

# Операционные системы

## лекции 3, 4

- Планирование распределения времени ЦП в многозадачных системах
- Обработка прерываний
- Системные вызовы. Интерфейс библиотечных функций и системных вызовов

**30.10.2010**

# Способы планирования времени ЦП

название, примеры обл. использования	условия смены выполняющейся задачи	преимущества
<b>пакетный режим</b>  суперкомпьютеры	1. завершение 2. запрос выполнен. ввода-вывода	эффективное использование вычислительной мощности ЦП
<b>режим разделения времени</b> программы, ведущие диалог с пользователем	1. завершение 2. оконч. выделенного кванта времени	задачи не оказываются заблокированными на долгий срок
<b>системы реального времени,</b>  автопилот	1. завершение 2. прерывание	гарантированное время реакции на непредпед. внешние события




# Алгоритмы планирования времени ЦП

## **2 вида задач:**

1. Выполняющие много вычислений
2. Требующие выполнения большого колич. операций ввода/вывода

Алгоритмы планирования: без переключения, с переключением

## **Алгоритмы для систем пакетной обработки**

1. Очередь
  2. Кратчайшую задачу выполнять первой  
среднее обратное время задачи
  3. Первой выполнять задачу, которая скорее завершится  
(наименьшее оставшееся время выполнения)
  4. Трехуровневое планирование
    - планировщик доступа
    - планировщик памяти
    - планировщик времени процессора
- 



# Алгоритмы планирования времени ЦП

## Алгоритмы для интерактивных систем

1. Циклическая очередь
2. Приоритетное планирование
  - динамическое изменение приоритетов
  - группировка процессов в классы по приоритетам
3. Несколько очередей процессов
4. Самый короткий — следующий
5. Гарантированное планирование
6. Потерейное планирование
7. Справедливое планирование





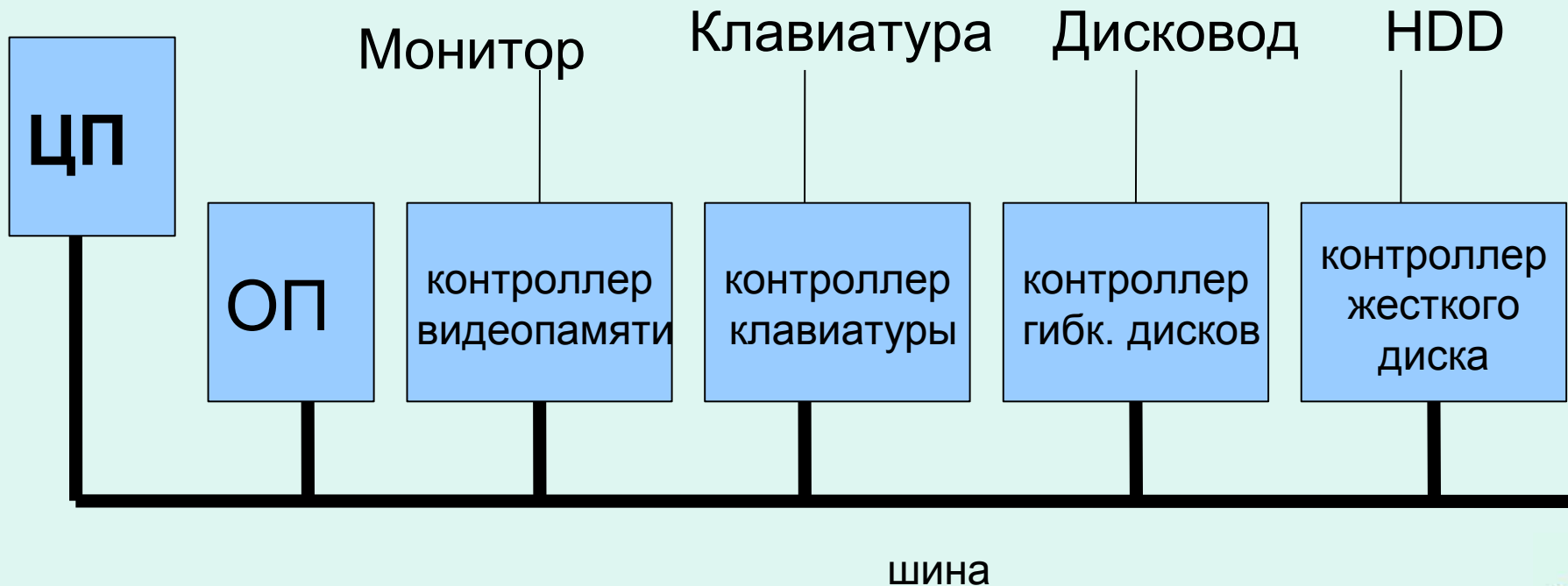
# Аппаратная поддержка мультипрограммного режима

1. Аппарат прерываний
2. Разделение режимов работы процессора: привилегированный и ограниченный
3. Защита памяти
4. Таймер




# Обзор аппаратного обеспечения компьютера

## Абстрактная модель ПК





# Аппарат прерываний

1. Внешние прерывания
  2. Внутренние прерывания
  3. Программные прерывания (системные вызовы)
- 




# Внешние прерывания

**Примеры ситуаций, в которых могут возникнуть внешние прерывания**

**Общая схема действий, выполняемых при возникновении и обработке прерывания.**

**Аппаратная часть + программная часть**

- 1. Устройство устанавливает на шине сигнал «запрос прерывания»**
  - 2. Процессор выполняет вычисления до точки, где их можно прервать (чтобы потом восстановить), выставляет на шине сигнал подтверждения прерывания**
  - 3. Блокировка остальных прерываний**
  - 4. Передача номера прерывания (число идентифицир. устройство)**
  - 5. Малое упрятывание: сохранение счетчика команд (PC) и слова состояния процессора (PSW)**
  - 6. Устанавл. привилег. режим работы ЦП, передается управление на спец. процедуру — ОС**
- 





# Обработка прерываний

(продолжение)

управление передано на опред. адрес программы (ОС)

## КОРОТКОЕ ПРЕРЫВАНИЕ

Управление сразу возвращается активной задаче  
(IRET)

Отмена блокировки прерываний, восстановление PC и PSW

## ДЛИННОЕ ПРЕРЫВАНИЕ

Полное упрятывание, разблокировка прерываний, выполнение функции обработчика





# Внутренние прерывания

**Примеры ситуаций, в которых могут возникнуть внутренние прерывания**

**Деление на 0, обращение к памяти по недопустимому адресу, попытка выполнения недопустимой команды (инструкции)**


**Общая схема действий, выполняемых при возникновении и обработке прерывания.**

**Аппаратная часть + программная часть**

- 1. Устройство устанавливает на шине сигнал «запрос прерывания»**
- 2. Процессор выполняет вычисления до точки, где их можно прервать (чтобы потом восстановить), выставляет на шине сигнал подтверждения прерывания**
- 3. Блокировка остальных прерываний**
- 4. Передача номера прерывания (число идентифицир. устройство)**
- 5. Малое упрятывание.**

...





# Программные прерывания (системные вызовы)

Системный вызов — обращение пользовательской задачи (процесса) к ядру операционной системы за услугами.

Для переключения в привилегированный режим используется прерывание.


Прерывание происходит по инициативе пользовательской задачи.

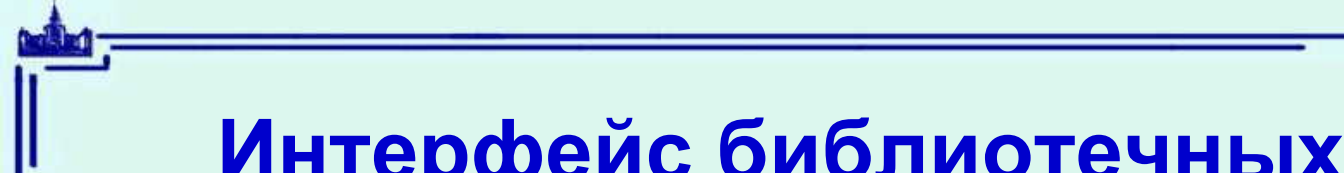
**Вопрос:**

scanf            printf

sin              strlen

getchar        fopen





# Интерфейс библиотечных функций языка C и системных вызовов

errno.h

errno

-----

string.h

char\* strerror (int errnum);

-----

stdio.h

void perror (const char\* s);

-----

```
if (<имя_системного_вызова>(<параметры>) == -1) {  
    perror(<имя_системного_вызова>);  
    return 1; /* или exit(1); */  
}
```



# Пример системного вызова - open

Дескриптор

ТОФ (Таблица Открытых Файлов)

```
#include <sys/types.h>
```

```
#include <sys/stat.h>
```

```
#include <fcntl.h>
```

```
int open(const char *pathname, int flags);
```

```
int open(const char *pathname, int flags,  
         mode_t mode);
```

```
O_RDONLY      O_WRONLY      O_RDWR
```

```
O_CREAT      O_EXCL
```

```
O_TRUNC      O_APPEND
```