1. **Операционные системы и системное программирование**.
2. **5 семестр**: 34 часа лекций, 34 часа лабораторных занятий, **зачет**.
3. **6 семестр**: 36 часов лекций, 54 часа лабораторных занятий, **экзамен**.
4. **Инструменты:** OS Windows 10, OS Linux (CentOS, Ubuntu), C/C++.
5. **Связь с другими дисциплинами:** «Языки программирования», **«**ЭВМ и периферийное оборудование», «Программирование и безопасность сетевых приложений».
6. **Литература:**

- А. Таненбаум «Современные операционные системы»;

- А. Побегайло «Системное программирование в Windows»;

- Н.Мэтью «Основы программирования в Linux»;

- Р.Лав «Системное программирование»;

- эл. конспект лекций (diskstation).

1. **Лектор:** Смелов Владимир Владиславович, к.т.н., доцент.

**ОС, ПОИТ-3, Лекция 01**

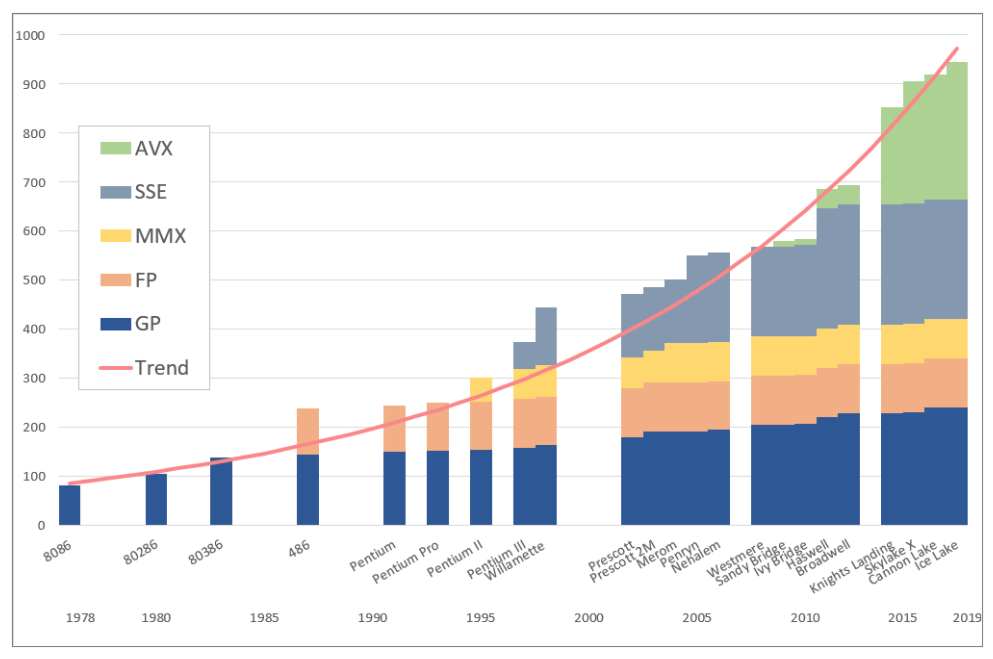
1. **OS**: **Operating System** – операционная система.
2. **OS**: комплекс программ.
3. **OS**: **интерфейс**, пользовательский интерфейс, интерфейс с аппаратурой (драйверы), программный интерфейс (фреймворк), утилиты, системные программы, ядро.



1. **OS**: **Hardware:** оборудование(аппаратное обеспечение)



* архитектура процессора(CISC, **RISC**, MISC, VLIW);
* набор команд процессора(x86, x64, **x86-64,** ARM);



* постоянная память (ПЗУ);
* тактовый генератор;
* форматы (представление)данных (LE, BE), IEEE-754;
* разрядность данных(8, 16, 32,64,128);
* набор регистров и их назначение;
* память: регистры, кэш, оперативная память, внешняя память (диски, ленты);
* система адресации;
* система прерываний;
* часы, таймер.

1. **OS**:**BIOS:** Basic Input/Output System – набор микропрограмм (в энергонезависимой памяти), низкоуровневые примитивы:

* тестирование оборудования (при включении, POST -Power-On Self-Test);
* поиск и подключение устройств типа plug and play;
* первоначальная загрузка операционной системы;
* API для работы с оборудованием (на этапе загрузки ОС, потом используются драйверы);
* активация в offline (SLIC — software licensing description table);
* пользовательский интерфейс для настройки оборудования (устройство для загрузки, частота, лимиты, перечень устройств и т.п.).

1. **OS**: **Drivers**: драйверы – программы, предназначенные для унифицированного доступа программного обеспечения к аппаратуре:

* HAL – Hardware Abstraction Layer – слой аппаратных абстракций – по сути драйвер материнской платы; реализуется в программном обеспечении, расположенном между аппаратурой и ядром OS (аппаратно-зависимая компонента); по разному реализован а разных OS;
* драйверы разрабатывают поставщики оборудования;
* драйверы для большинства устройств (материнских плат) – в составе дистрибутива OS;
* для периферии (принтеры, сетевых карты, мониторы) драйверы, как правило, поставляются вместе с оборудованием;
* в Linux драйвер является частью ядра, в Windows – это dll-файлы (расширение sys);
* в операционной системе есть процедура установки драйвера (в разных OS по-разному);
* DDK – Driver Development Kit – инструментарий для разработки драйверов (WDK – Windows Driver Kit);
* виртуальные драйверы;
* NDIS – Network Driver Interface Specification (Microsoft, 3Com), в основном применяется Microsoft.

1. **OS**: **OS Core (Kernel)**: ядро операционной системы:

* резидентная часть программного обеспечения OS;
* всем ресурсам соответствует объекты OS;
* обработка аппаратных и программных прерываний;
* координированный доступ к ресурсам OS: процессорное время, оперативная память, устройства ввода вывода;
* единица работы – процесс (в простейшем случае приложение);
* создание и уничтожение процессов;
* межпроцессное взаимодействие;
* управление памятью;
* синхронизация процессов;
* диспетчеризация доступа к ресурсам OS;
* поддержка абстракции данных – файловая система;
* разная структура в разных OS: монолитная, модульная, микроядро, экзоядро, микроядро, …;
* в Windows ядро поддерживает графику, в Unix/Linux графика отдельный процесс;
* безопасность: аутентификация, авторизация.



1. **OS**: **Programs Interface:** интерфейс между приложениями и ядром OS:

* OS API;
* OS Framework, .NET Framework, .NET CORE Framework;
* POSIX, IEEE Std 1003; IBM AIX, HP-UX, UnixWare, QNX Neutrino, iOS, FreeBSD;
* Windows API.



1. **OS**: **Programs**:

* компоновщик, загрузчик, компилятор С/C++, assembler;
* утилиты графические, утилиты командной строки;
* пользовательские программы.



1. **OS**: **Users Interface** (UI): Graphic UI, CLI (Command Line Interface).



* GUI: Windows (Explorer, Desktop, Try), KDE(Linux), GNOM(Linux),MATE (Linux), Xfce (Ubuntu/Linux),
* CLI: Bourne Shell (sh), CMD (Windows), Power Shell (Windows), bash, ksh;
* <https://ru.wikipedia.org/Сравнение_командных_оболочек>

1. **OS**:



1. **OS**: в курсе «**Операционные системы**» изучаются объекты и принципы работы ядра операционных систем:

* процессы;
* потоки;
* диспетчеризация потоков;
* отсчет времени (часы, таймеры);
* управление памятью;
* файловая система;
* ввод/вывод данных;
* обмен данными между процессами;
* синхронизация потоков и процессов.

1. **OS**: в курсе **«Системное программирование»** изучается OS API и применение OS API при разработке приложений.
2. **OS**: история: развитие аппаратного обеспечения и программного обеспечения, один пользователь, много пользователей, сетевые, мобильные, облачные, ….
3. **OS**: разновидности:

* **для мейнфреймов** (много одновременных заданий, пакетная обработка данных), IBM OS/390, IBM Z/OS;
* **серверные:** линейка Windows Server 2008/2012/2016, FreeBSD, Solaris, Linux Server;
* **сетевые**: Novel NetWare
* **персональные**: Windows, Mac OS, Ubuntu;
* **мобильные**: iOS, Android;
* **встроенные**: Embedded Linux, QNX;
* **реального времени**: мягкое и жесткое реальное время, FreeRTOS ([https://www.freertos.org](https://www.freertos.org/));
* **смарт-карт:** MULTOS

1. **OS**: накоплен большой опыт разработки OS,поэтому можно говорить об **общих принципах устройства OS**.