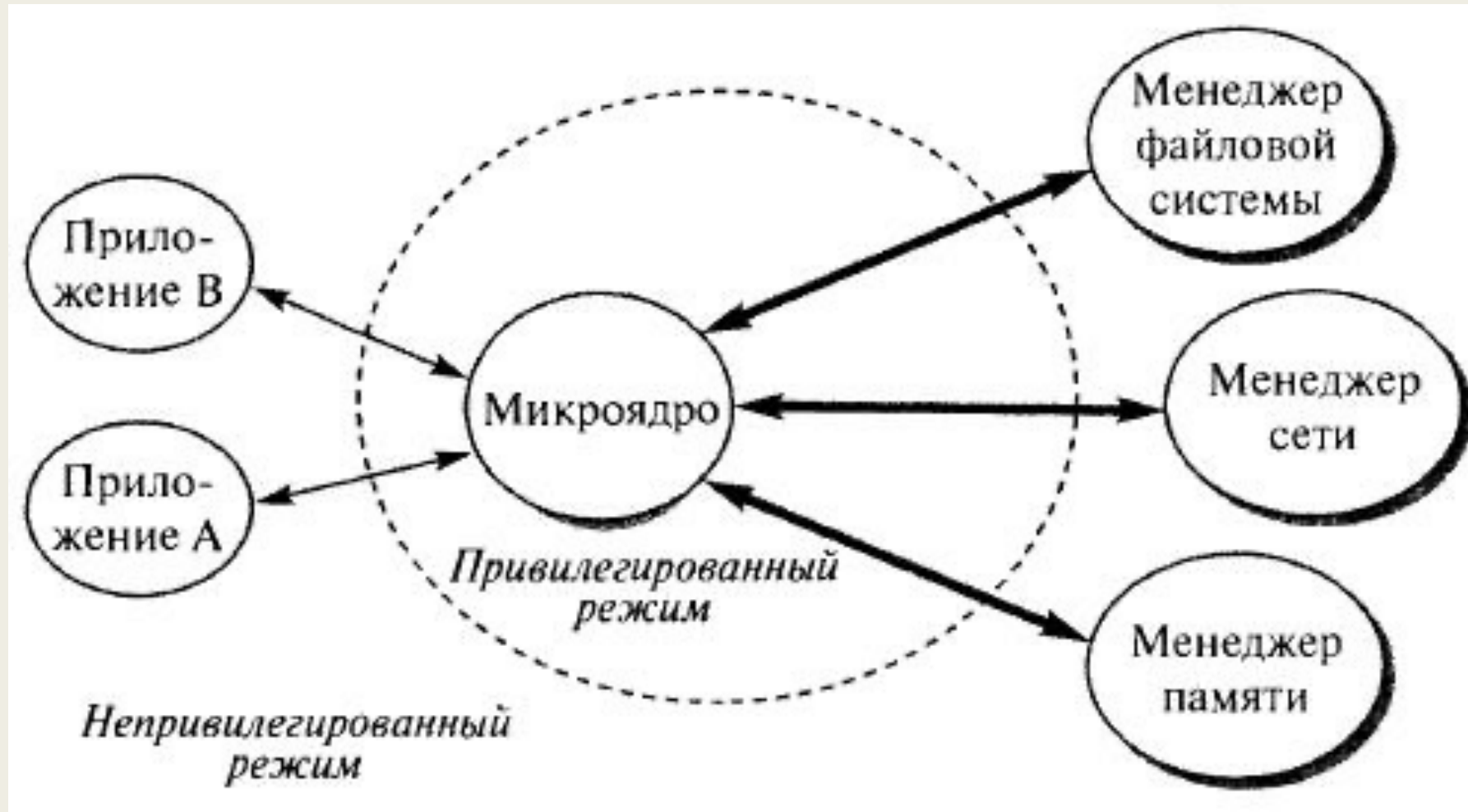


TRADITIONAL ARCHITECTURE



VIRTUAL ARCHITECTURE

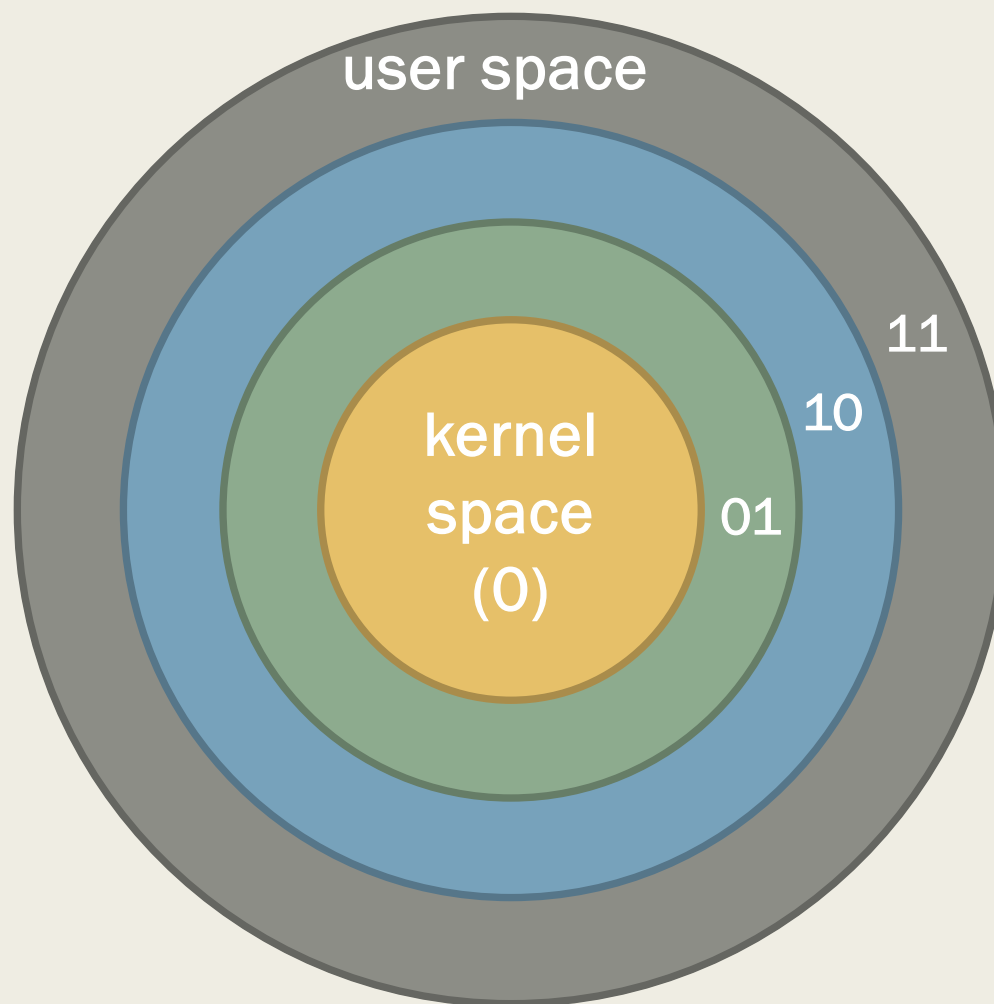
ОС с микроядром



Программный стек ОС



Режим ядра и режим пользователя



Прерывания в ОС

- Ввод-вывод
- Таймер
- Исключения
- Системные вызовы

Обращение приложений к ядру

подготовка данных в собственном адресном пространстве → вызов обработчика прерываний
→ создание/использование готового буфера в памяти ядра → копирование аргументов для исполнения процедуры ядра → вызов процедуры ядра → копирование результата в память приложения → возвращение управления приложению

Системные вызовы

Системный вызов – обращение прикладной программы к сервису операционной системы (программное прерывание с определенным номером)

- open / close
- read / write
- fork / clone
- ioctl

Системные библиотеки

Read file (syscalls)

```
mov ebx, eax
mov eax, 3
mov ecx, esp
mov edx, 0xffff
int 0x80
```

Read file (язык C)

```
char buf[SZ];
int fd;
...
read (fd, buf, SZ);
```

Системные драйверы

- **Драйвер** – это специальный компонент операционной системы, который позволяет получать доступ к физическому или виртуальному оборудованию либо реализует протокол
- **Устройства:**
 - *СИМВОЛЬНЫЕ*
 - *блочные*
 - *сокеты*

Системные драйверы

- `/dev` – специальный каталог, содержащий драйверы (файлы устройств)

Примеры:

- `/dev/hda` – жесткие диски
- `/dev/eth` – сетевые интерфейсы
- `/dev/mouse` – мышь