**Практическая работа №3  
Изучение структуры операционной системы**

**Цель занятия:** изучить структуру операционной системы, назначение ее отдельных  
компонентов, функции операционных систем, их классификацию и критерии оценки.

**Задание:** составить конспект основных теоретических положений.

**Операционная система (operating system)** – комплекс программ, предоставляющий  
пользователю удобную среду для работы с компьютерным оборудованием

Операционная система позволяет запускать пользовательские программы; управляет всеми ресурсами компьютерной системы – процессором (процессорами), оперативной памятью, устройствами ввода вывода; обеспечивает долговременное хранение данных в виде файлов на устройствах внешней памяти; предоставляет доступ к компьютерным сетям

Все компоненты можно разделить на два больших класса – программы или программное  
обеспечение (ПО, **software**) и оборудование или аппаратное обеспечение (**hardware**).

**Функции операционной системы**  
К основным функциям, выполняемым операционными системами, можно отнести:  
• обеспечение выполнения программ – загрузка программ в память, предоставление  
программам процессорного времени, обработка системных вызовов;  
• управление оперативной памятью – эффективное выделение памяти программам,  
учет свободной и занятой памяти;  
• управление внешней памятью – поддержка различных файловых систем;  
• управление вводом-выводом – обеспечение работы с различными периферийными устройствами;  
• предоставление пользовательского интерфейса;  
• обеспечение безопасности – защита информации и других ресурсов системы от  
несанкционированного использования;  
• организация сетевого взаимодействия.

**Ядро** - это основная, «самая системная» часть операционной системы. Имеются разные определения ядра. Согласно одному из них, ядро — это **резидентная** часть системы Остальные модули ОС являются **транзитными**, т.е. подгружаются в память с диска

**К транзитным частям системы относятся**:  
• **утилиты** (utilities) — отдельные системные программы, решающие частные задачи, такие как форматирование и проверку диска, поиск данных в файлах, мониторинг  
(отслеживание) работы системы и многое другое;  
• **системные библиотеки подпрограмм**, позволяющие прикладным программам  
использовать различные специальные возможности, поддерживаемые системой (например, библиотеки для графического вывода, для работы с мультимедиа и т.п.);  
• **интерпретатор команд** — программа, выполняющая ввод команд пользователя,  
их анализ и вызов других модулей для выполнения команд;  
• **системный загрузчик** — программа, которая при запуске ОС (например, при  
включении питания) обеспечивает загрузку системы с диска, ее инициализацию и старт;  
• другие виды программ, в зависимости от конкретной системы.  
Особую роль в структуре системы играют **драйверы устройств**.

В качестве программного интерфейса системы, т.е. средств для обращения прикладных  
программ к услугам ОС, используется документированный набор **системных вызовов** или  
**функций API** (Applied Programming Interface).

API-функций обращаются к одному и тому же недокументированному системному вызову, но имеют различные обрамляющие части, работающие в режиме пользователя.  
Там, где различие между двумя этими понятиями несущественно, можно использовать  
нейтральный термин «**системные функции**».

**Существует два основных вида ядер** – монолитные ядра (**monolithic kernel**) и микроядра (**microkernel**).

В монолитном ядре реализуются все основные функции операционной системы, и оно является, по сути, единой программой, представляющей собой совокупность процедур. В микроядре остается лишь минимум функций, который должен быть реализован в привилегированном режиме: планирование потоков, обработка прерываний, межпроцессное взаимодействие. Остальные функции операционной системы по управлению приложениями, памятью, безопасностью и пр. реализуются в виде отдельных модулей в пользовательском режиме. Ядра, которые занимают промежуточные положение между монолитными и микроядрами, называют гибридными (**hybrid kernel**).

**Примеры различных типов ядер:**  
• монолитное ядро – MS-DOS, Linux, FreeBSD;  
• микроядро – Mach, Symbian, MINIX 3;  
• гибридное ядро – NetWare, BeOS, Syllable.  
Кроме ядра в привилегированном режиме (в большинстве операционных систем) работают драйверы (driver) – программные модули, управляющие устройствами.  
В состав операционной системы также входят:  
• системные библиотеки (system DLL – Dynamic Link Library, динамически подключаемая библиотека), преобразующие системные вызовы приложений в системные вызовы ядра;  
• пользовательские оболочки (shell), предоставляющие пользователю интерфейс –  
удобный способ работы с операционной системой.  
Пользовательские оболочки реализуют один из двух основных видов пользовательского  
интерфейса:  
• текстовый интерфейс (Text User Interface, TUI), другие названия – консольный интерфейс (Console User Interface, CUI), интерфейс командной строки (Command Line Interface,  
CLI);  
• графический интерфейс (Graphic User Interface, GUI).

**Классификацию операционных систем можно осуществлять несколькими способами:**  
1. По способу организации вычислений:  
- системы пакетной обработки (batch processing operating systems) – целью является выполнение  
максимального количества вычислительных задач за единицу времени; при этом из нескольких  
задач формируется пакет, который обрабатывается системой;  
- системы разделения времени (time-sharing operating systems) – целью является возможность  
одновременного использования одного компьютера несколькими пользователями; реализуется  
посредством поочередного предоставления каждому пользователю интервала процессорного времени;  
- системы реального времени (real-time operating systems) – целью является выполнение каждой  
задачи за строго определённый для данной задачи интервал времени.  
2. По типу ядра:  
- системы с монолитным ядром (monolithic operating systems);  
- системы с микроядром (microkernel operating systems);  
- системы с гибридным ядром (hybrid operating systems).  
3. По количеству одновременно решаемых задач:  
- однозадачные (single-tasking operating systems);  
- многозадачные (multitasking operating systems).  
4. По количеству одновременно работающих пользователей:  
- однопользовательские (single-user operating systems);  
- многопользовательские (multi-user operating systems).  
5. По количеству поддерживаемых процессоров:  
- однопроцессорные (uniprocessor operating systems);  
- многопроцессорные (multiprocessor operating systems).  
6. По поддержке сети:  
- локальные (local operating systems) – автономные системы, не предназначенные для работы в компьютерной сети;  
- сетевые (network operating systems) – системы, имеющие компоненты, позволяющие работать с компьютерными сетями.  
7. По роли в сетевом взаимодействии:  
- серверные (server operating systems) – операционные системы, предоставляющие доступ к ресурсам сети и управляющие сетевой инфраструктурой;  
- клиентские (client operating systems) – операционные системы, которые могут получать доступ к ресурсам сети.  
8. По типу лицензии:  
- открытые (open-source operating systems) – операционные системы с открытым исходным  
кодом, доступным для изучения и изменения;  
- проприетарные (proprietary operating systems) – операционные системы, которые имеют  
конкретного правообладателя; обычно поставляются с закрытым исходным кодом.  
9. По области применения:  
- операционные системы мэйнфреймов – больших компьютеров (mainframe operating systems);  
- операционные системы серверов (server operating systems);  
- операционные системы персональных компьютеров (personal computer operating systems);  
- операционные системы мобильных устройств (mobile operating systems);  
- встроенные операционные системы (embedded operating systems);  
- операционные системы маршрутизаторов (router operating systems).