
Технологии параллельных и распределенных вычислений

Судаков А.А.

“Параллельные и распределенные вычисления” Лекция 3

План

- Операционные системы для параллельных и распределенных вычислений
- Программные технологии для параллельных и распределенных вычислений
- Аппаратные средства параллельных и распределенных вычислений

Операционные системы для параллельных и распределенных вычислений

- Функции операционных систем
- Сетевые и распределенные операционные системы
- Unix-подобные операционные системы
- Установка, администрирование и работа в Linux

Операционные системы

- **Операционная система** – набор программных средств для управления аппаратурой и прикладными программами, а также для обеспечения их взаимодействием друг с другом и с пользователем
- Операционная система выполняет виртуализацию аппаратных и других ресурсов для прикладных программ и пользователей
 - виртуальная память
 - виртуальный процессор[ы]
 - виртуальная файловая система
 - виртуальные каналы обмена данными

Виртуальный ресурс

- Создается иллюзия того, что программа или пользователь монополюно использует ресурс:
 - **Виртуальный процессор** – в системе один процессор и «одновременно» выполняется много программ
 - **Виртуальная память** – все программы используют одинаковые адреса памяти, но по этим адресам у каждой программы свои данные
 - **Виртуальная файловая система** – программа одинаково использует файлы независимо от того, где они физически находится
 - **Виртуальные каналы обмена** – две программы взаимодействуют друг с другом независимо от того, где они физически выполняются

Типы операционных систем

- **Многозадачные** – в операционной системе может одновременно выполняться несколько задач (возможен псевдопараллелизм)
- С поддержкой **многопроцессорности** – задачи и сама операционная система может выполняться сразу на нескольких процессорах одной машины (SMP, NUMA)
- **Сетевые** – операционные системы разных компьютеров должны взаимодействовать между собой. Никаких других требований не выдвигается
- **Распределенные** – с точки зрения прикладных программ несколько компьютеров представляются единой большой системой

Примеры

- Большинство современных операционных систем являются многозадачными, многопроцессорными, сетевым (MS Windows NT, Unix/Linux)
 - Если WWW документ находится на машине с Unix, то его можно «скачать» и прочитать под Windows
 - Если файлы находятся на Windows машине, то их можно использовать и на нескольких других Windows или Unix машин

Примеры распределенных систем

- Кластер рабочих станций – задача запускается на той машине, которая меньше всего загружена
- MOSIX - задача, запущенная на одном компьютере может перемещаться между всеми компьютерами системы
- VAX VMS – все программы работающие в операционной системе могут использовать все ресурсы всех компьютеров системы

Типы распределенных ОС

- **Single system image** – операционная система управляет всеми ресурсами всех компьютеров системы
- **Distributed system image** – операционная система каждого компьютера управляет только ресурсами своего компьютера

Примеры

- **Beowulf** кластер – несколько образов операционной системы
- **MOSIX** кластер – один образ операционной системы

Принципы построения

- Прозрачность ресурсов
 - Расположения – не зависит от расположения ресурса
 - Миграции – не зависит от перемещений ресурса
 - Копирования – не зависит от количества копий ресурса
 - Конкуренции – все конфликты по совместному использованию ресурсов решаются автоматически
 - Параллелизма – параллельное выполнение возможно без участия пользователя
- Децентрализация
 - Отсутствие одной точки сбоя
- Масштабируемость
 - Можно наращивать или сокращать количество ресурсов
- Устойчивость
 - В случае ошибки система должна восстанавливаться

Сравнение типов операционных систем

	Сетевая ОС	Распределенная ОС	ОС мультипроцессора
Компьютерная система выглядит как одна машина	НЕТ	ДА	ДА
Одна и та же ОС выполняется на всех процессорах	НЕТ	ДА	ДА
Сколько копий ОС имеется в памяти	N	N	1
Как осуществляются коммуникации	Разделяемые файлы	Сообщения	Разделяемая память
Требуется ли согласованный сетевой протокол	ДА	ДА	НЕТ
Имеется ли единая очередь выполняющихся процессов	НЕТ	НЕТ	ДА
Имеется хорошо определенная семантика разделения файлов	Обычно НЕТ	ДА	ДА

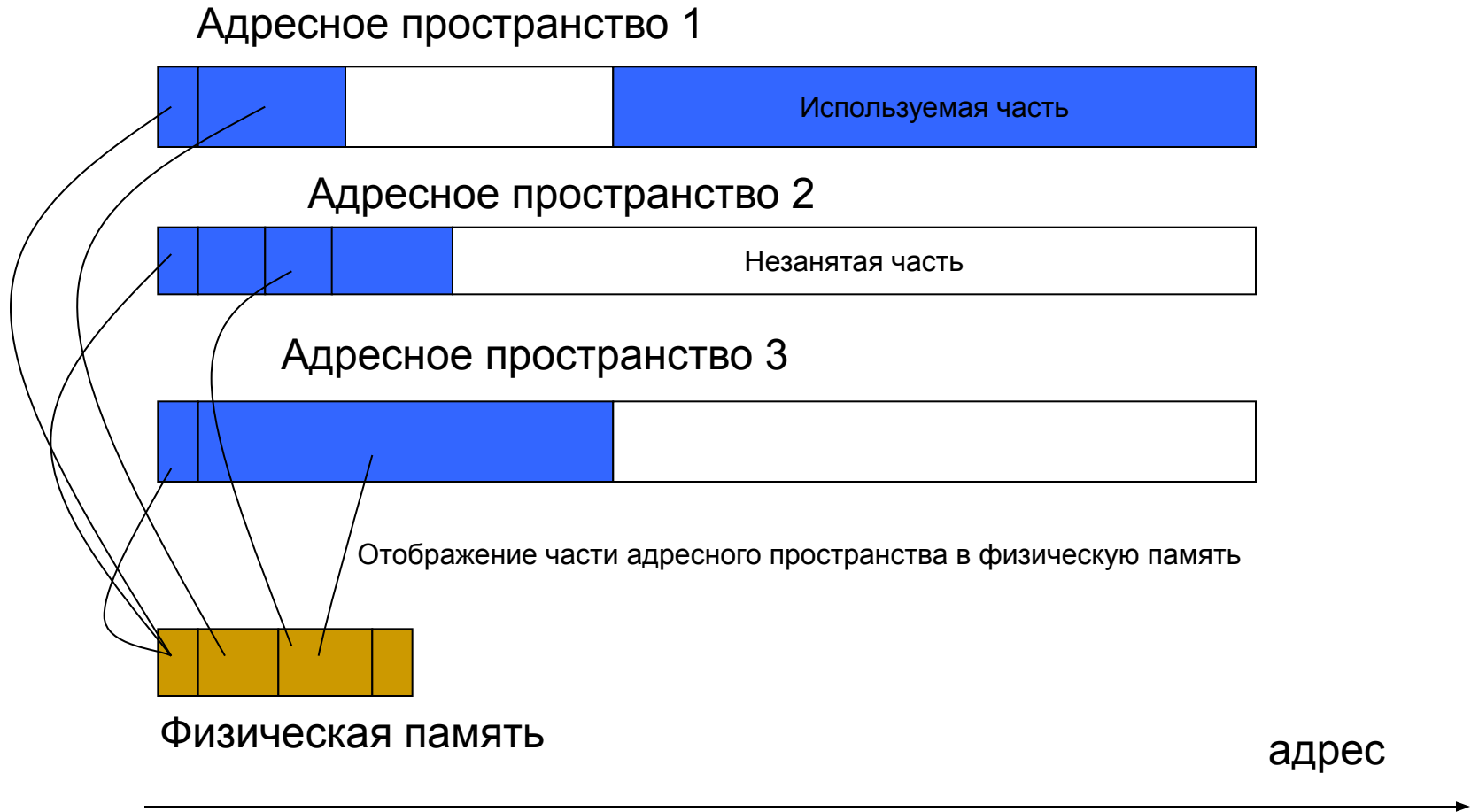
Физические и логические ресурсы

- Физические
 - процессор,
 - память,
 - хранилище данных,
 - коммуникационная подсистема
- Логические
 - процесс,
 - поток,
 - адресное пространство,
 - файл,
 - средства межпроцессного взаимодействия
 - сокет,
 - конвейер,
 - сообщение,
 - совместно используемая область памяти

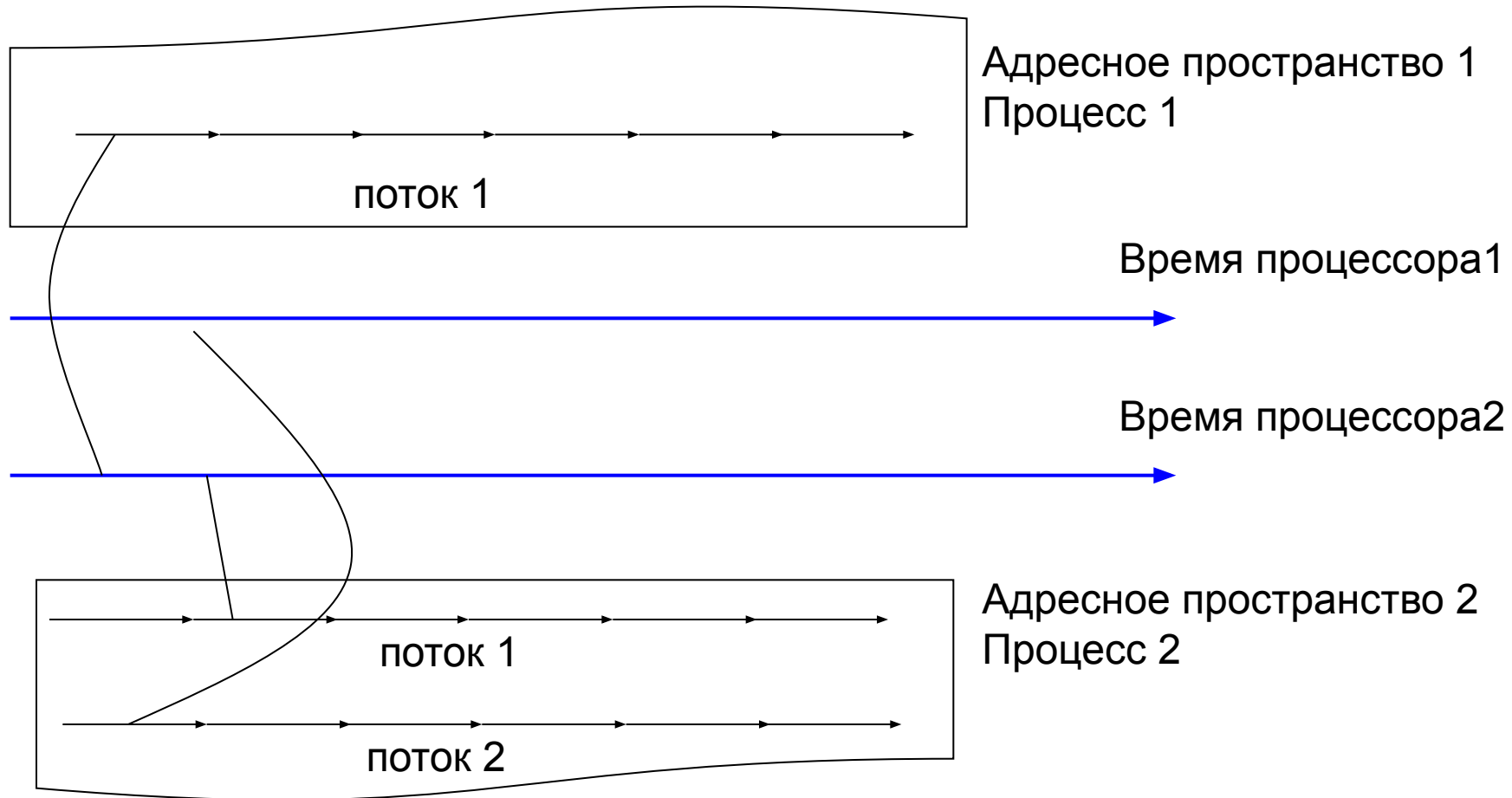
Процесс, поток, адресное пространство

- **Процесс (task, process)** – выполняющаяся программа
- **Адресное пространство** – как память видится для процесса
- **Поток (thread, lwp)** – часть процесса, которая выполняется параллельно или псевдопараллельно с другими такими же частями
- **Файл** – последовательность байтов, к каждому байту можно обратиться по его номеру

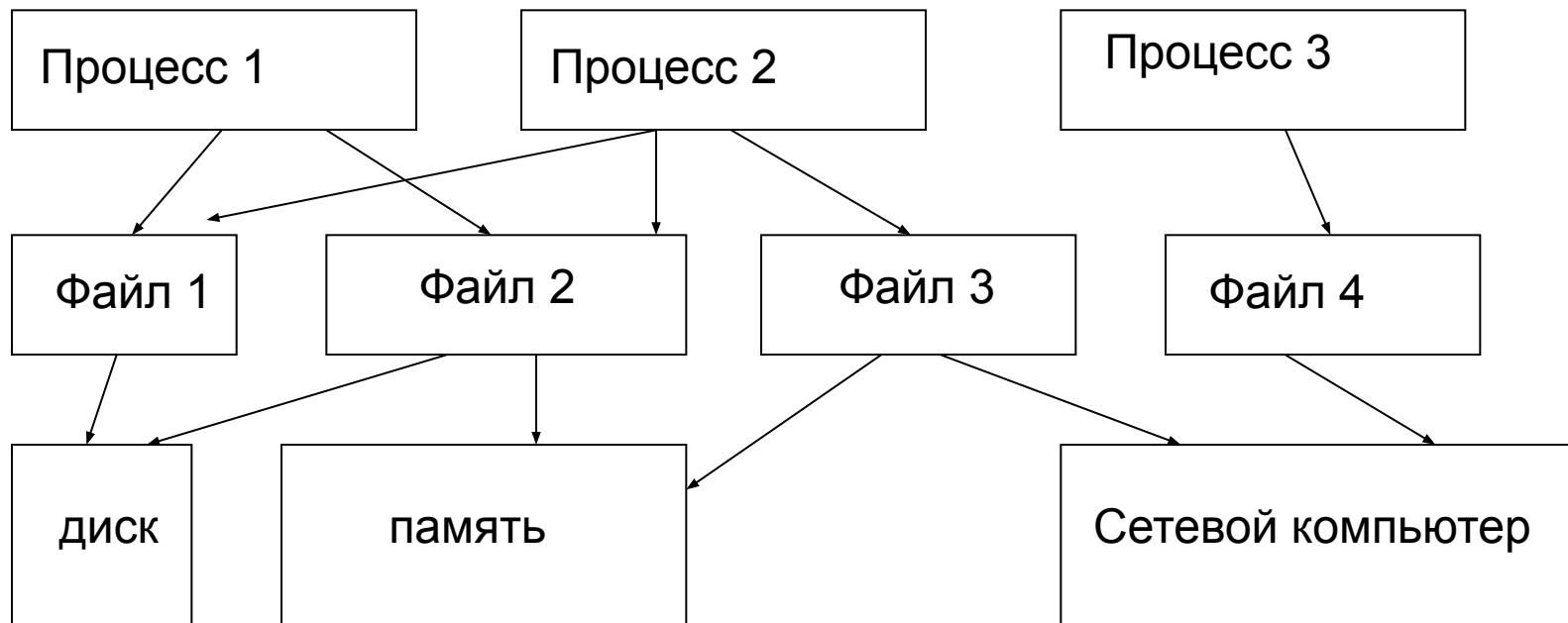
Адресные пространства (виртуализация памяти)



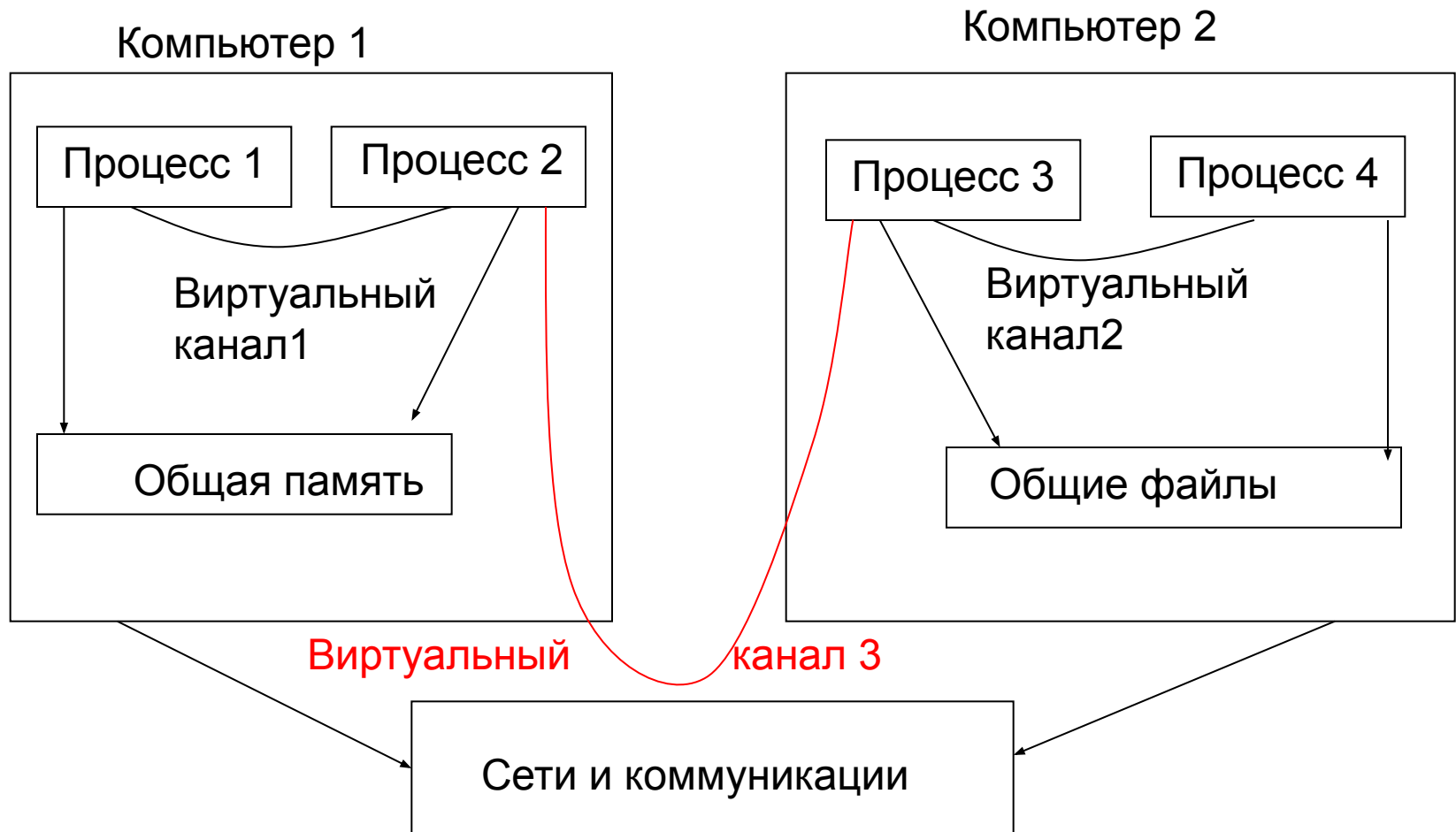
Процессы, потоки (виртуализация процессоров)



Файловые системы (виртуализация хранилищ данных)



Средства межпроцессного взаимодействия (виртуализация средств коммуникации)



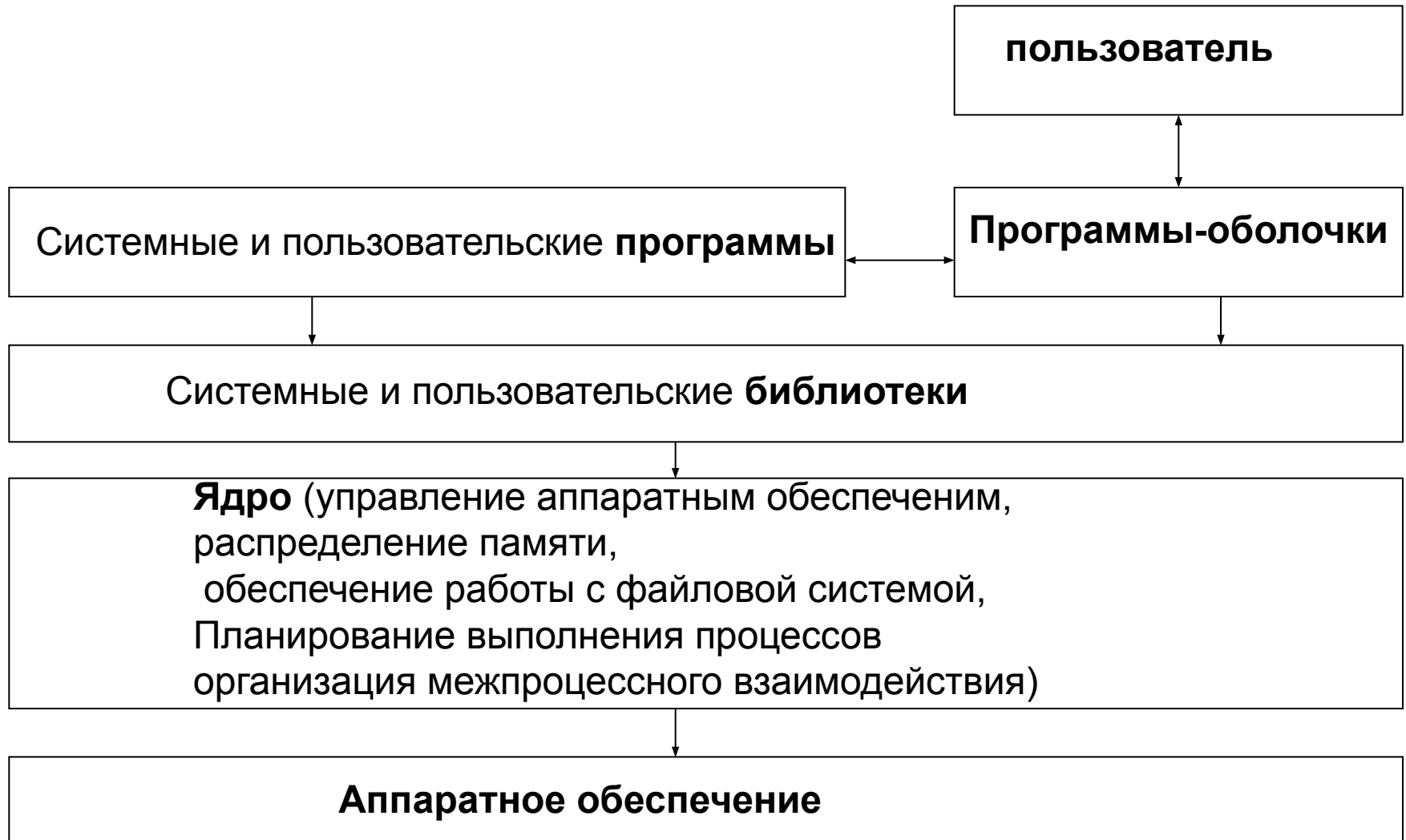
Сетевые и распределенные ОС

- **Сетевые ОС** – виртуализация файловых систем и каналов обмена для разных компьютеров
- **Распределенные ОС** – виртуализация процессоров, памяти файловых систем и каналов обмена для разных компьютеров

Unix-подобные операционные системы

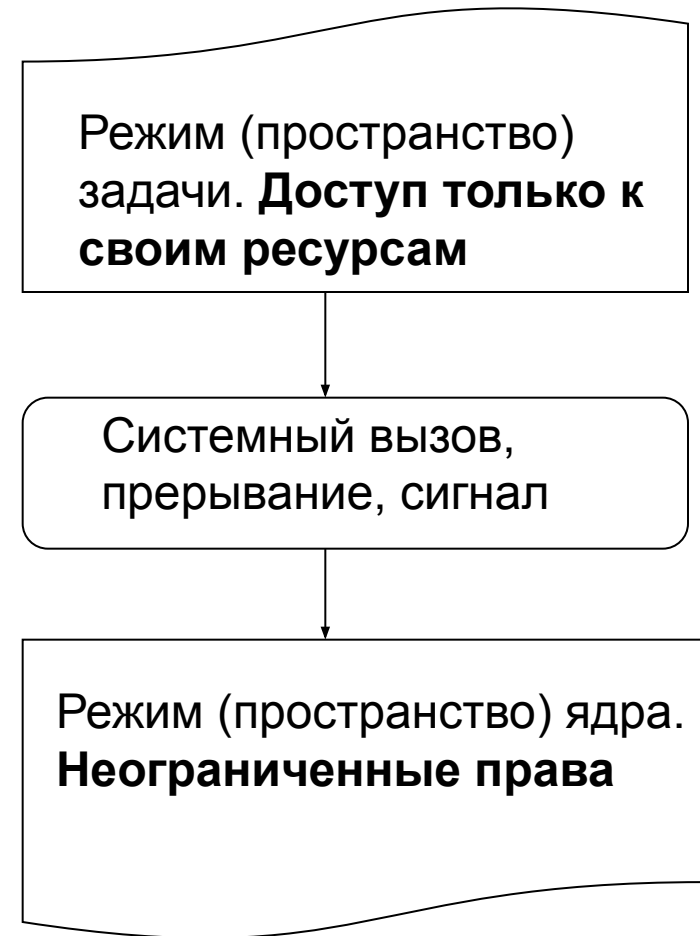
- Unix 1969 год
- Основные принципы
 - Простота перед интеллектуальностью
 - Предоставлять механизм, а не стратегию
 - Программы должны делать мало, но делать это хорошо
 - Результаты работы одной программы можно сразу же подавать на вход другой
 - Любой ресурс – это файл (по возможности)
 - Средства разработки программ и самой операционной системы должны быть включены в операционную систему

Иерархическая структура



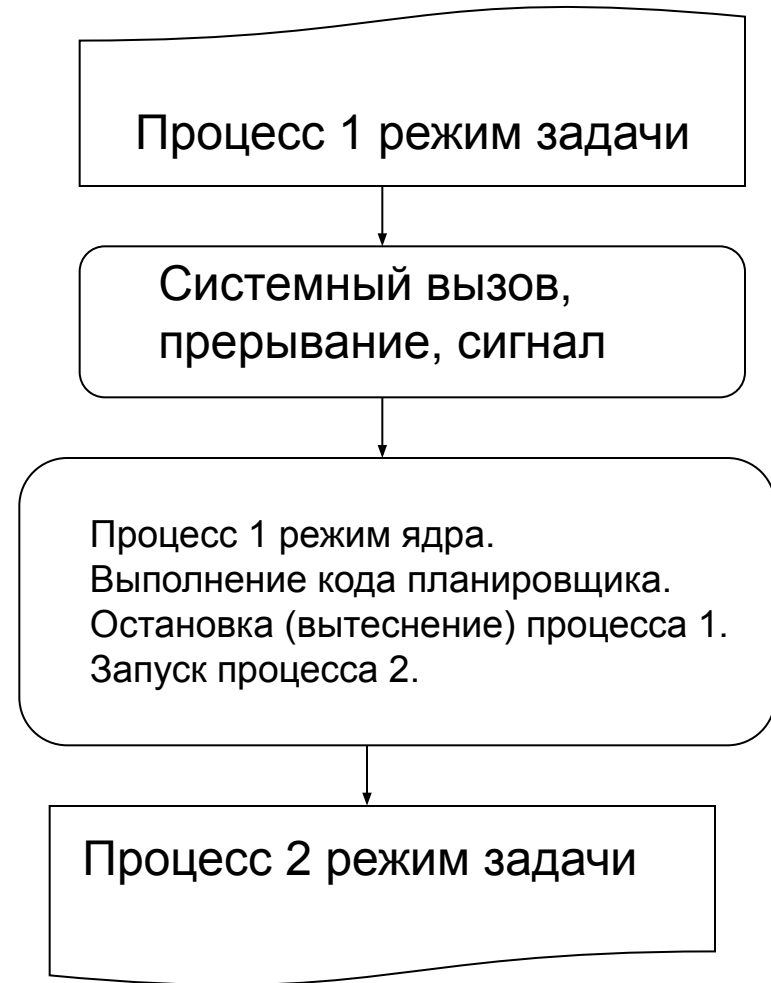
Выполнение процесса

- В режиме задачи процесс выполняется только от имени задачи и имеет доступ только к своему адресному пространству и своим ресурсам
- При получении прерывания, сигнала или при вызове системной функции процесс переключается в режим ядра
- В режиме ядра процесс имеет полный доступ ко всем ресурсам
- В режиме ядра процесс выполняет только код ядра
- Все процессы выполняются аналогично



Планирование процессов

- Каждый процесс выполняется в течение интервала (кванта) времени
- Либо после истечения кванта, либо по «желанию» процесса управление передается другому процессу
- Планирование нового процесса в режиме ядра
- Вытеснение (preemption) первого процесса
- Запуск второго процесса
- Процедура повторяется



Переключение контекста

- Каждый процесс использует свои ресурсы
- При переходе из одного режима в другой, остановке одного процесса и запуске другого необходимо, чтобы регистры процессора, виртуальная память и др. соответствовали новому процессу и режиму (**работа в контексте нового процесса**)
- Для этого происходит перегрузка регистров процессора и другие операции – **переключение контекста**
- Переключение контекста – достаточно «тяжеловесная» операция

Процессы и потоки

- Для выполнения процессов необходимо полностью переключать контекст
- Для выполнения потоков контекст переключать нужно не полностью
- Поток – light weight process

- Использовать процессы – проще, но производительность ниже
- Использовать потоки сложнее, но производительность выше

Обмен данными

- Для процессов при обмене данными необходимо переключение контекста
- Для потоков – все данные общие и переключения контекста нет

ОС Linux

- Linux 1991 г
- Ядро <http://www.kernel.org>
- Множество различных вариантов – поставок, **совместимых с ядром по системным вызовам**
- Для каждой поставки свои особенности инсталляции, администрирования, свои программы, версии, библиотеки

Ядро Linux

- ~300 Мбайт кода (архив ~30 МБайт)
- Версия 2.6.11
- Первая цифра – старший номер
- Вторая цифра – серия
- Третья цифра – номер выпуск
- Четные серии – стабильные ядра
- Нечетные серии – разрабатываемые ядра

Поставки Linux

- Наиболее характерные
 - RedHat и производные
 - Debian
 - Slackware
 - Gentoo
- Все имеют свои особенности ядра, но все будут работать со стандартным ядром соответствующей серии, которое необходимо компилировать

Інсталляцыя

- **Методы**
 - CD,
 - сеть (http, ftp, рхе),
 - локальный диск
- **Процедура**
 - Загрузка инсталлятора
 - Указание местоположение инсталляции
 - Определение оборудования
 - Разбивка жесткого диска
 - Выбор пакетов для инсталляции
 - Копирование инсталляции
 - Начальная конфигурация
 - Установка загрузчика