Компьютерные системы

Чернов Александр Владимирович
Доцент, к.ф.-м.н.

blackav@gmail.com

Лекционный курс

- 1.5 пары в неделю (два модуля)
- +3 лекции по архитектуре компьютеров, прочитанные в июне
- Темы:
  - Общие принципы построения операционных систем
  - POSIX API
  - Архитектура современной ОС на примере Linux

Семинарские занятия

- 1.5 пары в неделю (два модуля)
- Темы:
  - Знакомство с Linux
  - Представление данных в компьютере
  - Низкоуровневое программирование
  - Программирование с помощью POSIX API в Linux

Форма отчетности

- Оценка за курс: 40% экзамен + 60% накопленная оценка
- Экзамен письменный, теоретический
- 3 контрольных работы по семинарским занятиям в течение семестра
- Работа на семинарах
- Домашняя работа

Проверка задач

- Автоматическая проверка на тестах
- Code Review

- Контрольные работы, семинарские занятия — жесткое ограничение по времени сдачи
- Домашние работы: срок выполнения — 1-2 недели, затем максимальный балл резко падает до 0 в течение недели
- Штрафы за непрохождение тестов
- Бонусы первым сдавшим

Проектная работа

- 1 пара в неделю
- Проекты будут вести преподаватели семинарских занятий
- Анонс и распределение по проектам — ориентировочно 3 неделя сентября

Особенности расписания

- Часть семинаров, отведенных под проектную деятельность, будет занята регулярными семинарами по курсу компьютерных систем
- 9 сентября — неучебный день
  - Лекция переносится на 12 сентября
  - Семинар, выпадающий на 9 сентября, будет проведен за счет проектного семинара
  - Семинар, пропавший 12 сентября из-за переноса лекции, будет проведен за счет проектного семинара
  - В группах, которые не пострадали из этих переносов, проектный семинар на 2 и 3 неделе не проводится

Ресурсы

- Вики: http://wiki.cs.hse.ru/Компьютерные_системы
- GitHub: https://github.com/hseos/hseos-course
- Ejudge: hse.ejudge.ru

Литература

- Э. Таненбаум, Х. Бос. **Современные операционные системы.** 4-е издание. СПб.: Питер, 2015. ISBN 978-5-496-01395-6
- Р. Лав. **Ядро Linux: описание процесса разработки**. 3-е издание. М.: ООО И.Д. Вильямс, 2015. ISBN 978-5-8459-1944-1
- Р. Э. Брайант, Д. Р. О'Халларон. **Компьютерные системы: архитектура и программирование.** СПб.: БХВ-Петербург, 2005. ISBN 5-94157-433-9
- С. А. Раго, У. Р. Стивенс. UNIX. **Профессиональное программирование**. 3-е издание. М.: Символ-Плюс, 2014. ISBN 978-5-93286-216-2

Дополнительно

- Э. Таненбаум, Т. Остин. Архитектура компьютера. 6-е издание. СПб.: Питер, 2015. ISBN 978-5-496-00337-7
- Д. Паттерсон, Дж. Хеннесси. Архитектура компьютера и проектирование компьютерных систем. 4-е издание. СПб.: Питер, 2015. ISBN 978-5-459-00291-1

- Simon, Garfunkel et al. The UNIX haters handbook. http://web.mit.edu/simsong/www/ugh.pdf
- Cooper, Hofstadter. IOS for nerds: trendy, stylishly, youthly

Компьютерная система

- Компьютерная система — совокупность аппаратных и программных средств, функционирующих в единой системе и предназначенных для решения задач определенного класса

  - Аппаратное обеспечение (hardware)
  - Программное обеспечение (software)
    - Системное — обеспечение работы самой компьютерной системы и подготовка прикладного ПО
      - **ОПЕРАЦИОННАЯ СИСТЕМА**
    - Прикладное — решение задачи, для которой предназначена компьютерная система

Компьютерная система — иерархия «исполнителей»

- Исполнитель - «устройство»
  - Фиксированный набор команд
  - Язык (нотация) для записи команд
- Исполнитель — виртуальная машина
- Компьютерная система может рассматриваться с разных сторон, на разных уровнях абстракции, получая разные исполнители (виртуальные машины)

Иерархия исполнителей

- Виртуальная машина командного процессора Unix (shell):

  cp -rpd ~friend/project1 .

  - Язык командного процессора
- Виртуальная машина C++:

```
#include <boost/filesystem.hpp>
using boost::filesystem;
for (directory_entry &e : directory_iterator(path(s))) {
}
```

Иерархия исполнителей

- Виртуальная машина POSIX (системные вызовы)

  int fd = open(path, O_RDONLY | O_DIRECTORY, 0);
  getdents
- Виртуальная файловая система (VFS) — компонент ядра

  vfs_open(...)

Иерархия исполнителей

- Интерфейс блок-ориентированных устройств (компонента ядра) — чтение блока данных, запись блока данных, кеширование блоков
- Физическое устройство (SATA интерфейс, USB, …) - система команд контроллера устройства

- Чем ниже по иерархии исполнителей, тем более узкий класс устройств, специфические детали моделей устройств и т. п.

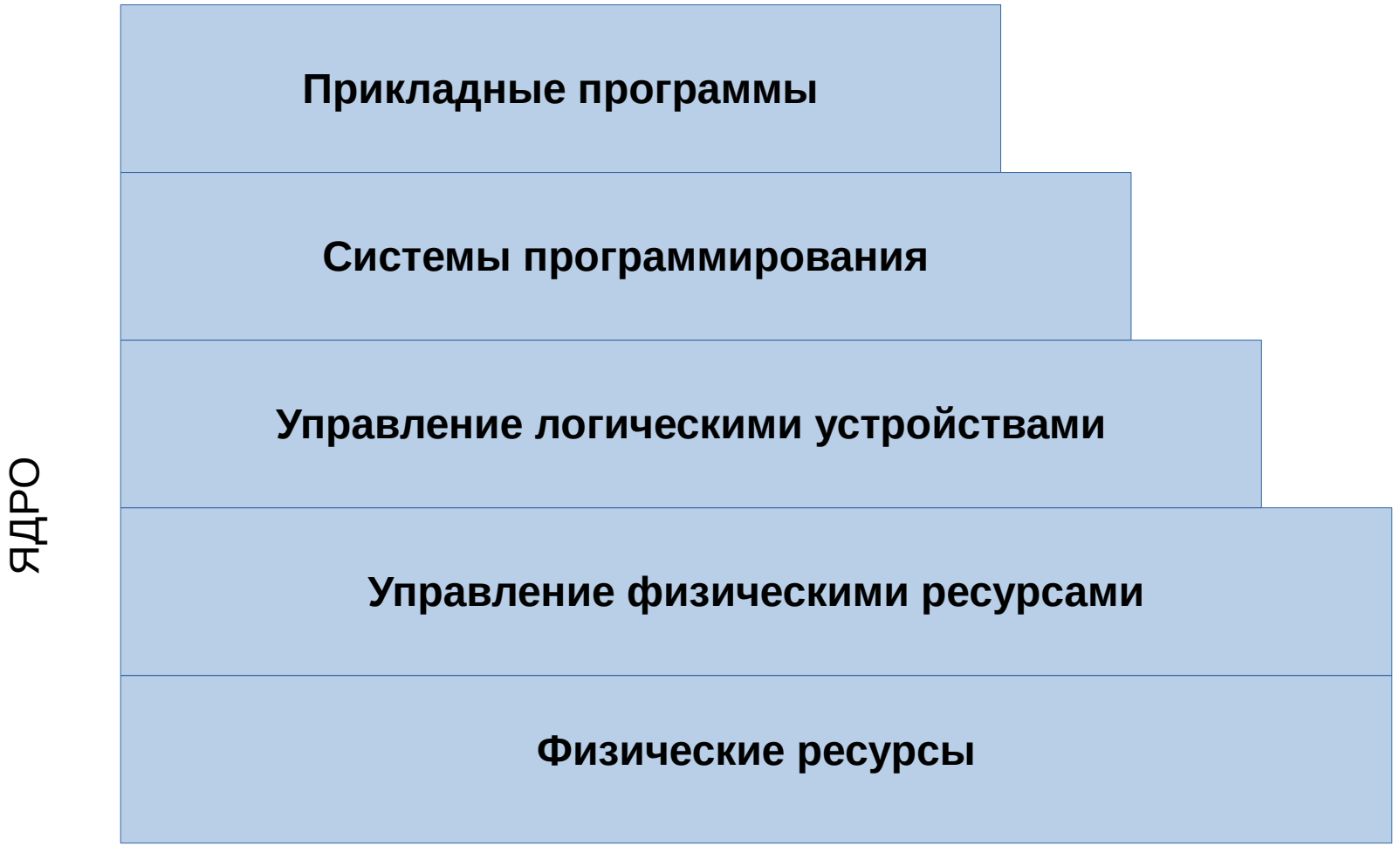
Задача операционной системы

- Задача ОС: предоставить унифицированный интерфейс для работы с широкими классами устройств (спрятать «странности» и «ошибки» работы разных специфических устройств за внешним «красивым» интерфейсом)
- Пример: POSIX API включает в себя стандартный интерфейс для работы с файловыми системами
- Более универсальный (напр. переносимый между POSIX и Windows) интерфейс может предоставляться библиотеками (boost, Qt...)

Ядро ОС

- Исполнители ниже определенного уровня размещаются в ядре операционной системы
- Ядро:
  - Работает в привилегированном режиме процессора
  - Резидентно в памяти
- Функциональность ядра — результат компромиссов:
  - Компактность
  - Устойчивость к ошибкам
  - Наличие необходимого функционала по обеспечению корректной работы компьютерной системы

Структура компьютерной системы



Ресурсы КС

- Ресурсы КС — совокупность аппаратных и программных ресурсов. Например, память, время ЦП, место на диске и т. п.
- Ресурсы КС ограничены, возникает ситуация нехватки ресурсов и конкуренции за ресурсы
- Ресурсы КС могут иметь разные права доступа для разных пользователей
- **Управление ресурсами — вторая задача операционной системы**

Управление ресурсами

- Потребитель ресурса не должен иметь возможности нарушать функционирование других потребителей, если они этого не желают.
- Потребитель ресурса не должен иметь возможности полностью блокировать использование этого ресурса.
- Потребитель ресурса не должен иметь возможность управлять потреблением ресурса другими

Операционная система

- Предоставляет унифицированный интерфейс работы с ресурсами КС и скрывает аппаратные особенности функционирования ресурсов
- Управляет ресурсами компьютерной системы:
  - Мультиплексирование по времени (определение порядка, в котором ресурс предоставляется)
  - Мультиплексирование по пространству (определение части ресурса, который предоставляется)
- Время ЦП и ОЗУ — это ресурсы, которыми тоже нужно управлять!

Процессы

- Программа — это текст на некотором языке, понятный для соответствующей машины. Например, программа в машинных кодах, программа на Си и т. д.
- Процесс — это программа в состоянии выполнения в компьютерной системе
- Процессы можно рассматривать как потребителей ресурсов ВС

История развития ОС

- Первое поколение ЭВМ — ламповые. ОС отсутствует (мало ресурсов)
- Второе поколение — транзисторные. Системы пакетной обработки.
- Третье поколение — микросхемы (ранние).
  - IBM 360 — многозадачность, разделение времени, многопользовательские
  - PDP-11 — Unix

История развития ОС

- Четвертое поколение — микропроцессоры, персональные компьютеры — GUI, сетевые и распределенные ОС
- Пятое поколение — мобильные компьютеры — низкое энергопотребление, беспроводная связь, облачные технологии