

Общий план-график курса «Операционные системы»

Месяц	Разделы	Лабораторные работы и курсовой проект
II	<p>Раздел 1. ОСНОВНЫЕ ПОНЯТИЯ. СИСТЕМА ПРЕРЫВАНИЙ</p> <p>Раздел 2. УПРАВЛЕНИЕ ПРОЦЕССАМИ. Общие понятия и принципы.</p> <p>Раздел 3. УПРАВЛЕНИЕ ПАМЯТЬЮ. Методы управления</p>	<p>Лаб.1. Виртуализация ОС</p> <p>Лаб.2 Система прерываний. Разработка обработчика прерывания</p>
III- IV	<p>Раздел 4. ОРГАНИЗАЦИЯ ВНЕШНЕЙ ПАМЯТИ. Логическая организация жесткого диска. Системные структуры ОС</p> <p>Раздел 5. ОРГАНИЗАЦИЯ ФАЙЛОВЫХ СИСТЕМ. Основные понятия. Файловая система FAT. Файловая система NTFS.</p>	<p>Лаб.3. Конфигурирование жесткого диска</p> <p>Лаб.4 Анализ системных структур жесткого диска</p> <p>Лаб.5 Организация логических дисков</p> <p>Лаб.6 Механизмы доступ к объектам в файловой системе FAT</p>
IV-V	<p>Раздел 6. ПЛАТФОРМА процессоров x86 ДЛЯ ОС</p> <p>Архитектура реального режима процессоров. Особенности ОС на базе реального режима</p> <p>Защищенный режим процессоров для ОС. Управление памятью. Механизмы защиты для ОС. Реализация многозадачности и прерываний</p>	<p>КП. Разработка системных низкоуровневых программ. Темы: Операции с разделами/ логическими дисками, доступ к объектам файловой системы</p> <p><i>Задание на КП выдается после выполнения и защиты лабораторных работ. Закончить ЛБ надо до середины апреля, тогда КП будет выполнен в срок</i></p>
VI	Экзамен	<i>К экзамену допускаются лица, выполнившие учебную программу</i>

Раздел 1. Общие понятия

1-1. Понятие операционной системы

- «Операционная система» - это комплекс программ, управляющих работой вычислительной системы (компьютера)
 - Операционная система базируется на особенностях архитектуры процессора – «аппаратной платформе», то есть является аппаратно-зависимой программой.
 - Развитие операционных систем - всегда следствие развития архитектуры процессоров
-

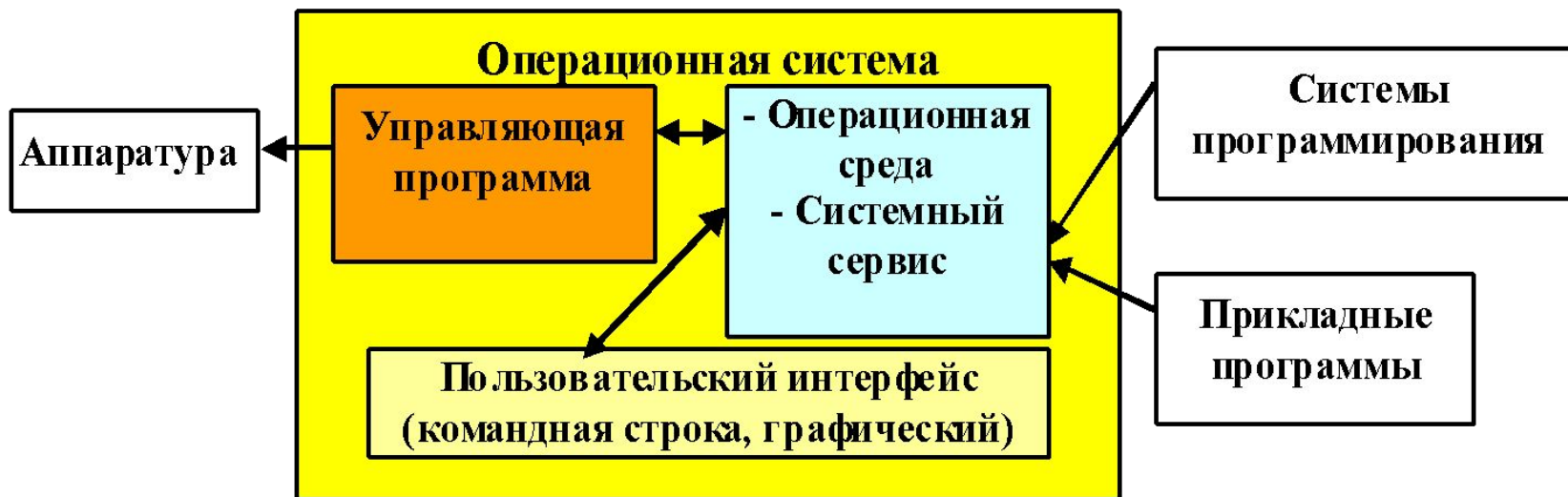
Задачи операционной системы

Операционная система автономного компьютера решает две основные задачи:

- управление исполнением программ процессором («организация вычислительного процесса»)
- управление ресурсами вычислительной системы

Ресурс - любой объект вычислительной системы (программный или аппаратный), который выделяется исполняемой программе.

Обобщенная структура ОС



-
- «Операционная среда» представляет вычислительную систему для пользователя и программ в «абстрактном» виде
 - «Системный сервис ОС» –набор программных функций (сервисов), предназначенный для прикладных программ. В основном, обслуживают запросы программ на обращение к ресурсам.

Набор всех системных функций ОС и правил их вызова программами называют **API (application program interface)-интерфейс для прикладных программ** в данной ОС.

1-2. Система прерываний

- Управление исполнением программ и управление системными ресурсами в операционной системе основано на механизме прерываний.
- Система прерываний – это аппаратно-программный механизм в вычислительной системе, заставляющий процессор начать исполнять некую программу ОС («процедуру прерывания») в ответ на внешние или внутренние события («источники прерываний»)
- Типы прерываний: внешние, системные, программные

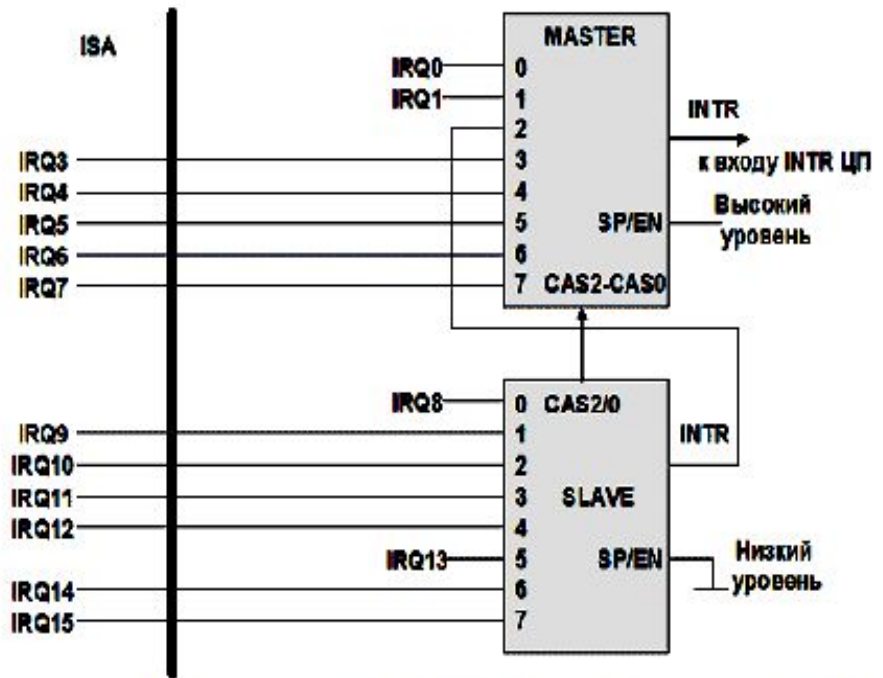
Типы прерываний в вычислительной системе

1. Внешние прерывания

Сигналы прерываний от внешних устройств (IRQ- Interrupt Request) поступают в **контроллер прерываний**.

КП программируется на определенную схему приоритетов прерываний.

Наиболее приоритетный запрос из КП поступает на вход прерывания **INTR процессора**



Каскадное соединение контроллеров прерываний

Маскируемые и немаскируемые внешние прерывания

- Прерывания, поступающие с контроллера на вход **INTR** процессора, называют **«маскируемые»**. Их можно программно разрешить/запретить с помощью **флага IF** (команда **STI** – установить флаг, команда **CLI** – сбросить флаг)
- **«Немаскируемые»** – прерывания, поступающие на особый вход прерывания **NMI** процессора (non-masked interrupt). Эти прерывания невозможно программно запретить.

2. Системные прерывания - внутренние прерывания, генерируемые самим процессором, в ответ на обнаруженные им событиях при чтении команды из памяти, ее дешифрации или исполнении.

Например: несуществующий адрес памяти, неправильный код операции, попытка деления на ноль в команде деления и т.д.

3. Программные прерывания - создаются командой `INT n` из исполняемой программы

Обязанности ОС в системе прерываний

- За каждым источником прерывания в системе закрепить **номер прерывания**.
- Разместить в памяти процедуры (**«обработчики прерывания»**) для каждого источника прерывания
- Создать в памяти **Таблицу прерываний** с адресами (**сегмент: внутрисегментный адрес**) «обработчиков прерываний».

Замечание:

Процессор всегда **знает адрес** размещения Таблицы прерываний в памяти. В зависимости от режима работы процессора – этот адрес фиксирован или занесен в определенный регистр процессора.

Таблица прерываний для реального режима процессора (16-разрядные обработчики прерываний)

- Таблица состоит из **4-байтовых элементов (векторов)**.
- Один вектор содержит адрес одной процедуры: указатель сегмента (2 байта) и внутрисегментное смещение (2 байта).
- **Номер прерывания** в команде INT – это номер вектора, с адресом процедуры (0, 1, 2,n)

Пример:

Указатель сегмента	Смещение	
0060	0000	Вектор прерывания 0
ff00	0000	Вектор прерывания 1
ff00	0080	Вектор прерывания 2
...	...	

В результате выполнения команды `Int 2h` будет вызвана процедура ОС, размещенная в памяти с адреса **ff00 : 0080**

Работа процессора при поступлении прерывания

- *Принять по ШД от контроллера прерываний n -номер прерывания (только для внешних прерываний)*
- **Сохранить в стеке CS:IP** и текущее состояние регистров процессора
- Вычислить адрес нужного вектора Таблицы прерываний:
адрес Таблицы прерываний + $n * 4$
(4 байта -размер вектора для 16-разр.программ)
- Прочитать в регистры CS и IP значения «сегмент» и «смещение» из вектора прерывания. Так, процессор начнет исполнение процедуры обработки прерывания
- Последняя команда процедуры – **IRET** заставит процессор **восстановить из стека** в CS и IP адрес возврата, в регистры – их прежнее состояние. Так, продолжится выполнение прерванной программы

Пример: иллюстрация вызова обработчика прерываний (на примере программного прерывания)

Вызов процедуры по прерыванию

Таблица векторов прерываний

