

# **Системы реального времени**

Автор: к.т.н., доц. К.Е. Климентьев  
Лекции: 429 и 517 ауд.  
Лабораторные работы: 512 ауд.

## Литература

1. Климентьев К.Е. Системы реального времени. Обзорный курс лекций. — Самара: Самар. гос. аэрокосм. ун-т. Самара, 2008. — 45 с.
2. Олсон Г., Пиани Д. Цифровые системы автоматизации и управления. — СПб: Невский Диалект, 2001. — 557 с.
3. Михайлов А.А. Система реального времени. Программно-технический комплекс: учеб. пособие / Юж. — Рос. Гос. техн. Ун-т. — Новочеркасск: ЮРГТУ, 2010. — 292 с.
4. Древс Ю.Г. Системы реального времени: технические и программные средства: Учебное пособие. М.: МИФИ, 2010. — 320 с.
5. Денисенко В.В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. — М.: Горячая линия — Телеком, 2009. — 608 с.
6. Горев С.М. Автоматизация производственных процессов нефтяной и газовой промышленности. Курс лекций. — Петропавловск-Камчатский: КамчатГТУ, 2003. — 121 с.

# Классификация автоматизированных систем

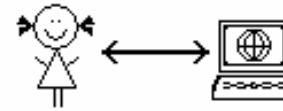
Задачи автоматизированных систем:

- получение информации;
- передача информации;
- обработка информации;
- хранение информации;
- представление (отображение) информации;
- управляющее и регулирующее воздействие на объекты внешнего мира.

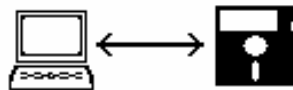
## 1. АИС (обработка)



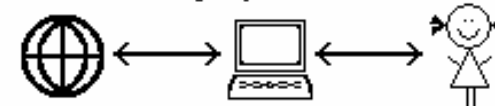
## 2. Моделирование и проектирование



## 3. Хранение



## 4. АСУ (управление)



## 5. Передача (связь)



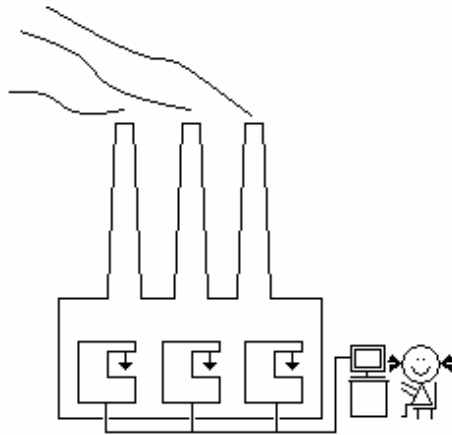
# Примеры автоматизированных систем управления

Различать термины:

- Автоматический VS Автоматизированный
- Управление VS Регулирование.

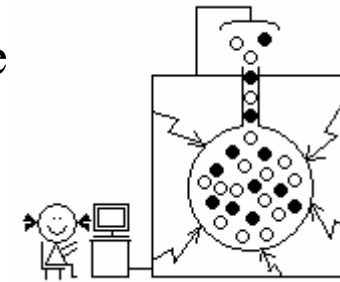
## АСУ ТП –

автоматизированные системы  
управления технологическими  
процессами



## АСНИ –

автоматизированные  
системы управления  
научными  
исследованиями и  
комплексными  
испытаниями



## Бортовые АСУ



## Встроенные АСУ



# Определение СРВ

**СРВ** – автоматизированная система с ограничениями на временные характеристики функционирования:

$$T_{\min} < T < T_{\max}$$

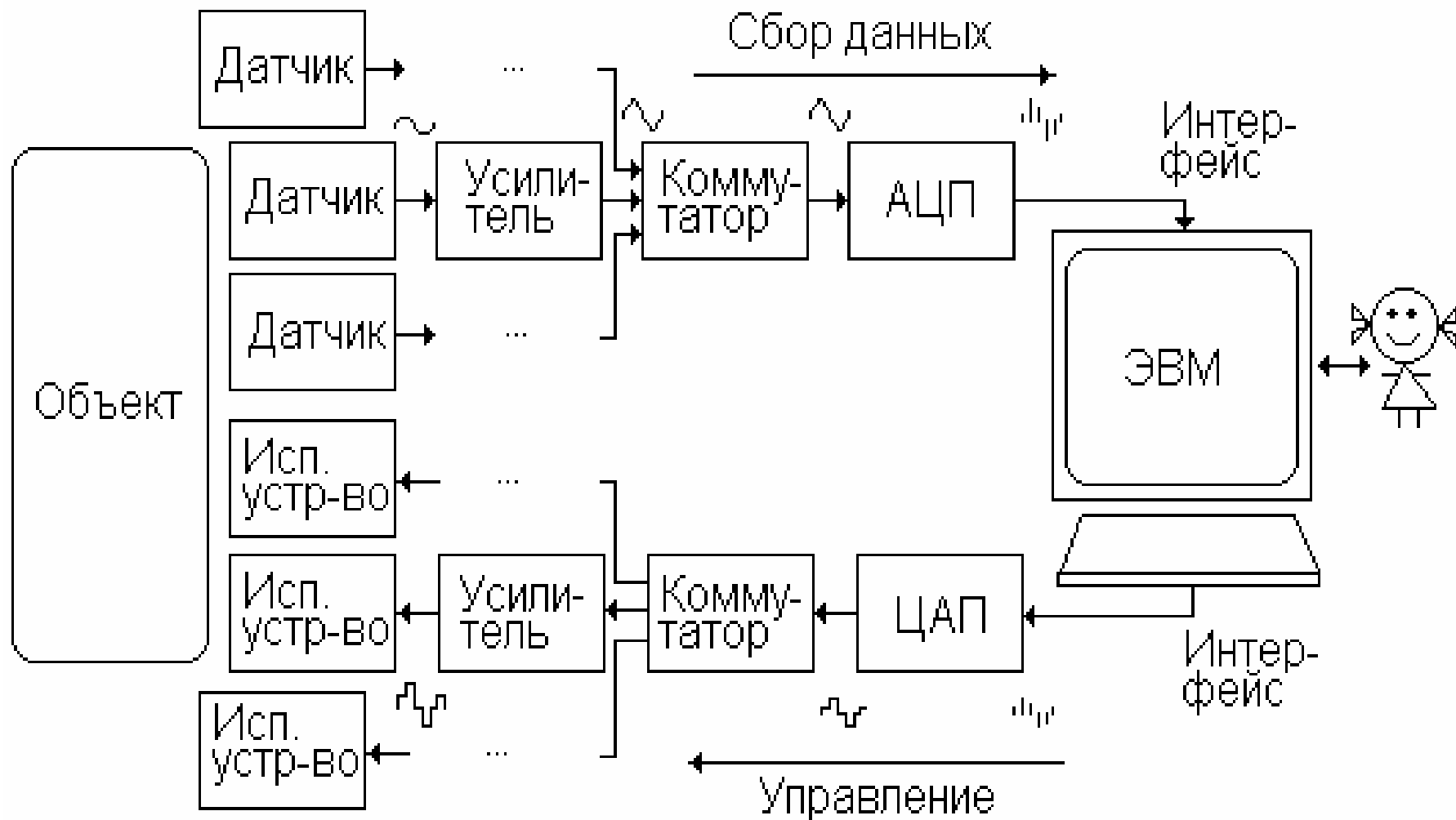
Примеры временных характеристик:

- моменты наступления внутренних событий;
- время реакции на внешние события;
- длительность выполнения какой-нибудь операции и т.п.

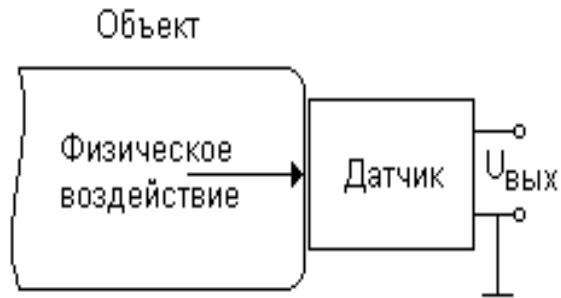
Назначение СРВ – взаимодействие с объектами внешнего мира в темпе процессов, протекающих в этих объектах.

СЖРВ и СМРВ

## Общая структура АСУ РВ



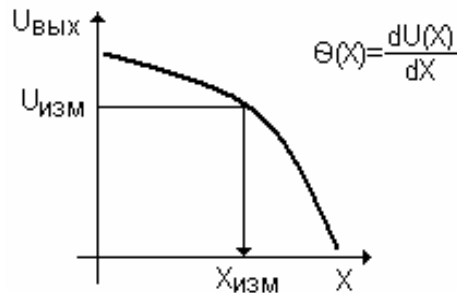
# Датчики



**Датчик** – устройство, преобразующее физическую величину в электрический сигнал. Информация может быть заключена в напряжении, силе тока или частоте сигнала.

Технические характеристики датчиков:

- диапазон измерения;
- функция преобразования и чувствительность;



- предельная погрешность.

**Сенсоры** – датчики, имитирующие органы чувств.

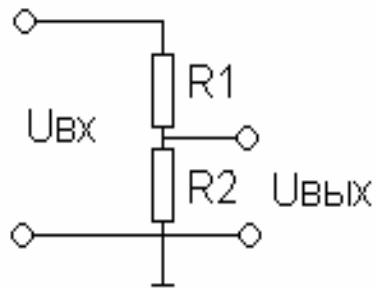
**Цифровые датчики** имеют на выходе не аналоговый сигнал, а 0 или 1 (+5В).

# Усилители и нормализаторы

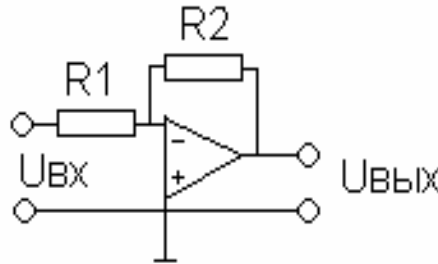


**Усилитель** – электронная схема, изменяющая уровень электрического сигнала, но не изменяющая его форму.

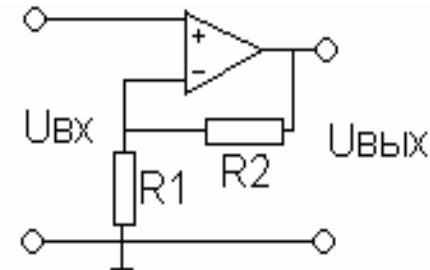
Примеры схем усилителей



$$U_{\text{ВЫХ}} = U_{\text{ВХ}} \frac{R_2}{R_1 + R_2}$$



$$U_{\text{ВЫХ}} = -U_{\text{ВХ}} \frac{R_2}{R_1}$$



$$U_{\text{ВЫХ}} = U_{\text{ВХ}} \left( 1 + \frac{R_2}{R_1} \right)$$

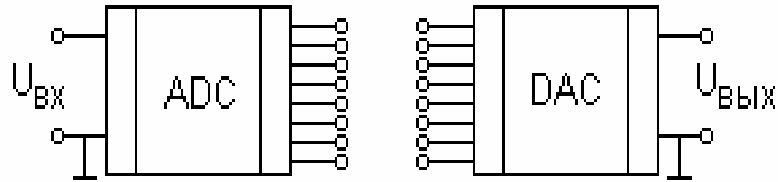
**Нормализатор** – усилитель, приводящий уровень сигнала к одному из стандартных диапазонов, например: 4..20 мА или -10В...+10В.

**Технические характеристики:**

- Входной диапазон;
- Функция преобразования;
- Предельная погрешность преобразования.



# АЦП и ЦАП

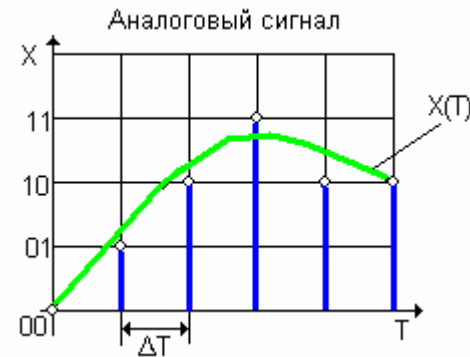


**АЦП** – микросхема, преобразующая электрический сигнал в двоичное число. **ЦАП** – выполняет обратное преобразование.

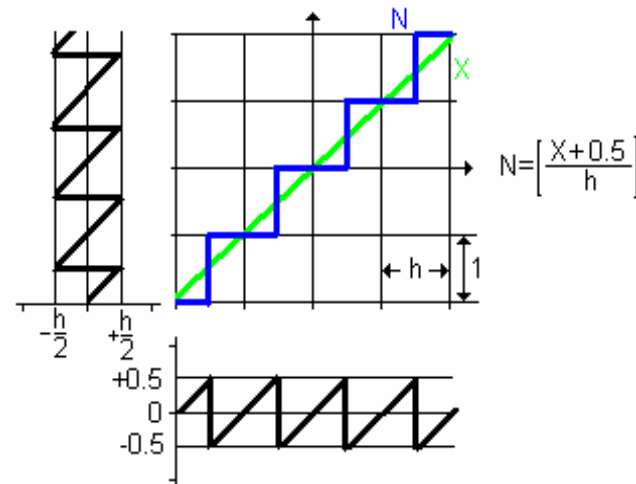
## Сущность преобразования:

- квантование по уровню и
- дискретизация по времени.

**Функция преобразования:**  
ступенчатая линия.



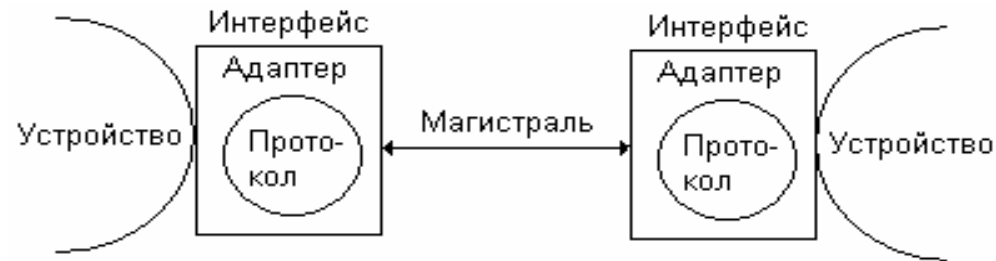
**Погрешность преобразования:**  
пилообразная функция с диапазоном  $-h/2..+h/2$ .



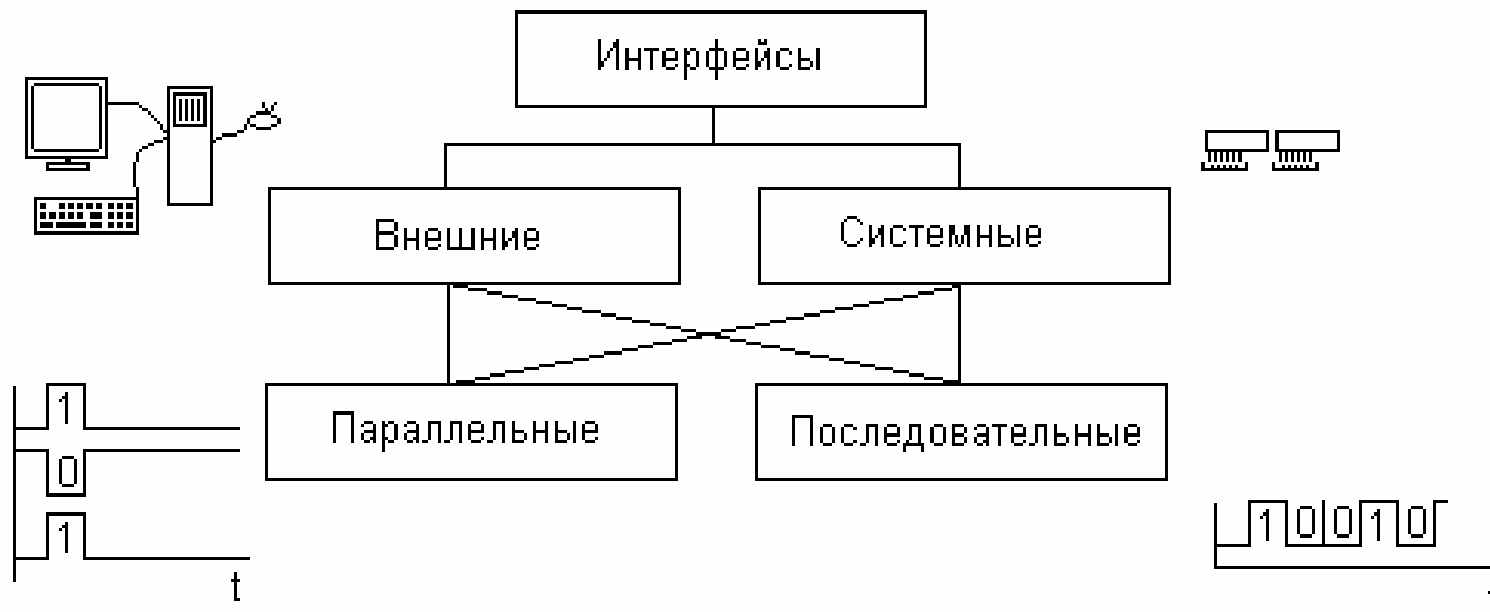
## Технические характеристики:

- входной диапазон;
- разрядность;
- время преобразования (или частота).

# Интерфейсы



Интерфейс – программно-аппаратный комплекс для обеспечения цифровой связи между двумя устройствами.



## Примеры интерфейсов

### RS-232.

**Назначение:** внешний

**Магистраль:** последовательная

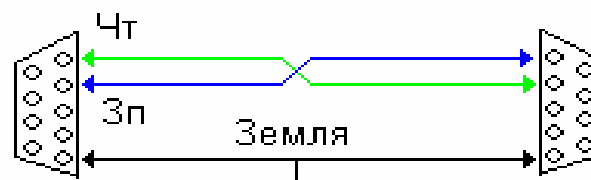
**Топология:** точка-точка

**Скорость передачи:** до 125 Кбит/с

**Расстояние:** до 15м на макс. скорости.

**Сигналы:** -3В...-12В и +3В...+12В

**Протокол:** см. рис.



### RS-485 и RS-422.

**Топология:** до 32 устройств

**Скорость передачи:** до 10 мбит/с

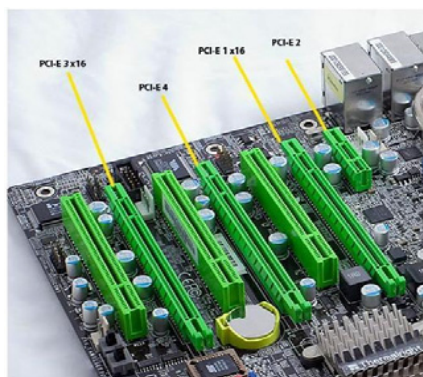
**Расстояние:** до 1200 м на макс. скорости

**Протокол:** нет, нужен внешний (Profibus, CanBus, ModBus...)

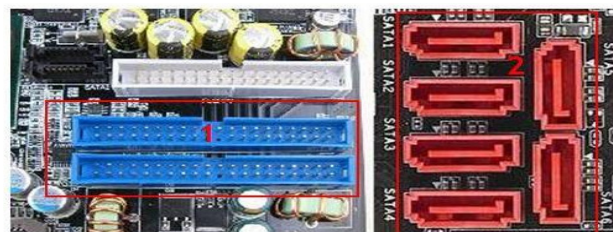
## Примеры интерфейсов (продолжение)



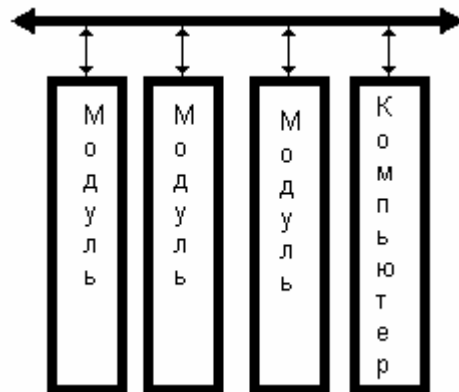
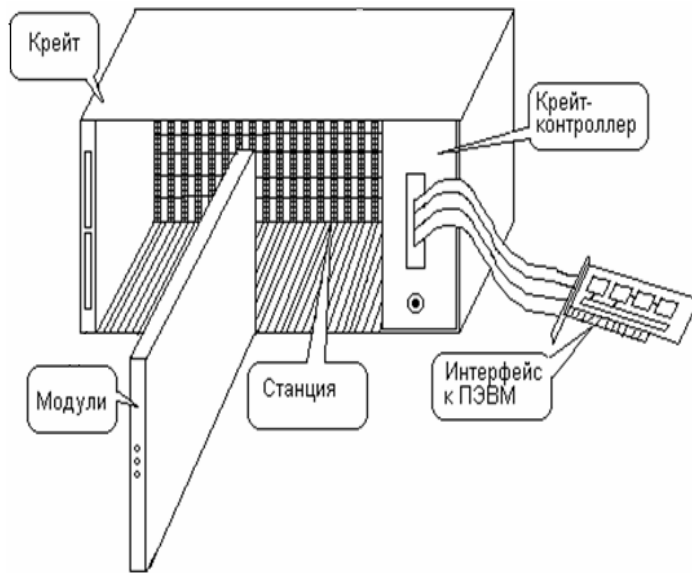
PCI



IDE (он же ATA) и SATA



# Магистрально-модульные интерфейсы



## Примеры:

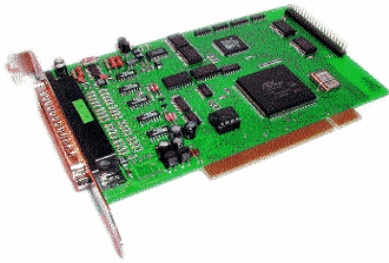
- CAMAC (KAMAK)
- VME/VXI
- PXI
- FieldPoint.





## Три способа подключения УСО к компьютеру

К системному интерфейсу    Ко внешнему интерфейсу    На общей шине ММИ



## Средства вычислительной техники

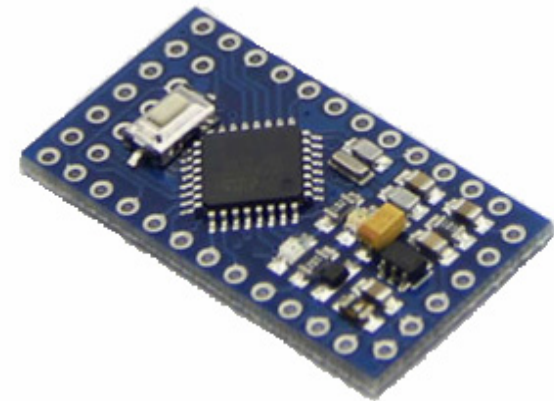
Промышленные  
компьютеры



Программируемые  
логические контроллеры



Микроконтроллеры



# Средства работы со временем

**RTC** – часы реального времени.

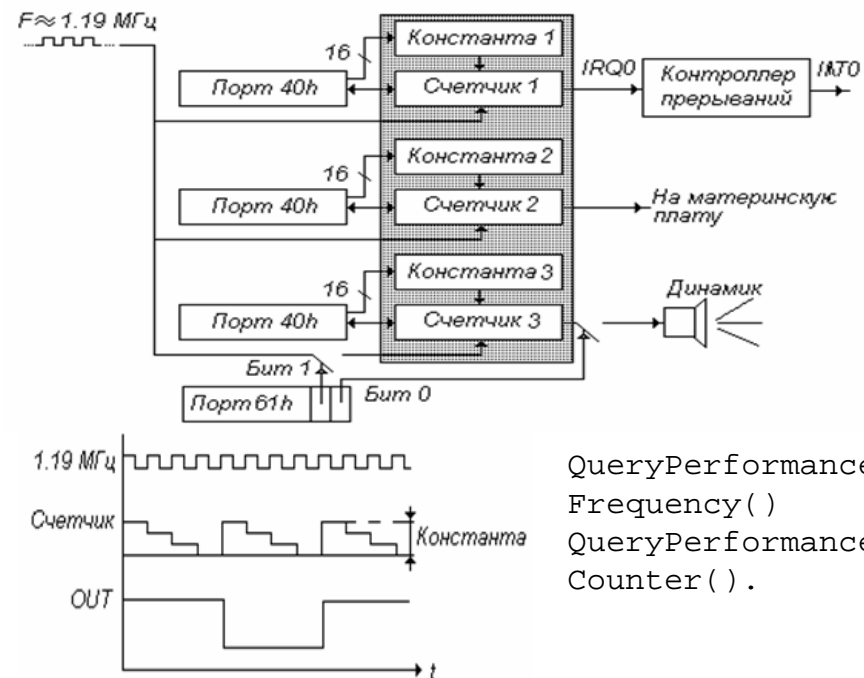
Режимы работы:

- пересчет времени (0.976 сек);
- будильник
- периодические прерывания.

00	секунды
01	секунды будильника
02	минуты
03	минуты будильника
04	часы
05	часы будильника
...	
0B	бит 5 - будильник, 7 - прерывания

**PIТ и HPET** – системные таймеры

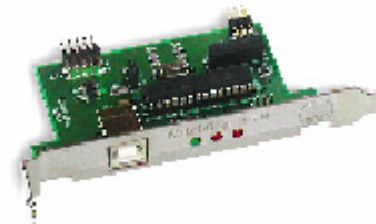
Режимы работы: генерация меандра.



## Счетчик тактов процессора

```
Rdtsc  
Mov H,EAX  
Mov L,EDX
```

## Сторожевой таймер (watchdog)



# Арматура

Шкаф (набор стоек)



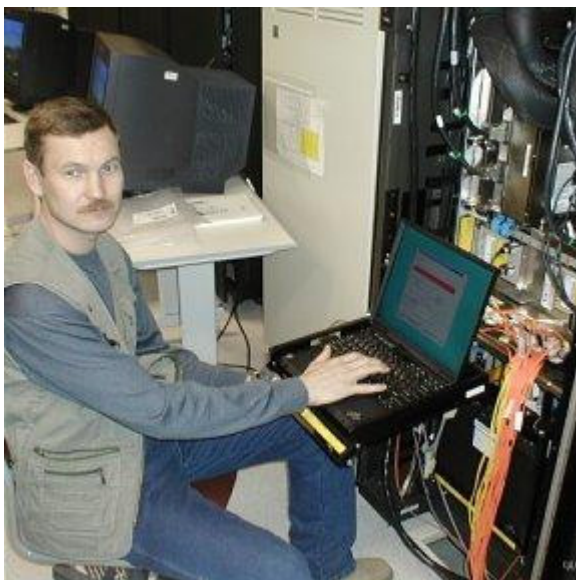
DIN-рейка



Операторская панель



Кросс-платформенное программирование



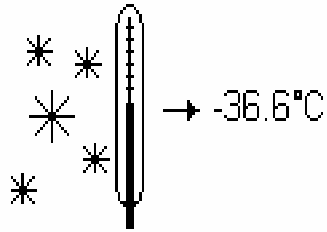
Геометрический стандарт  
«Евромеханика» (IEEE 1101):

- $1 U = 1.75'' = 44.45 \text{ мм}$ ;
- $1 TH = 0.2'' = 5.08 \text{ мм}$ .

Модули:  $3U \times 30TH$



# Метрология



**Измерение** = нахождение величины опытным путем с помощью специальных технических средств.

**Техническое средство** = средство измерения.

**Результат измерения** = значение величины, выраженное в единицах средства измерения.

**Значение величины:**

- **истинное** — такое, какое оно на самом деле (не знает никто);
- **действительное** — максимально близкое к истинному;
- **измеренное** — полученное в результате измерений.

**Погрешность измерения** = отклонение результата измерений от истинной величины:  $\Delta = X_{\text{изм}} - X$ .

Формы представления погрешности:

абсолютная:  $X_{\text{изм}} - X$

относительная:  $\frac{X_{\text{изм}}}{X} \times 100\%$

приведенная:  $\frac{X_{\text{изм}}}{X_{\text{max}} - X_{\text{min}}} \times 100\%$

## Метрология (продолжение)

### Погрешность по причине возникновения:

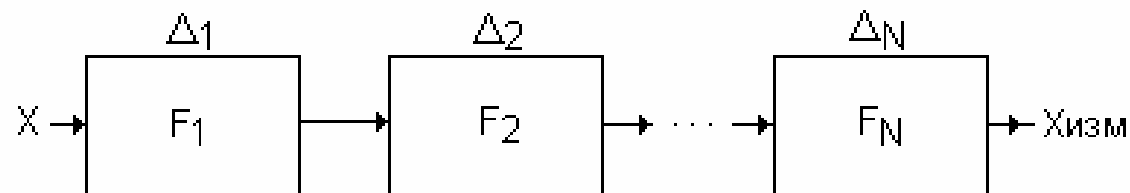
- инструментальная (приборная);
- методическая;
- в результате сбоя или промаха.

### Погрешность по характеру проявления:

- случайная
- систематическая.

Погрешность ИК и каждого компонента : **инструментальная и случайная.**

Задача:



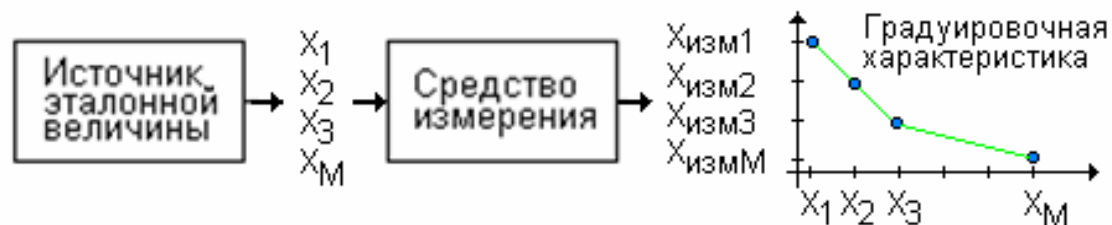
$$X_{изм} = F_z(X) = X + \Delta_z$$

$$\Delta_z = \Delta_1 * \Delta_2 * \dots * \Delta_N = ?$$

$$F_z = F_1 * F_2 * \dots * F_N = ?$$

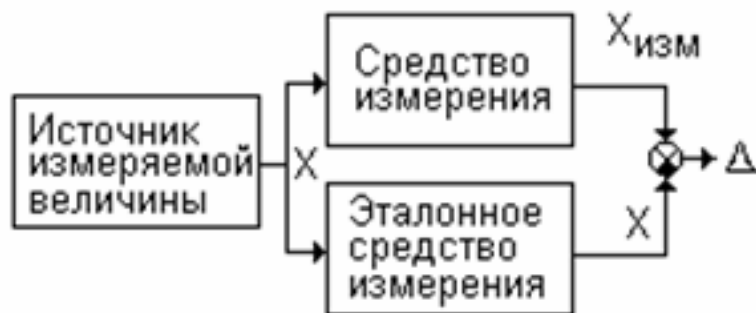
# Экспериментальные методы

## 1. Определение функции преобразования - градуировка



## 2. Определение предельной погрешности

### а) Метод эталонного компонента

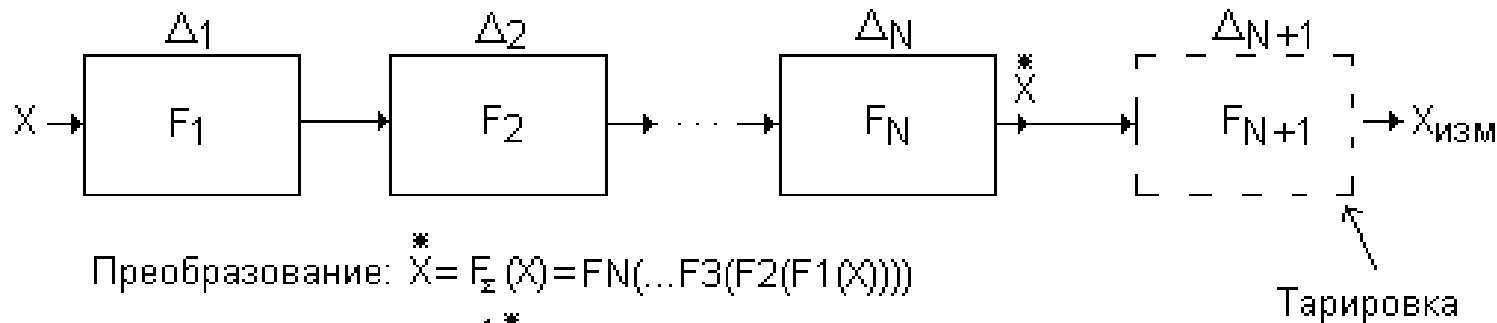


### б) Метод источника эталонной величины



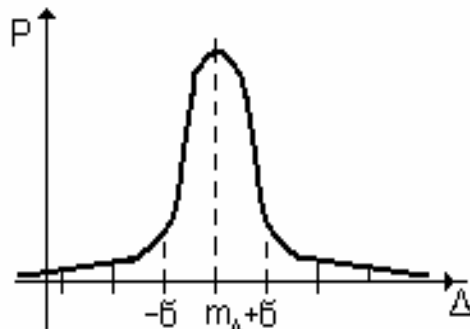
# Расчетные методы

## 1. Определение функции преобразования



## 2. Определение предельной погрешности

Математическая модель погрешности = нормально распределенная СВ



$\pm\Delta$	P
$\pm\delta$	0.67
$\pm 1.96\delta$	0.95
$\pm 3\delta$	0.997

Обычно  $m = 0$ ,  $P_{\text{дов}} = 1.95$ ,  
значит  $\Delta_{\min} = -1.96\delta$ ,  $\Delta_{\max} = +1.96\delta$

Все вычисления только над безразмерными величинами, т.е. над  $\gamma$  или  $\delta$  !!!

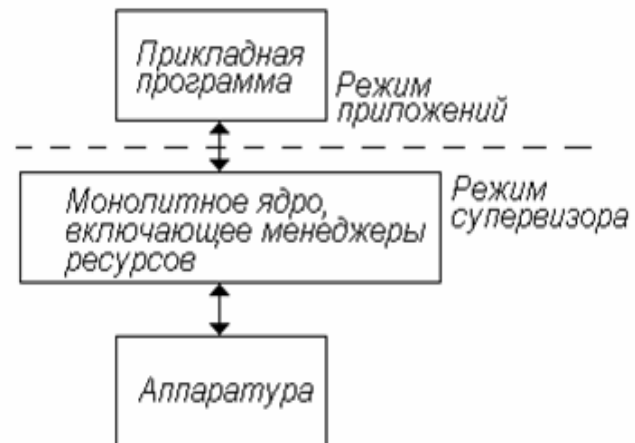
Первый способ: 
$$\begin{cases} m_{\Sigma} = m_1 + m_2 + \dots + m_M = \sum_{i=1}^M m_i \\ \sigma_{\Sigma} = \sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_M^2} = \sqrt{\sum_{i=1}^M \sigma_i^2} \end{cases}$$

Второй способ: 
$$\begin{cases} \Delta_{\min\Sigma} = \Delta_{\min 1} + \Delta_{\min 2} + \dots + \Delta_{\min M} \\ \Delta_{\max\Sigma} = \Delta_{\max 1} + \Delta_{\max 2} + \dots + \Delta_{\max M} \end{cases}$$

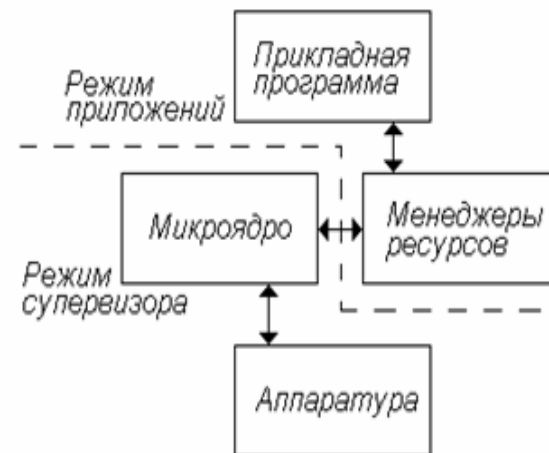
# Операционные системы



## Монолитная архитектура



## Микроядерная архитектура

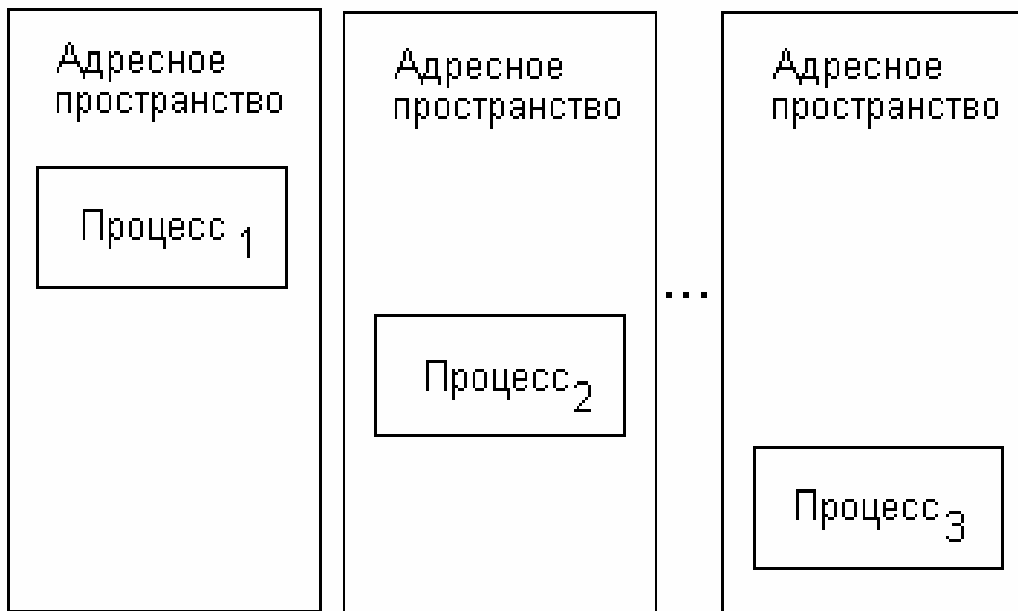


**В ОСРВ многозадачность и ввод-вывод устроены особым образом**

# Многозадачность

Задача = программа, предназначенная для одновременной работы с другими программами (задачи)

## А) Процессы



## б) Потоки (нити)



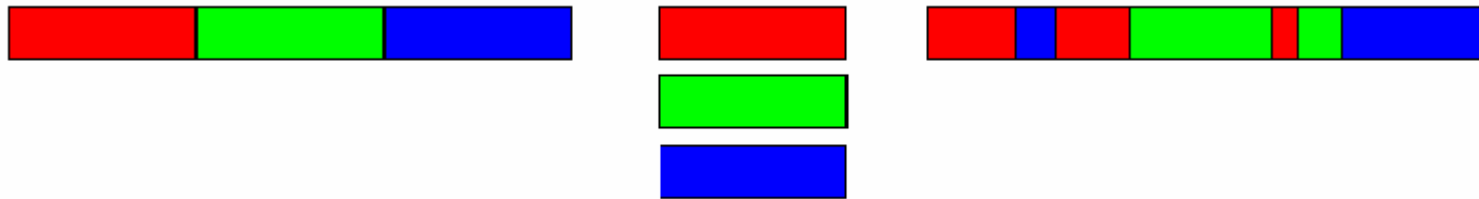
# Многозадачность (продолжение)

Организация выполнения нескольких задач

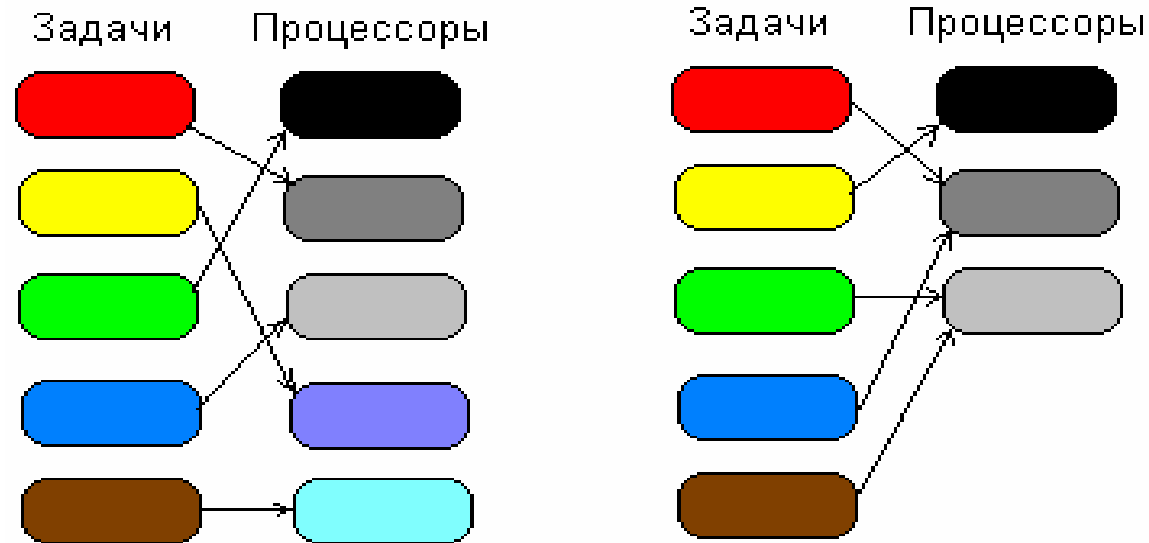
А) Последовательная  
(пакетный режим)

Б) Истинная  
параллельность

В) Псевдо- (квази-)  
параллельность



Аппаратные возможности



## Многозадачность (продолжение)

Диспетчер задач Windows

Файл Параметры Вид Завершение работы Справка

Приложения Процессы Быстродействие Сеть Пользователи

Имя образа	PID	Имя пользоват...	ЦП	Память	Баз.пр.	Потоков
VMNAT.EXE	4024		00	2 324 КБ	Средний	4
ALG.EXE	3936	LOCAL SERVICE	00	3 376 КБ	Средний	7
wscntfy.exe	3772	Administrator	00	2 072 КБ	Средний	1
vmnetdhcp.exe	3112		00	1 968 КБ	Средний	2
WMIPRVSE.EXE	3012		00	6 548 КБ	Средний	7
wuauclt.exe	2836	SYSTEM	00	8 296 КБ	Средний	9
vmware-usbarbitr...	2384		00	2 668 КБ	Средний	2
nimxs.exe	2012		00	6 756 КБ	Средний	10
LKTSRV.EXE	1856		00	4 068 КБ	Средний	13
LKADS.EXE	1812		00	3 948 КБ	Средний	7
LKCITDL.EXE	1780		00	4 288 КБ	Средний	11
FsUsbExService.Exe	1756		00	2 524 КБ	Средний	2
SVCHOST.EXE	1704		00	23 392 КБ	Средний	89
SVCHOST.EXE	1644		00	3 208 КБ	Средний	4
taskmgr.exe	1488		01	4 500 КБ	Высокий	3
NPSAgent.exe	1296		00	4 164 КБ	Средний	4
ATI2EVXX.EXE	1276		00	4 640 КБ	Средний	5
CTFMON.EXE	1232		00	2 892 КБ	Средний	1
hqtray.exe	1224		00	4 972 КБ	Средний	2
Winampa.exe	1220		00	1 936 КБ	Средний	1
DAEMON.EXE	1164		00	7 012 КБ	Средний	2

☒ Отображать процессы всех пользователей

Завершить процесс

Процессов: 44    Загрузка ЦП: 2%    Выделение памяти: 223МБ / 6



## Многозадачность (продолжение)

```
top - 21:15:25 up 2 min,  1 user,  load average: 0.35, 0.43, 0.18
Tasks:  78 total,   1 running,  77 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.3%sy,  0.0%ni, 99.3%id,  0.0%wa,  0.0%hi,  0.3%si,  0.0%st
Mem:    250984k total,  247236k used,    3748k free,    37184k buffers
Swap:      0k total,    0k used,    0k free,   166704k cached
```

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	772	304	260	S	0.0	0.1	0:04.94	init
2	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	15	-5	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/0
5	root	15	-5	0	0	0	S	0.0	0.0	0:00.02	events/0
6	root	15	-5	0	0	0	S	0.0	0.0	0:00.04	khelper
111	root	15	-5	0	0	0	S	0.0	0.0	0:00.06	kblockd/0
113	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
114	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kacpi_notify
175	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	ata/0
176	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	ata_aux
177	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	ksuspend_usbd
183	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	khubd
186	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kseriod
221	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pdflush
222	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pdflush
223	root	15	-5	0	0	0	S	0.0	0.0	0:00.04	kswapd0
267	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0

## Многозадачность (продолжение)



**Задачи переключаются по собственному желанию.**

**Достоинства:**

- Простота реализации
- Предсказуемость поведения задач

**Недостатки:**

- Неустойчивость к сбоям.

**Примеры:** Windows 1.X - 3.X, некоторые ОСРВ.

**Задачи переключаются операционной системой**

**Недостатки:**

- Сложность реализации
- Непредсказуемость поведения задач

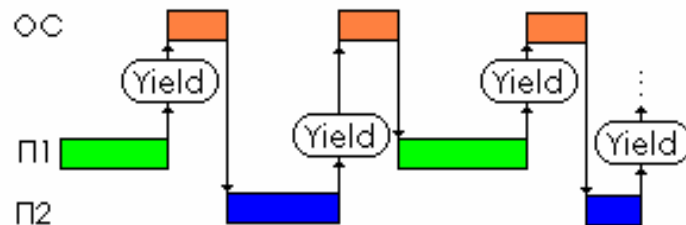
**Достоинство:**

- Устойчивость к сбоям.

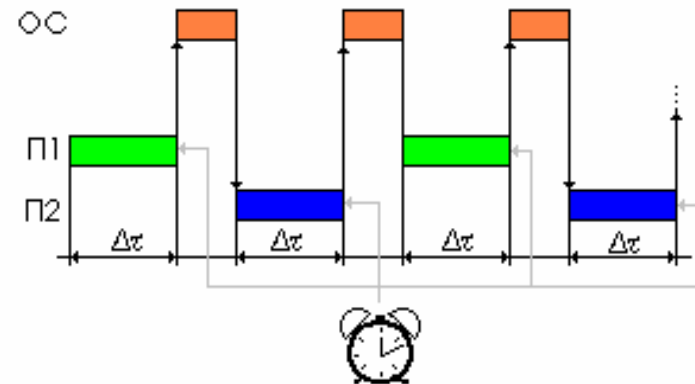
**Примеры:** все современные ОС

## Многозадачность (продолжение)

А) Кооперативная м/з



Б) Вытесняющая м/з



Причины переключения задач

Кооперативная м/з	Вытесняющая м/з
	Желание ОС
Желание задачи	Желание задачи
Желание пользователя	Желание пользователя
Завершение задачи	Завершение задачи

## Многозадачность (продолжение)

```
// Вывод строки по отдельным буквам
printstr(char *s) {
    int i = 0;
    while (s[i] != 0) cout << s[i++];
}
```

### А) Кооперативная м/з

```
while (1) {
    printstr("Hello ");
    Yield();
}
```

```
while (1) {
    printstr("world!\n");
    Yield();
}
```

```
Hello world!
Hello world!
Hello world!
```

### Б) Вытесняющая м/з

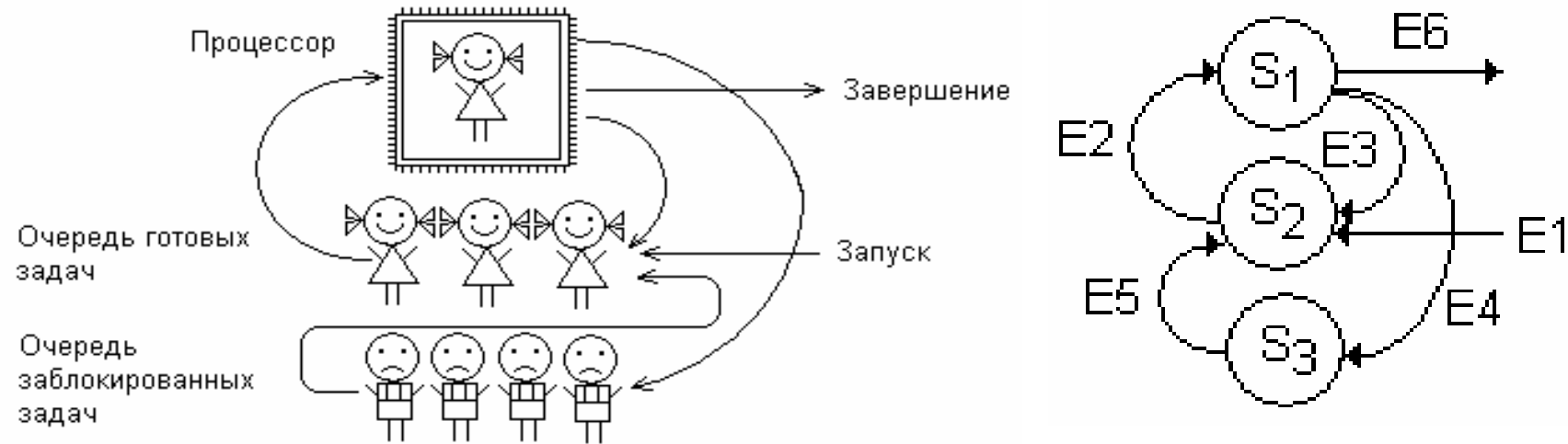
```
while (1) {
    printstr("Hello ");
}
```

```
while (1) {
    printstr("world!\n");
}
```

```
HelloHellworld!
wooHell Hellrld!
worl
```

# Многозадачность (продолжение)

Граф многозадачности



## Состояния:

$S_1$  — выполнение;  $S_2$  — ожидание;  $S_3$  — блокировка.

## События:

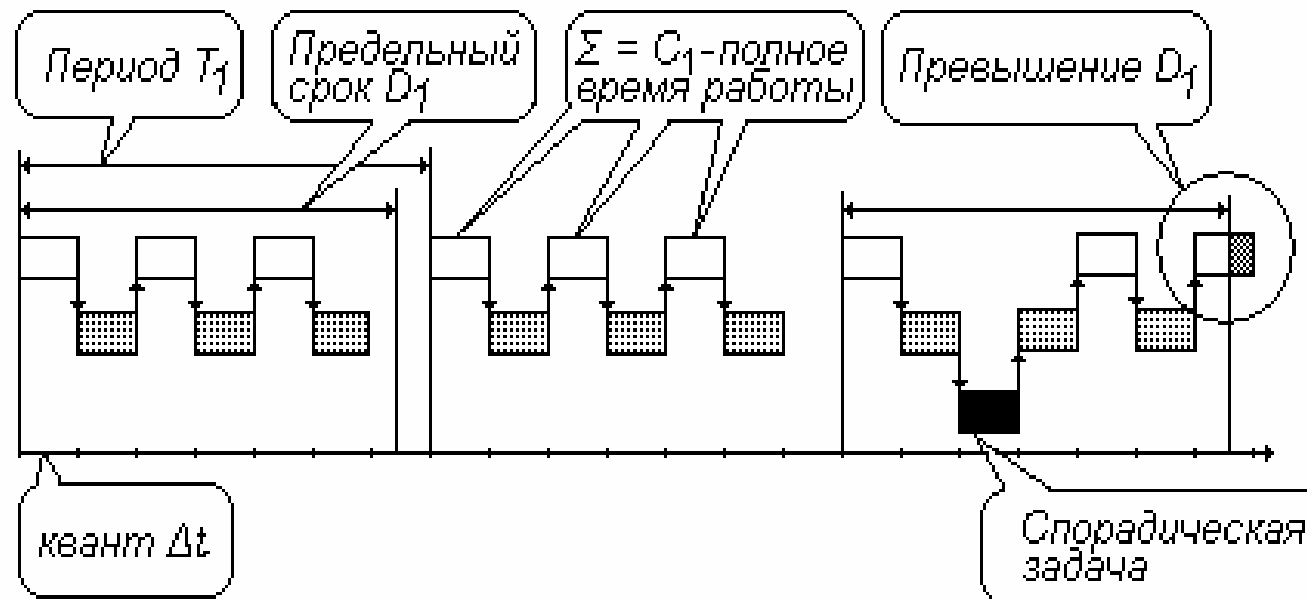
$E_1$  — запуск;  $E_2$  — начало выполнения;  $E_3$  — вытеснение;  $E_4$  — блокировка;  $E_5$  — снятие блокировки;  $E_6$  — завершение.

# Многозадачность (продолжение)

## Модель многозадачности

Типы задач:

- 1) периодические;
- 2) спорадические;
- 3) фоновые.

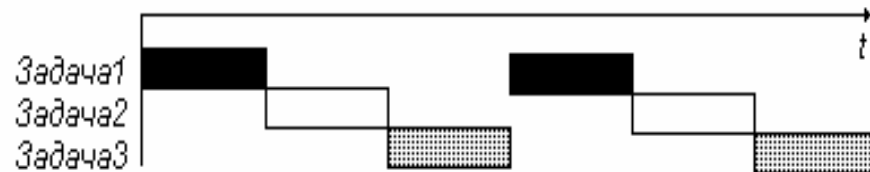


Условие работоспособности: 
$$\mu = \sum_{i=1}^N \frac{C_i}{T_i} \leq 1$$

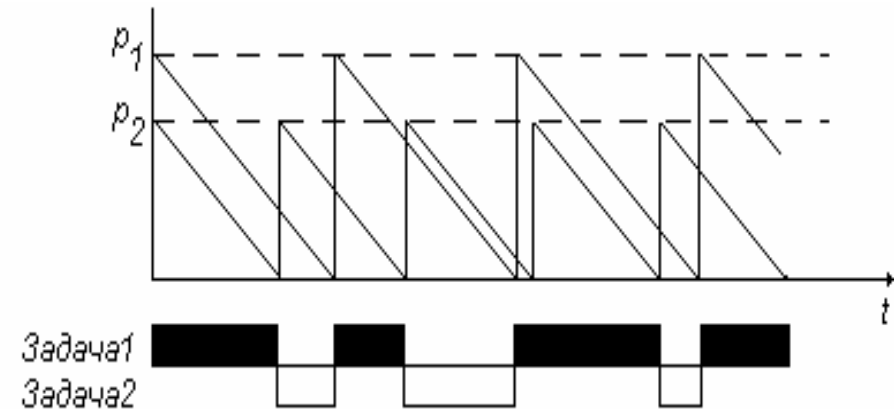
# Многозадачность (продолжение)

Алгоритмы работы планировщика задач  
(алгоритмы диспетчеризации)

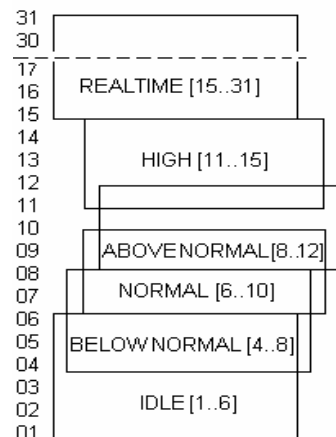
## 1) RR - Round Robin («карусель»)



## 2) UNIX (старение приоритетов)



## 3) Windows (классы приоритетов)



### Факторы:

- Открытое окно
- Фокус ввода
- «Возраст» задачи
- «Голод» задачи

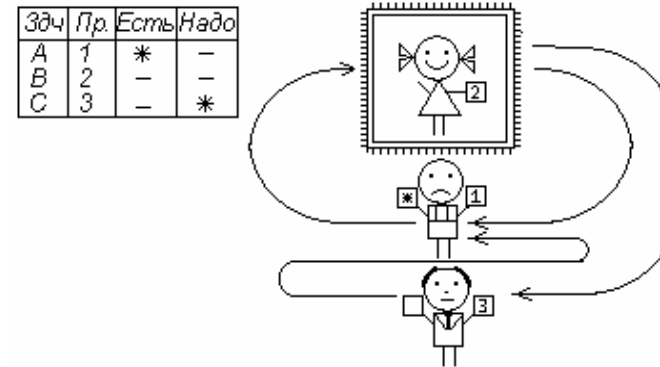
## 4) Алгоритмы реального времени:

- **RMS** (Rate Monotonic Scheduling) – чем меньше  $T$ , тем выше приоритет.
- **EDF** (Earliest Deadline First) – чем меньше до  $D$ , тем выше приоритет.

# Многозадачность (продолжение)

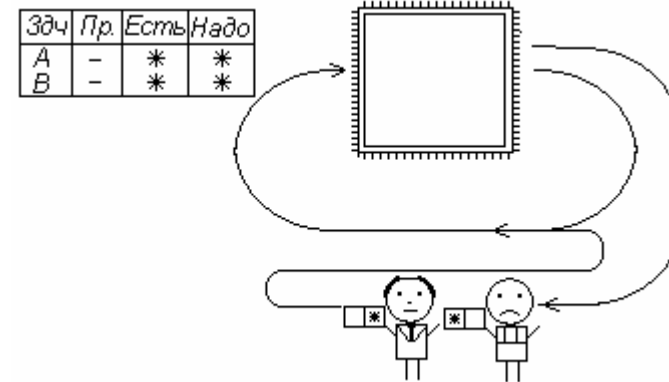
## Проблемы диспетчеризации:

- 1) «Голодание» задач (job starvation),  
решение – «разгон» приоритетов (priority boosting)
- 2) Инверсия приоритетов;



решение – «наследование» приоритетов;

- 3) «Гонки» (race condition);
- 4) «Взаимоблокировка» (deadlock).



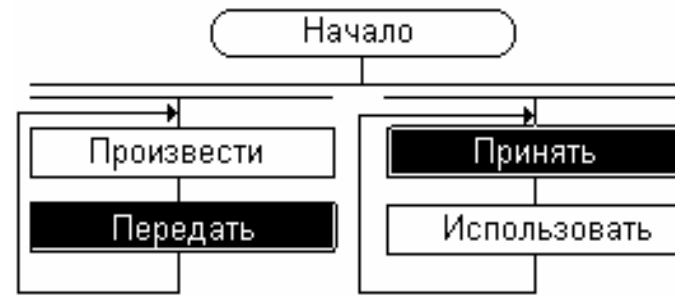


# Многозадачность (продолжение)

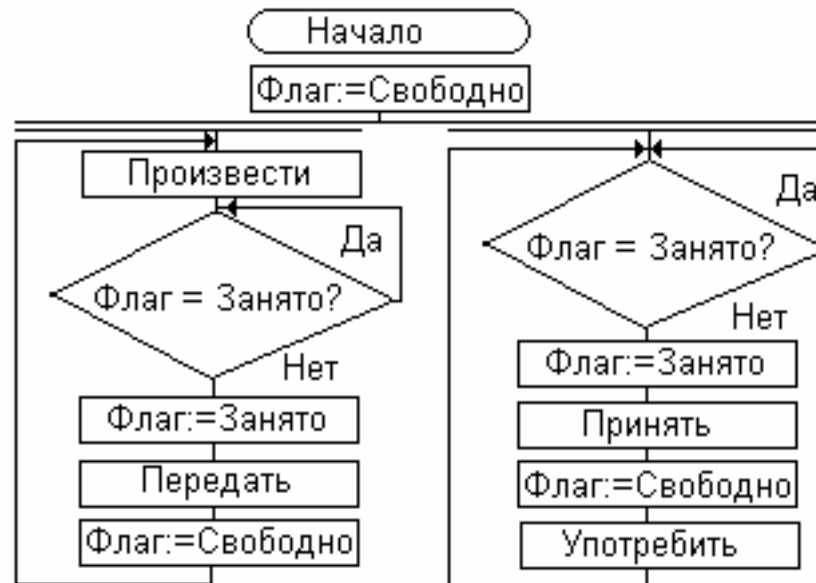
## Модель «поставщик-потребитель»,

условия: 1) одновременность доступа; 2) попеременность доступа.

А) Однозадачный алгоритм      Б) Многозадачный алгоритм без синхронизации

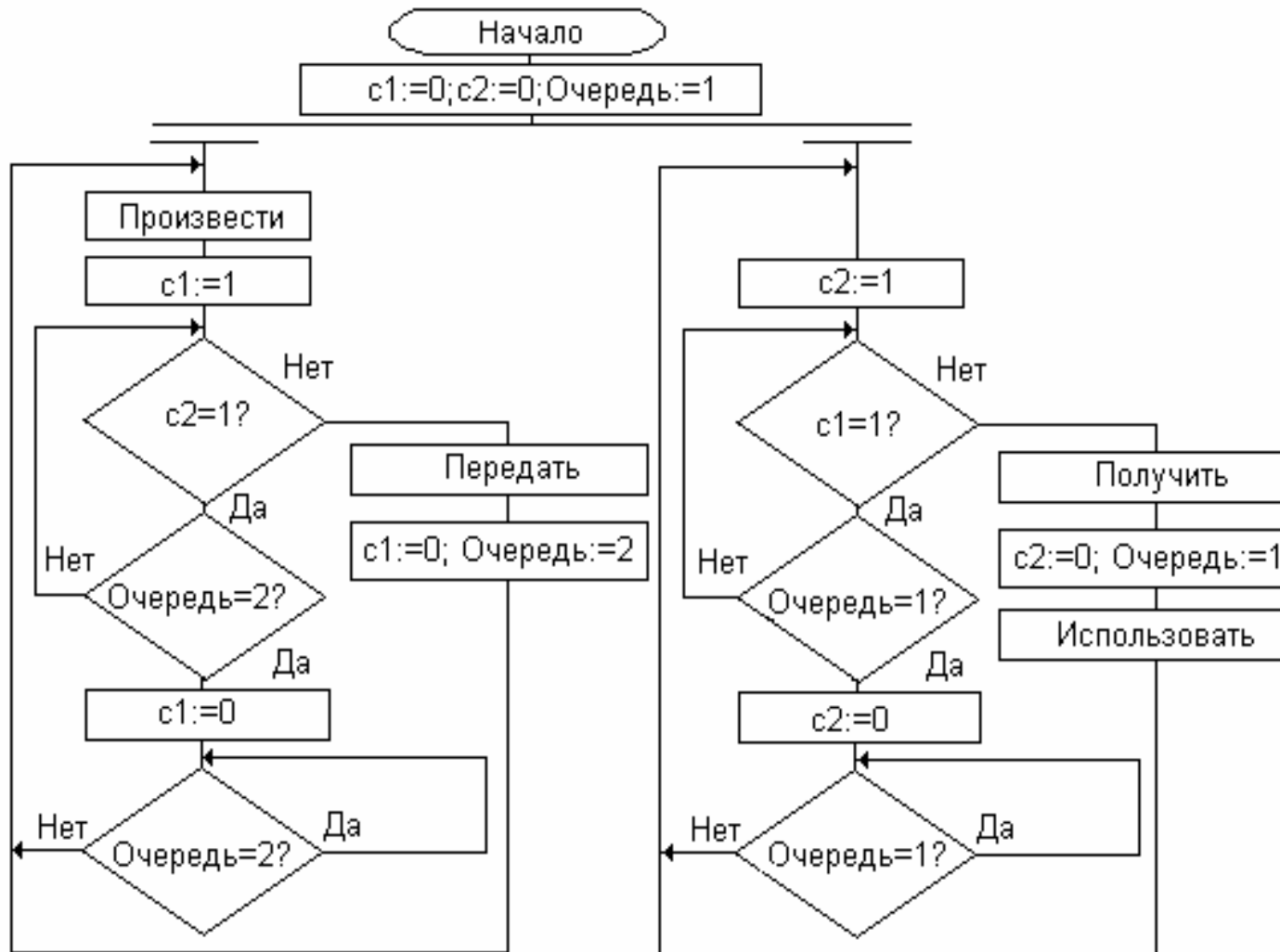


В) алгоритм с блокирующим флагом



## Многозадачность (продолжение)

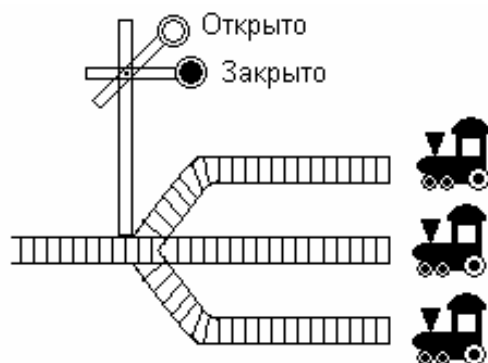
### Г) Алгоритм Деккера-Холта



Другой вариант: алгоритм Петерсона

# Многозадачность (продолжение)

## Д) Алгоритм с использованием семафоров Дейкстры



1. Целая переменная  $S$

2.  $P(S)$  - операция «Оградить»

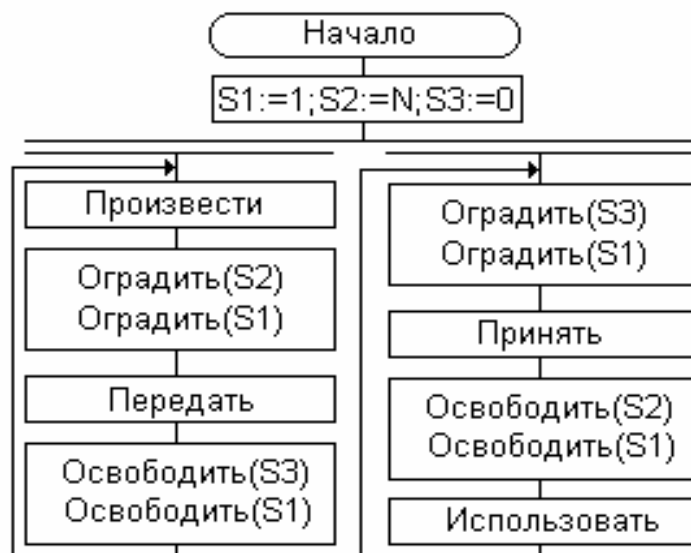
- $S := S - 1$

- Если  $S < 0$ , то текущая задача встает в очередь

3.  $V(S)$  – операция «Освободить»

- $S := S + 1$

- Если  $S > 0$ , то 1-я в очереди задача продолжает работу



# Многозадачность (окончание)

## Особенности старта процессов в UNIX

```
x = fork();  
cout << x;  
if (!x) { cout << " Я новенький"; exec(...); }  
else      cout << " Я старенький";
```

1 Я старенький                      0 Я новенький

## Финальный обзор методов синхронизации

	Windows	UNIX
Спин-блокировка	(+)	?
Семафор	+	+
Мутекс	+	+
Critical section	+	-
Событие	+	-
Сигнал	-	+
Сообщение	+	+

	Windows	UNIX
Разделяемая память	+	+
Файлы	+	+
Отображаемые файлы	+	+
Каналы	+	+
Почтовые ячейки	+	+
Сокеты	+	+

# Ввод-вывод

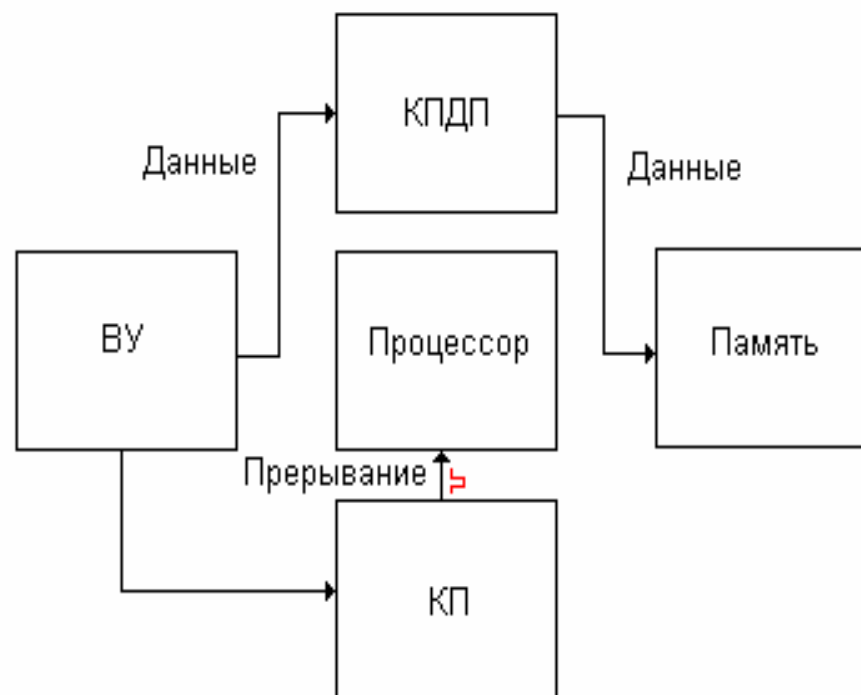
## Назначение:

- Управление внешними устройствами
- Передача данных на внешние устройства
- Прием данных со внешних устройств

### А) Режим PIO

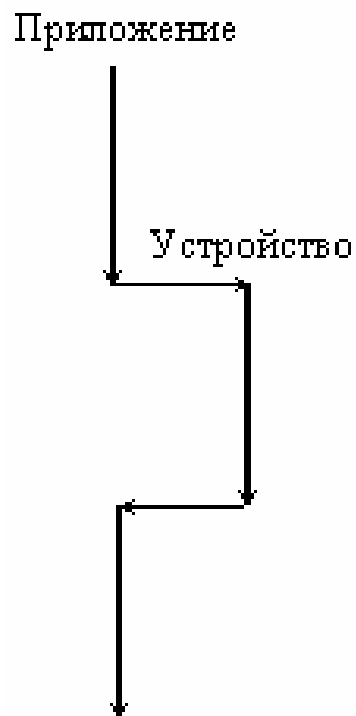


### Б) Режим DIO

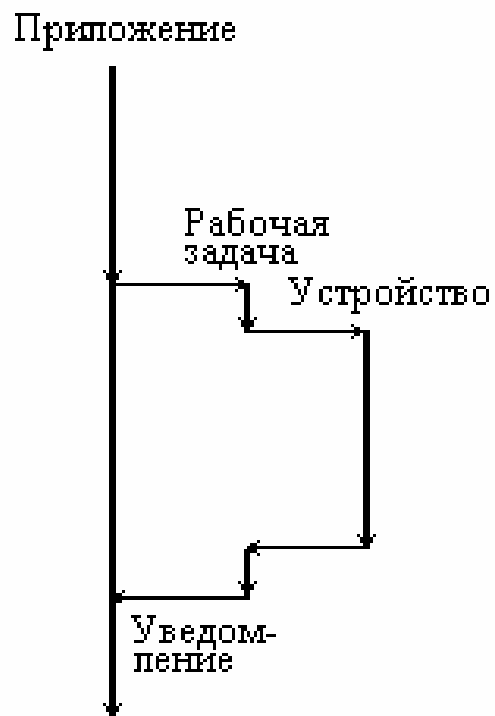


## Ввод-вывод

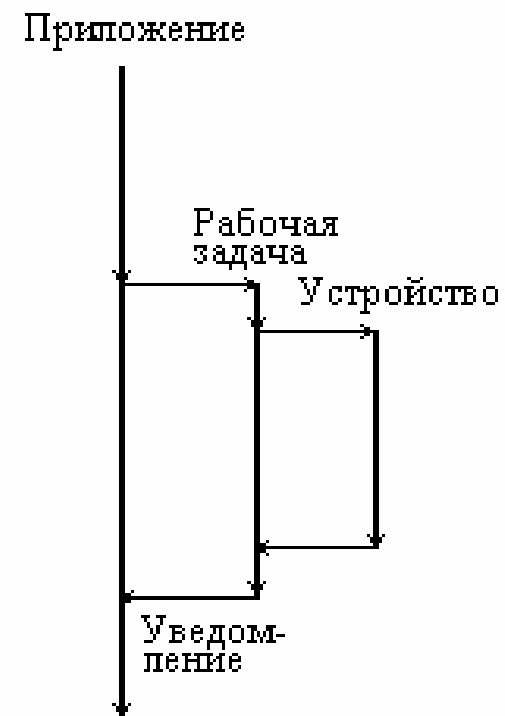
А) Синхронный  
ввод-вывод



Б) Асинхронный  
ввод-вывод  
с ожиданием  
завершения



В) Асинхронный  
ввод-вывод с  
сигналом завершения



Понятия «синхронный» и «асинхронный» везде!

# POSIX



ARINC 653

POSIX.5 – Интерфейс языка ADA

POSIX.9 – Интерфейс языка FORTRAN77

POSIX.16 – Интерфейс языка C

POSIX.19 – Интерфейс языка FORTRAN90

POSIX.20 - Интерфейс языка ADA для PB

POSIX.4 – требования PB: RR + RM или EDF; 2)

разделяемая память с блокировкой; 3) семафоры; 4)

сигналы; 5) очереди сообщений; 6) часы 20 мс; 7)

асинхронный в/в.

POSIX.4a: потоки.

POSIX.4b: тайм-ауты для системных вызовов; часы процессорного времени; спорадический сервер.

POSIX.13 – профили PB:

1. Минимальная система: 1 процесс + потоки;

2. Контроллер PB: (1) + файловая система и в/в;

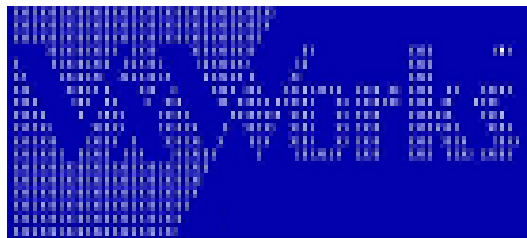
3. Специализированная система: (4) без файловой системы.

4. Многоцелевая система – все вместе.

# ОС РВ

Предоставляют прикладным программам набор возможностей для организации РВ. Основные отличительные свойства:

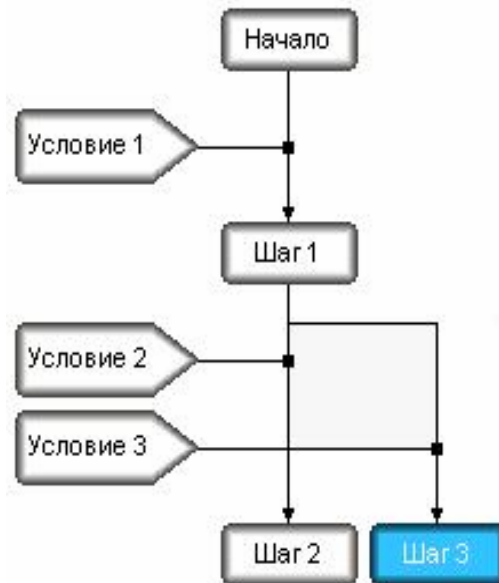
- предсказуемость поведения во времени;
- модульность и масштабируемость.



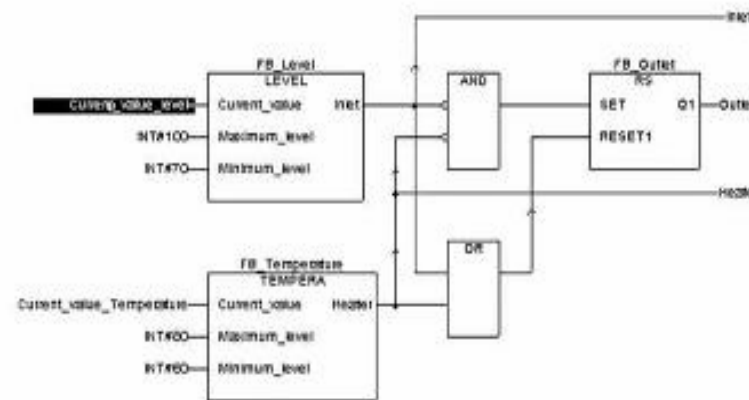


# Языки IEC-61131

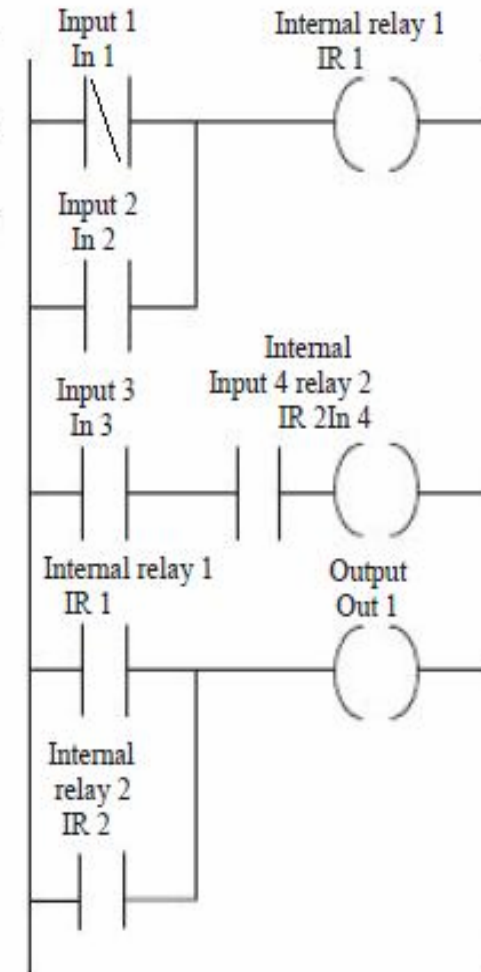
SEQUENTIAL FUNCTIONAL CHART



FUNCTION BLOCK-DIAGRAM



LADDER DIAGRAM



INSTRUCTION LIST

```

PROGRAM
  VAR_INPUT ARG_000 : REAL; END_VAR
  VAR_INOUT ARG_001 : REAL; END_VAR

  WHILE ARG_000 > 2 DO ARG_000 = ARG_000-1;
  ARG_001 = ARG_001 + 2;
  END_WHILE;

END_PROGRAM
  
```

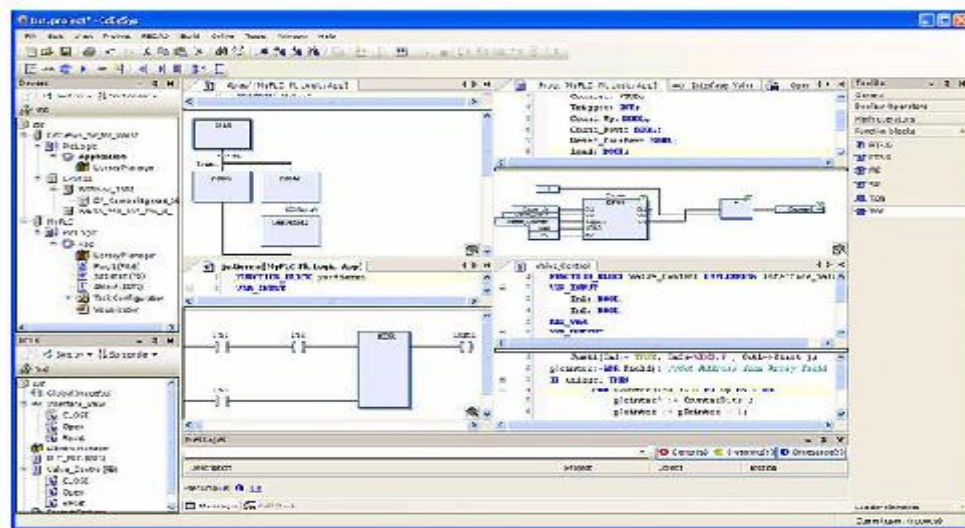
STRUCTURED TEXT

```

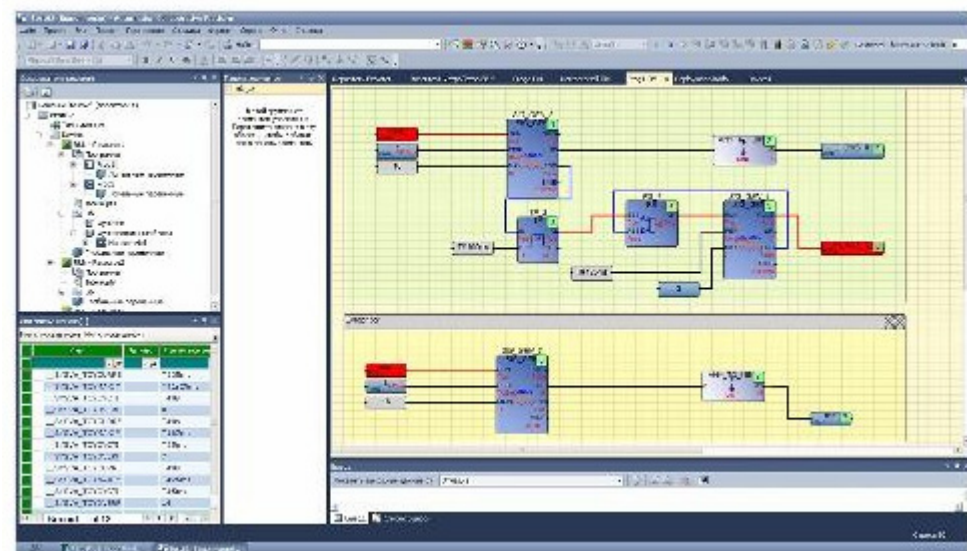
ADD VAR_000 2.6
LT VAR_000 VAR_001
JMPC label1
GT VAR_001 20
JMPC label2
LD 278
label1:
CAL FUNCTION1(VAR_000, 3)
label2: ST VAR_001
  
```

# Среды программирования ПЛК

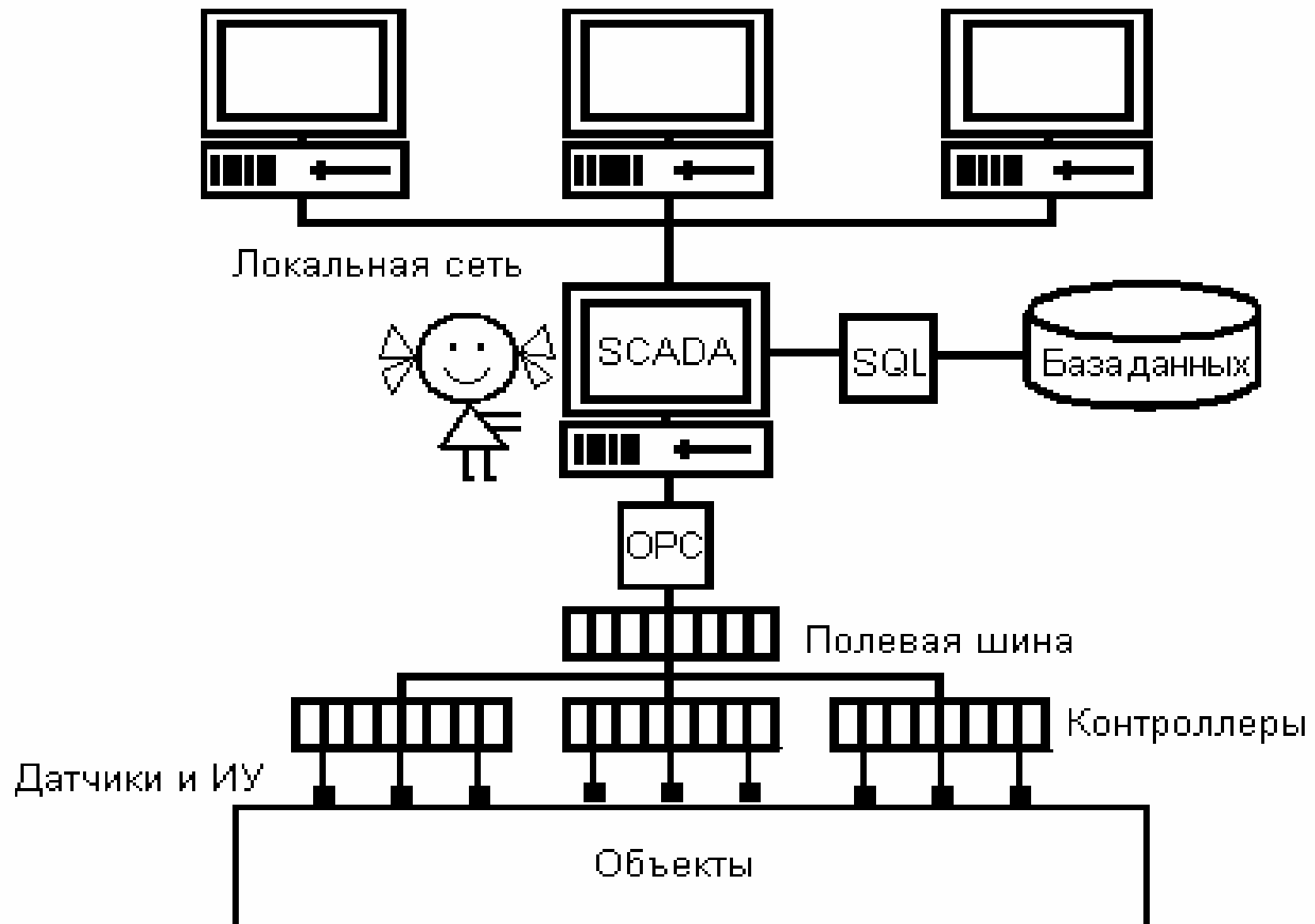
## CODESYS



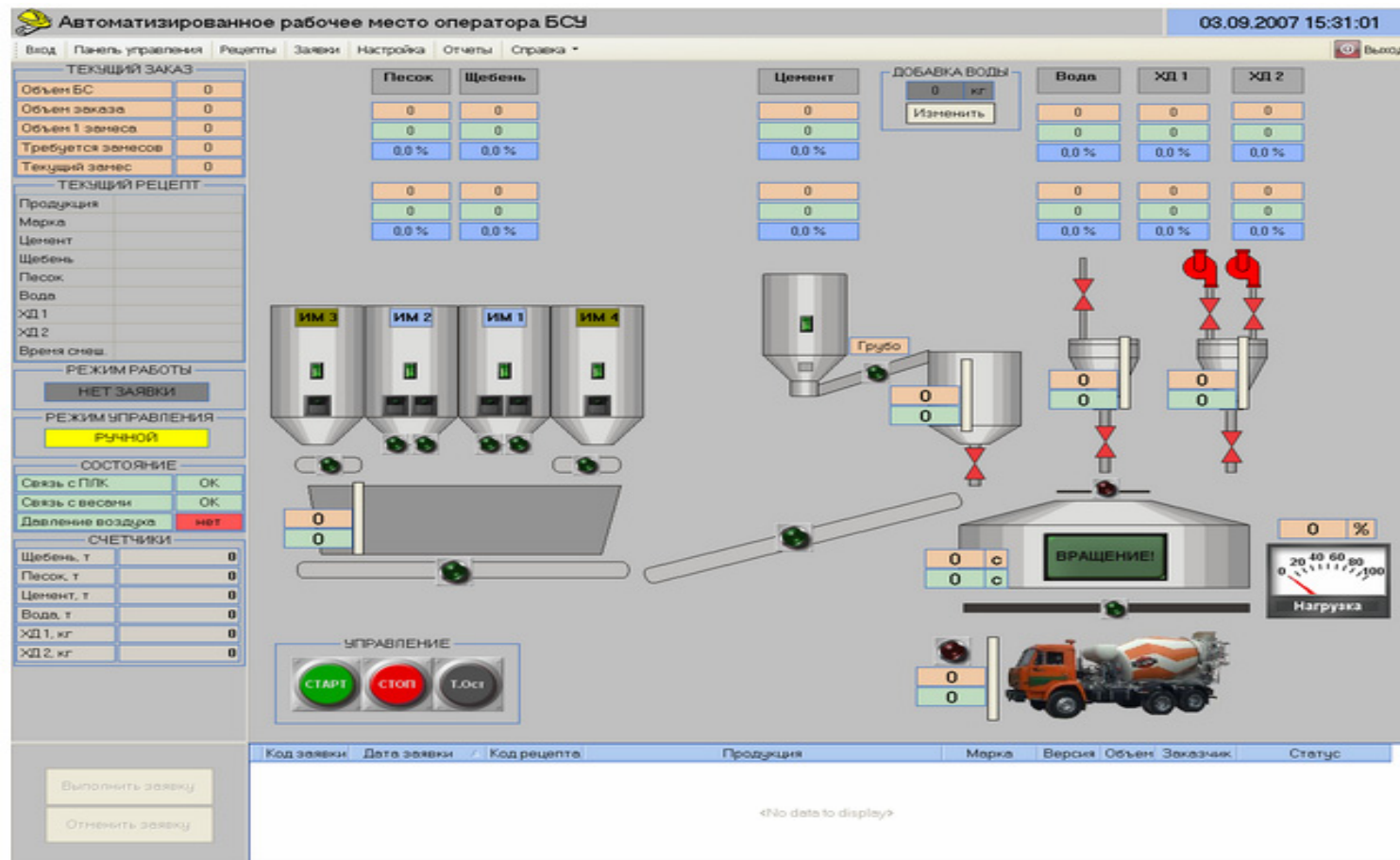
## ISAGRAF



# Структура АСУ ТП



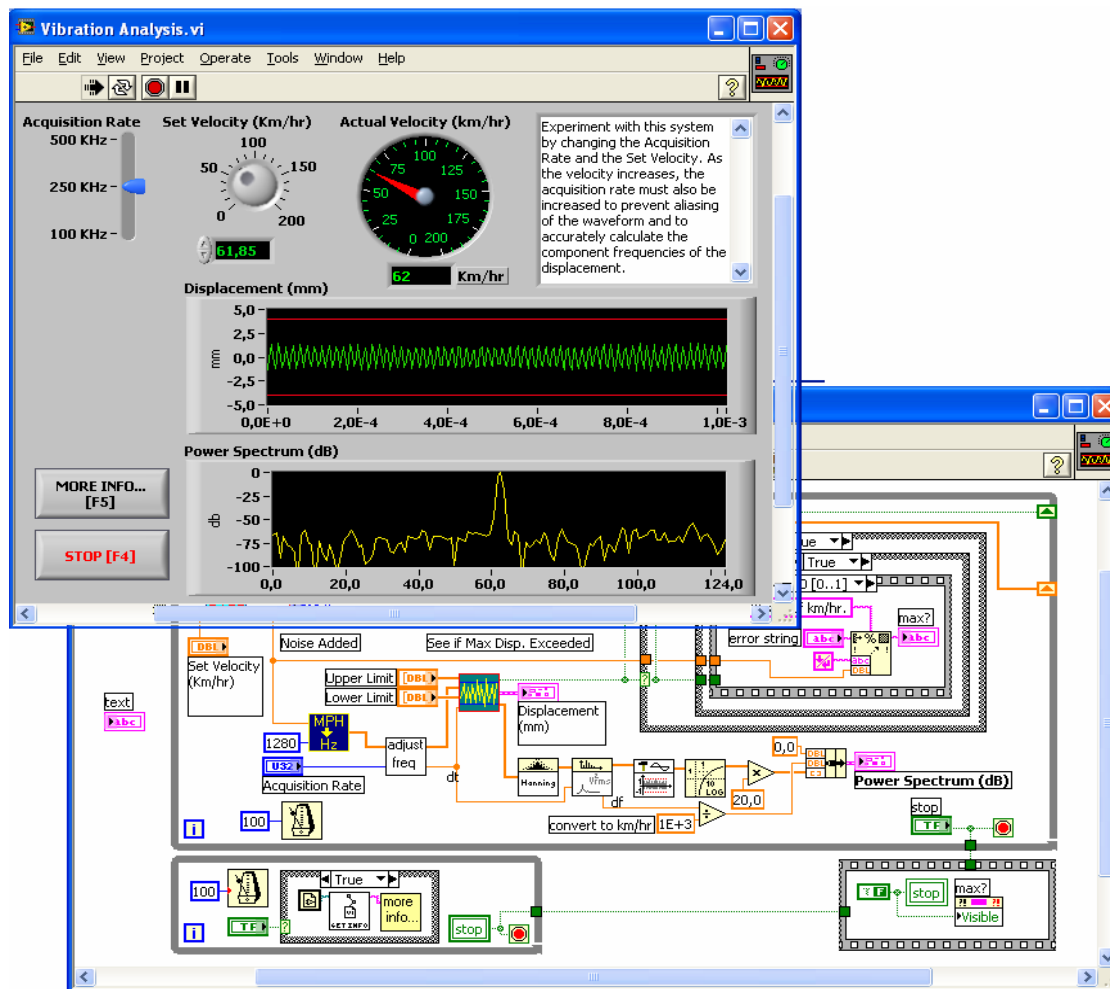
# SCADA-системы



**Термины:** HMI, мнемосхемы, тренды, уставки, алармы, скрипты.

**Примеры:** InTouch, TraceMode, WinCC, Citect, SimpleSCADA

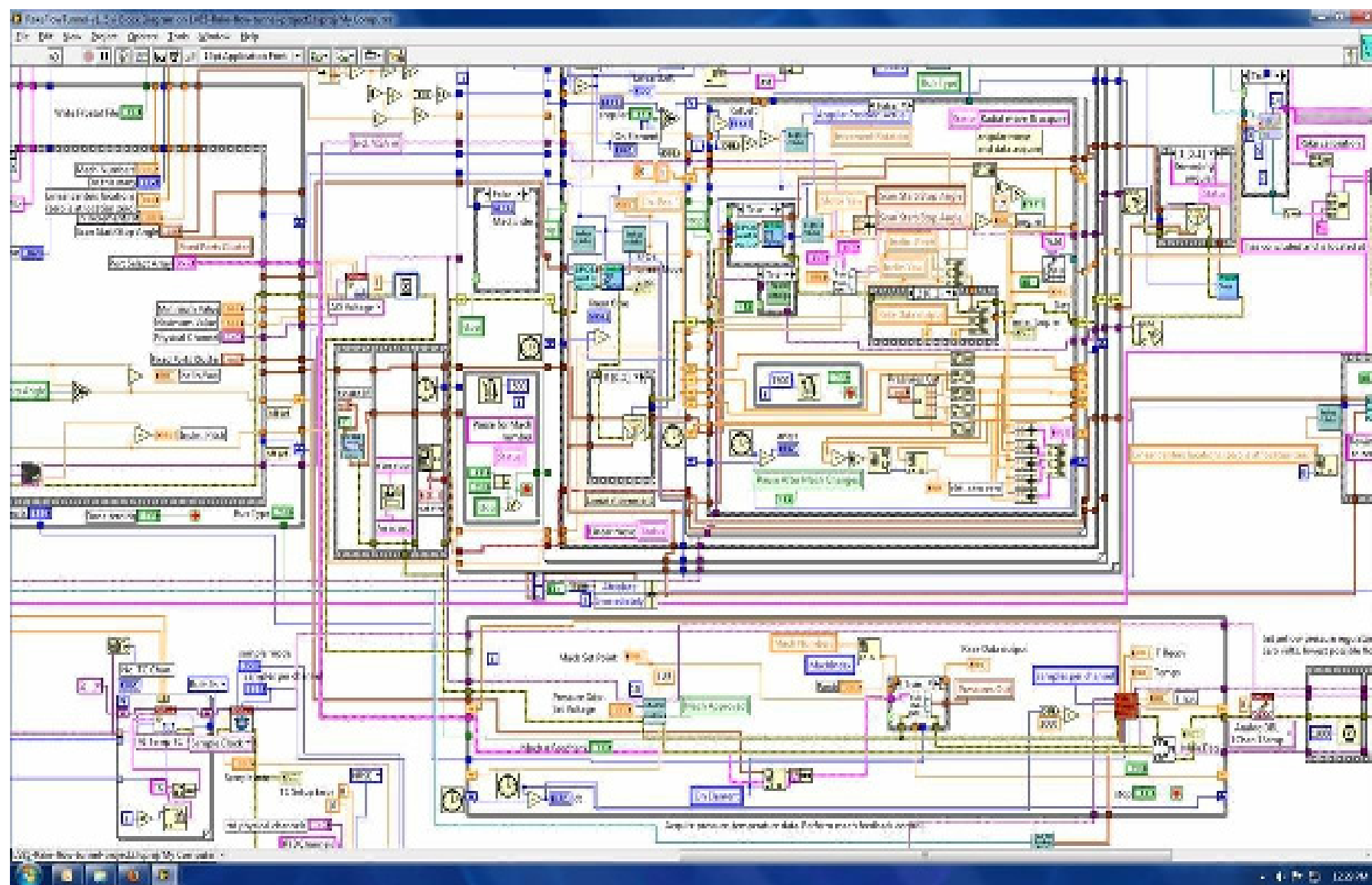
# Виртуальные приборы



**Термины:** HMI, виртуальный прибор, лицевая панель, блок-диаграмма.

**Примеры:** LabVIEW, DasyLab.

# Пример «правильной» блок-диаграммы ☺



# Языки РВ

```
// JAVA + RTSJ - в потоке каждые 100 мс измеряет текущее время
import javax.realtime.ImmortalMemory;
import javax.realtime.PeriodicParameters;
import javax.realtime.RealtimeThread;
import javax.realtime.RelativeTime;
public class HelloWorld extends RealtimeThread {
    //allocate long array from immortal memory
    public static long[] times = (long[]) ImmortalMemory.instance().newArray(long.class, 100);
    public HelloWorld(PeriodicParameters pp) {
        super(null, pp);
    }
    public void run() {
        for (int i = 0; i < 100; i++) {
            times[i] = System.currentTimeMillis();
            waitForNextPeriod(); //wait for next period
        }
    }
    public static void main(String[] argv) {
        //schedule real time thread at every 100 milisecond
        PeriodicParameters pp = new PeriodicParameters(new RelativeTime(100, 0));
        HelloWorld rtt = new HelloWorld(pp);
        rtt.start();
        //wait real time thread to terminate
        try {
            rtt.join();
        } catch (InterruptedException ex) {
            ex.printStackTrace();
        }
        //print results
        for (int i = 0; i < 100; i++) {
            System.out.println(times[i]);
        }
    }
}
```

# Языки РВ (окончание)

```
// ADA - в потоке каждые 100 мс измеряет текущее время
package body Example is
  T0 : constant Time := Clock;
  -- Gets set at elaboration time

  Task1_Priority : constant System.Priority := 3;
  Task1_Period : constant Time_Span := To_Time_Span (0.010);
  Task1_Offset : constant Time_Span := To_Time_Span (0.003);

  task Task1 is
    pragma Priority (Task1_Priority);
  end Task1;

  task body Task1 is
    Next_Time : Time := T0 + Task1_Offset;
  begin
    loop
      -- Do something

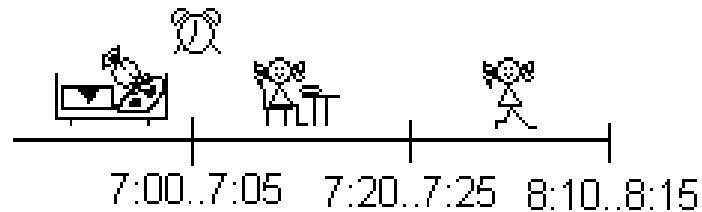
      Next_Time := Next_Time + Task1_Period;
      delay until Next_Time;
    end loop;
  end Task1;
end Example;
```



# Управление

## А) Императивное

**Циклограмма** = расписание,  
т.е.  $\langle \{A_i\}, \{T_i\}, \{A_i, T_i\} \rangle$ .



Период генератора тактовой частоты:

$$\Delta \tau = \frac{\text{НОД}(C \cdot T_1, C \cdot T_2, \dots, C \cdot T_N)}{C}$$

## б) Интерактивное

**Продукция** = правило вида  
«если условие то действие».

**Автомат** =

$\langle \{S\}, \{E\}, \{A\}, \{S \times E \rightarrow (A, S)\} \rangle$

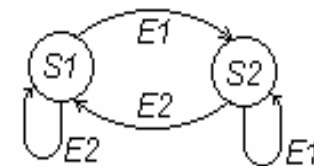


S1 – поезд едет; S2 – поезд стоит.

E1 – «красный»; E2 – «синий».

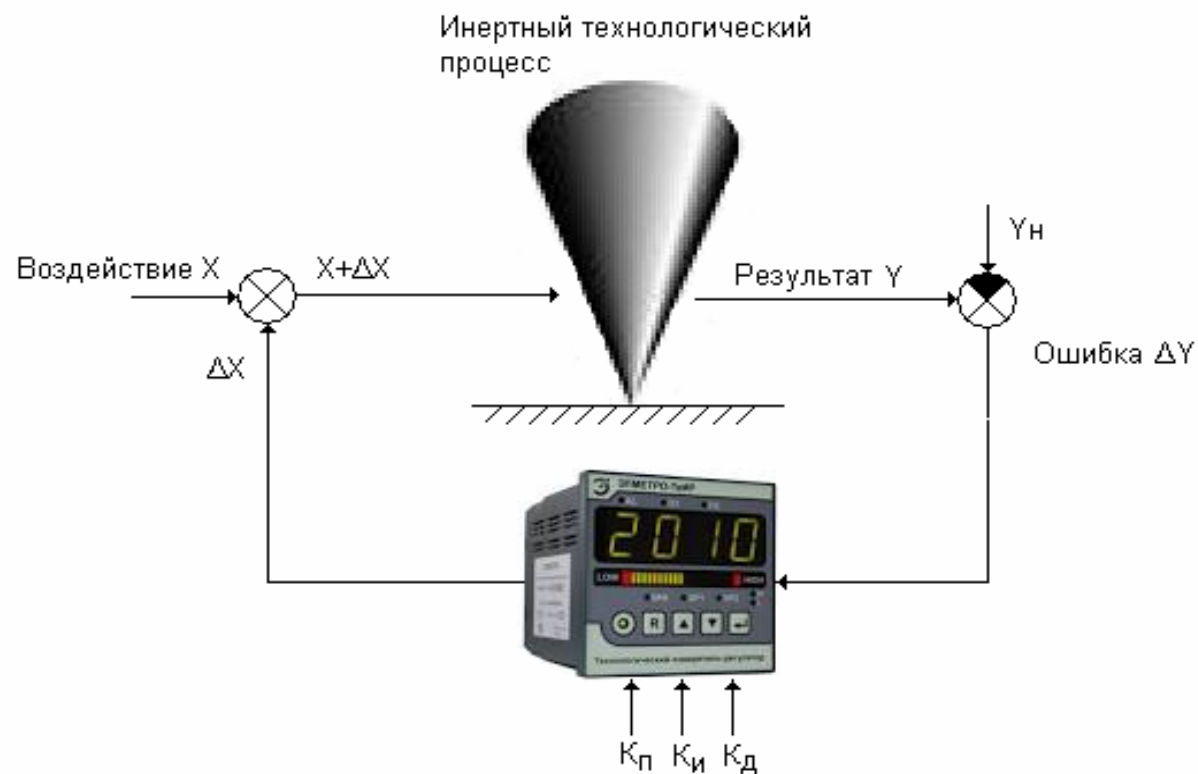
A1 – тормозить; A2 – разгоняться; A3 – сохранять

	E1	E2
S1	A1, S2	A3, S1
S2	A3, S2	A2, S1



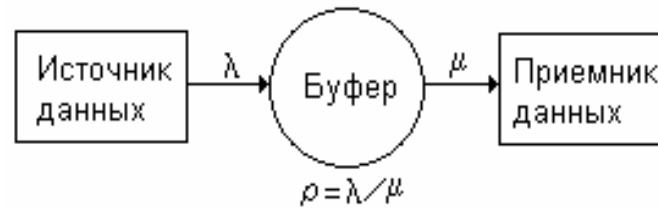
## Управление (продолжение)

### в) Адаптивное

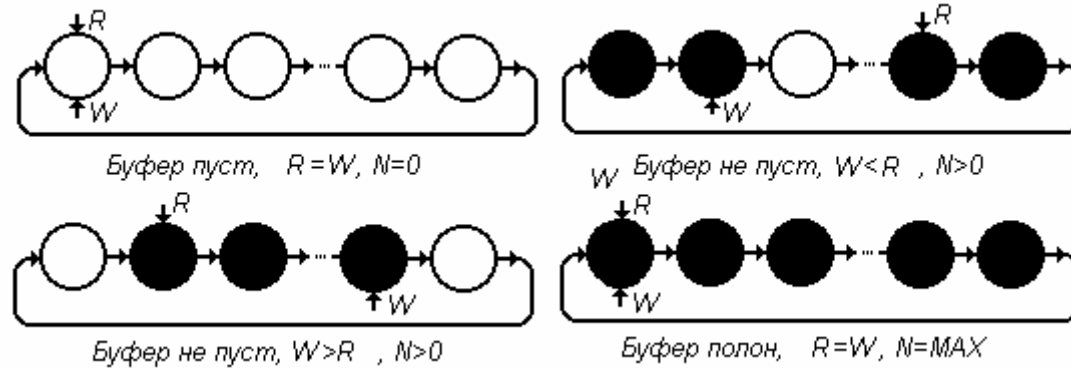


$$\Delta X = \Delta X_p + \Delta X_i + \Delta X_d = K_p \cdot \Delta Y + K_i \cdot \int \Delta Y dt + K_d \cdot \frac{d\Delta Y}{dt}$$

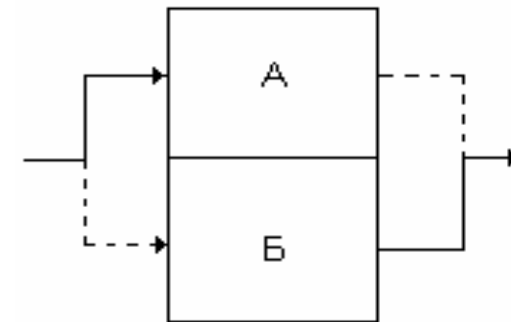
# Сбор данных. Буферизация



## А) Кольцевой буфер



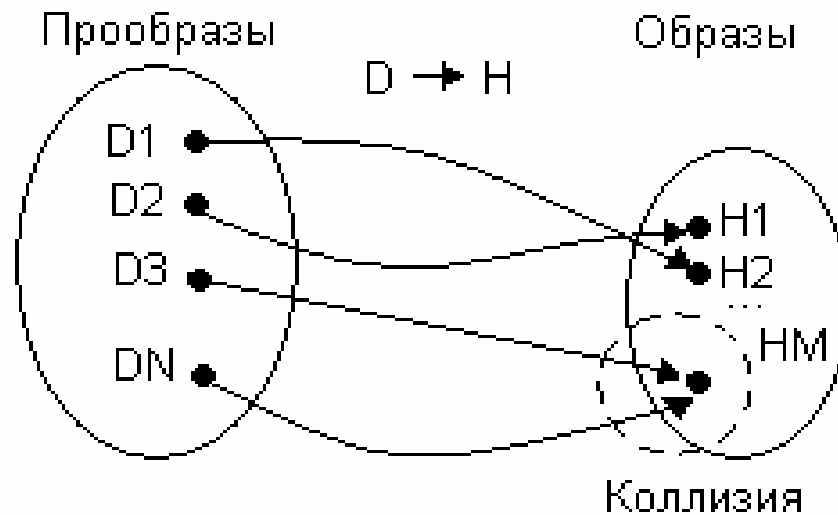
## Б) Двухтактный (двойной) буфер



Вероятность переполнения буфера размером  $m$ :

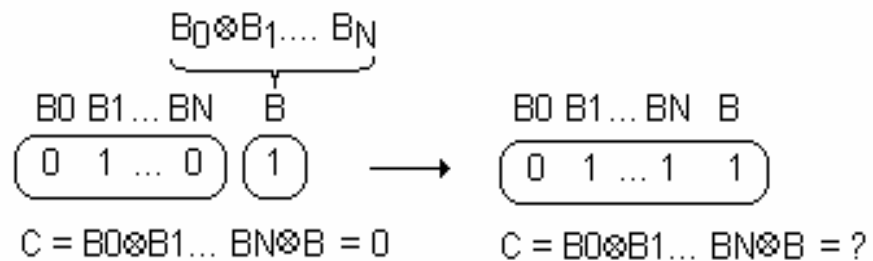
$$P_m = \rho^{m+1} \frac{1 - \rho}{1 - \rho^{m+2}} = \frac{\rho^{m+1}}{\sum_{i=0}^{m+1} \rho^i}.$$

## Сбор данных. Контроль целостности



Критерий качества –  
равномерность распределения.

### Бит четности (паритета)



# Контроль целостности (продолжение)

## Простые контрольные суммы

Арифметическое суммирование:

```
unsigned char csum=0;  
for (i=0;i<N;i++) csum = csum + s[i];
```

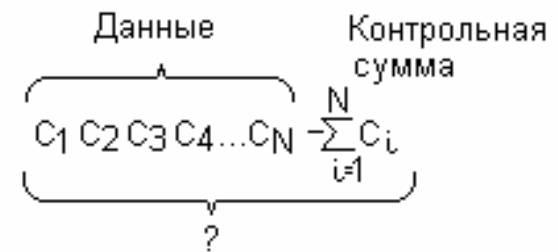
Суммирование по mod2:

```
unsigned char csum=0;  
for (i=0;i<N;i++) csum = csum ^ s[i];
```

Алгоритм Бернштейна:

```
unsigned char csum=0;  
for (i=0;i<N;i++) csum = csum*31 + s[i];
```

Вариант применения:



# Контроль целостности (окончание)

## CRC – Циклические избыточные коды

### 1. Двоичные полиномы

$$10110001 \sim 1 \cdot x^7 + 0 \cdot x^6 + 1 \cdot x^5 + 1 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$$

### 2. Деление полиномов с остатком

$$\underline{x^4 + x^2} / x^2 + x + 1$$

Порождающий полином

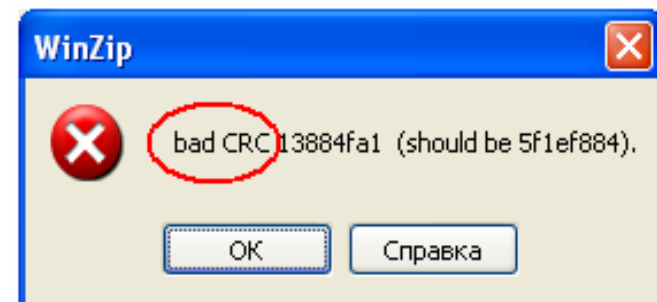
$$\begin{array}{r} \oplus 10100 \overline{) 111} \\ \underline{111} \phantom{00} \\ \oplus 100 \phantom{00} \\ \underline{111} \phantom{00} \\ \oplus 110 \phantom{00} \\ \underline{111} \phantom{00} \\ 1 \leftarrow \end{array}$$

### 3. Стандартные порождающие полиномы

CRC-16 1EDB88320<sub>16</sub>

CRC-32 1A001<sub>16</sub>

### 4. Пример



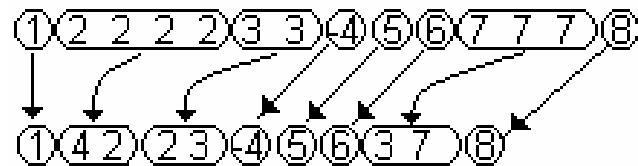
## Сбор данных. Сжатие

Сжатие = уменьшение объема данных

- Без потерь информации
- С частичными потерями (наименее значимой) информации.

$$K_{\text{сж}} = N_{\text{до}} / N_{\text{после}}$$

### А) RLE – метод группового кодирования

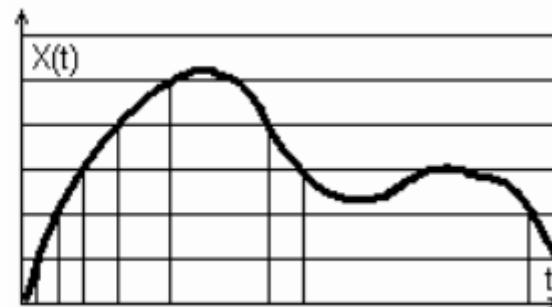
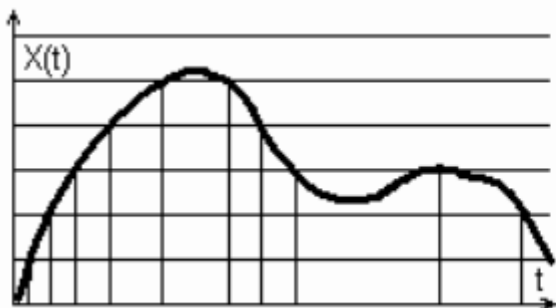


Служебный бит: 1 -4 2 -2 3 -1 -4 5 6 -3 7 8

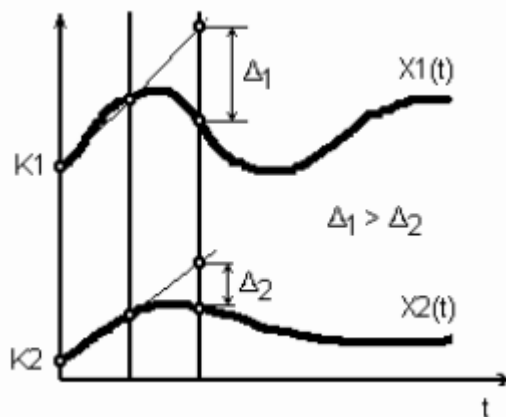
Служебный байт: 01100010=0x62 1 4 2 2 3 -4 5 6 3 7 8

## Сбор данных. Сжатие (окончание)

### Б) Методы с адаптивной дискретизацией

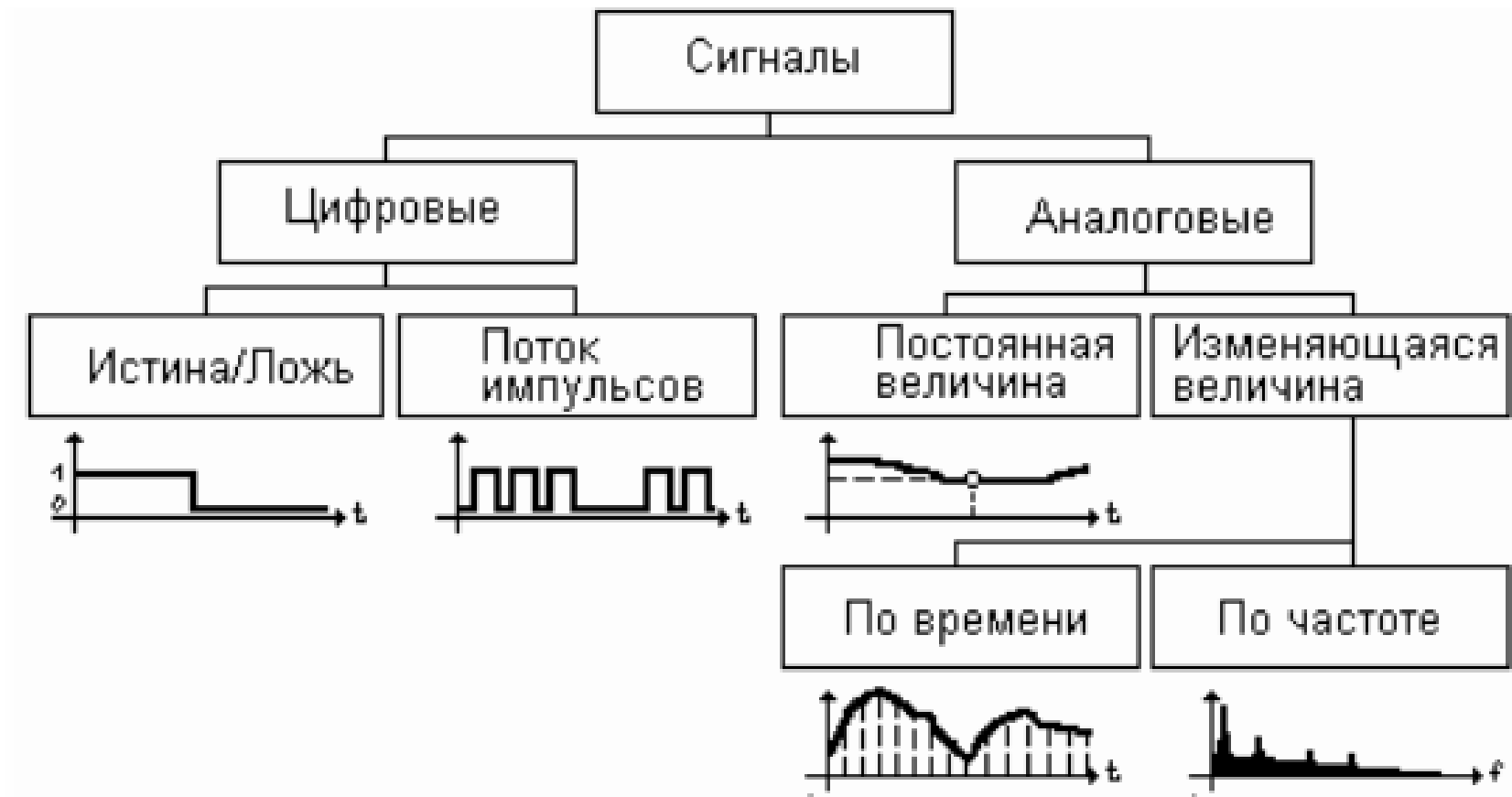


### В) Методы с адаптивной коммутацией





# Обработка данных



# Измерение постоянной величины

**Величина:**  $X$ .

**Измерения многократные:**  $N$  наблюдений  $X_1, X_2 \dots X_N$ .

**Математическая модель:** нормально распределенная случайная величина.

**Результат оформляется в виде:**

- Оценка величины  $X$ ;
- Допустимые границы отклонения  $\Delta$ ;
- Доверительная вероятность оценки:  $P_{\text{дов}}$ .

$$X = \frac{1}{N} \sum_{i=1}^N X_i ,$$

$$\Delta X_{\text{изм}} = K \cdot \tilde{\sigma}_{\text{изм}},$$

$$\tilde{\sigma}_{\text{изм}} = \sqrt{S^2 + \theta^2} ,$$

$$S^2 = \frac{1}{N-1} \sum_{i=1}^N (X_i - \tilde{X})^2$$

$\theta^2$  = с.к.о. предельной погрешности средства измерения.

$K$  соответствует  $P_{\text{дов}}$ , например при  $P_{\text{дов}}=0.95$   $K=1.96$ , из таблицы  $K$ -тов Стьюдента.

## Измерение вектора

**Вектор:**

$$\vec{X} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \pm \Delta_1 \\ x_2 \pm \Delta_2 \\ \dots \\ x_N \pm \Delta_N \end{pmatrix}$$

**Корреляционная матрица:**

$$\begin{pmatrix} 1 & r_{1,2} & r_{1,3} & r_{1,4} & \vdots & r_{1,N} \\ r_{2,1} & 1 & r_{2,3} & r_{2,4} & \vdots & r_{2,N} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ r_{N,1} & r_{N,2} & r_{N,3} & r_{N,4} & \vdots & 1 \end{pmatrix}$$

**Коэффициент корреляции:**

$$r_{x,y} = \frac{1}{\sigma_x \sigma_y} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

# Оптимизация циклических алгоритмов

## Обозначения:

$\omega$  – числовая последовательность;  $F(\omega)$  – некоторая функция;  
 $x$  – новый член последовательности;  $*$  - операция добавления к  $\omega$ .

## Определения:

$F$  – индуктивна, если существует  $G : F(\omega * x) = G(F(\omega), x)$ .

Пример:  $F$  = сумма членов  $\omega$ , тогда  $F(\omega * x) = F(\omega) + x$ .

$H$  – индуктивное расширение  $F$ , если  $F(\omega * x) = G(F(\omega), H(\omega))$ .

Пример:  $F$  = количество максимумов в  $\omega$ , тогда  $H(\omega) = \max(\omega)$ .

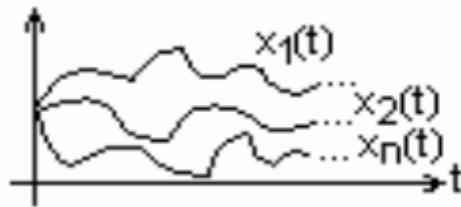
```
(* Традиционный метод *)
mx := 0; n := 0;
while not конец do begin
  mx := mx + Новое_значение;
  n := n + 1;
end;
mx := mx / n;
```

```
(* «Индуктивный» метод *)
mx := 0; n := 0;
while not конец do begin
  x := Новое_значение;
  mx := (mx * n + x) / (n + 1);
  n := n + 1;
end;
```

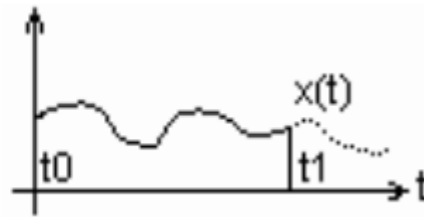
# Измерение случайных процессов

**Случайный процесс** = случайная функция неслучайного аргумента

Случайный процесс

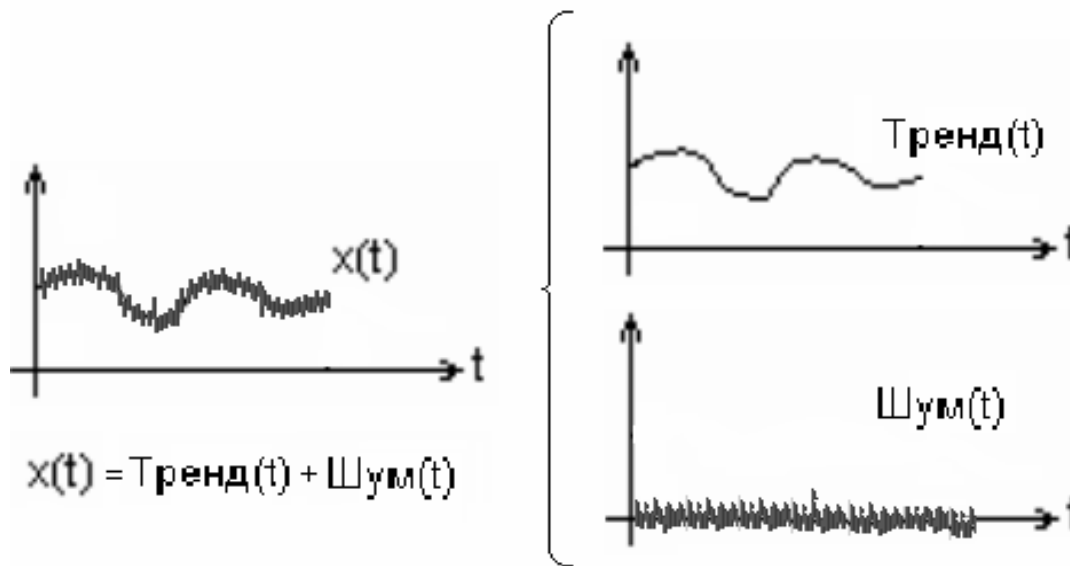


Единственная реализация  
конечной длины



- 1) Стационарность
- 2) Эргодичность

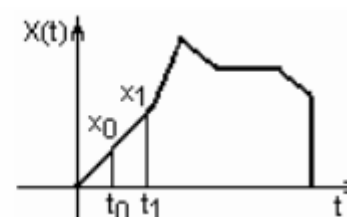
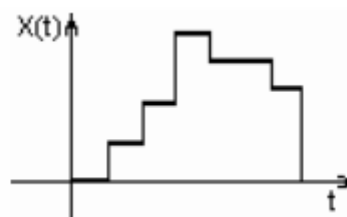
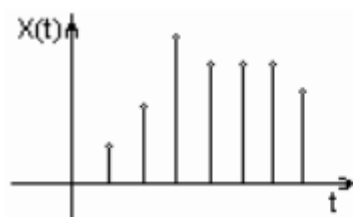
**Разделение на тренд и шум**



**Сглаживание:**

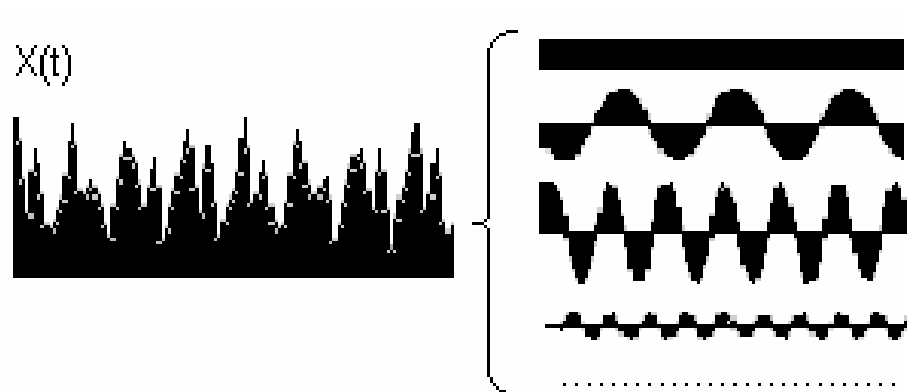
- 1) Скользящее среднее  
 $X'_i = (x_{i-1} + x_i + x_{i+1})/3$ .
- 2) Медианный фильтр  
 $X'_i = \text{med}(x_{i-1} + x_i + x_{i+1})$ .

## Восстановление сигнала



$$x(t) = k \cdot t + L, \text{ где } k = (x_1 - x_0) / (t_1 - t_0), L = (t_1 x_0 - x_1 t_0) / (t_1 - t_0)$$

## Спектр сигнала



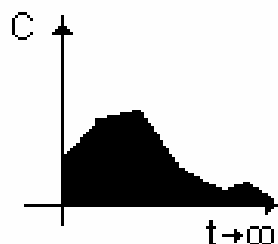
$$y_v = A_0 + \sum_{m=1}^N \left[ A_m \cos \frac{2\pi m v}{2N+1} + B_m \sin \frac{2\pi m v}{2N+1} \right], \text{ где}$$

$$1) A_0 = \frac{1}{2N+1} \sum_{v=-N}^N y_v;$$

$$2) A_m = \frac{2}{2N+1} \sum_{v=-N}^N y_v \cos \frac{2\pi m v}{2N+1};$$

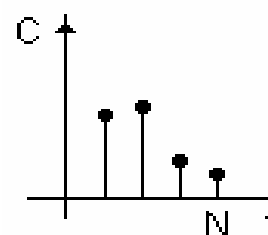
$$3) B_m = \frac{2}{2N+1} \sum_{v=-N}^N y_v \sin \frac{2\pi m v}{2N+1}.$$

### Непрерывный спектр



### Спектральная плотность

$$C_m = \sqrt{A_m^2 + B_m^2}$$



# Быстрое преобразование Фурье

Пусть  $\omega = \frac{2\pi\nu}{2T+1}$ ,

$$\text{тогда } y_\nu = y(\nu) = A_0 + \sum_{m=1}^N [A_m \cos(m\omega\nu) + B_m \sin(m\omega\nu)] =$$

$$= y(\nu) = A_0 + \sum_{m=1}^N [A_m \cos(m\omega\nu + \varphi_m)] =$$

$$\text{Формула Эйлера: } e^{jx} = \cos x + j \sin x \Leftrightarrow \cos x = \frac{1}{2}(e^{jx} + e^{-jx})$$

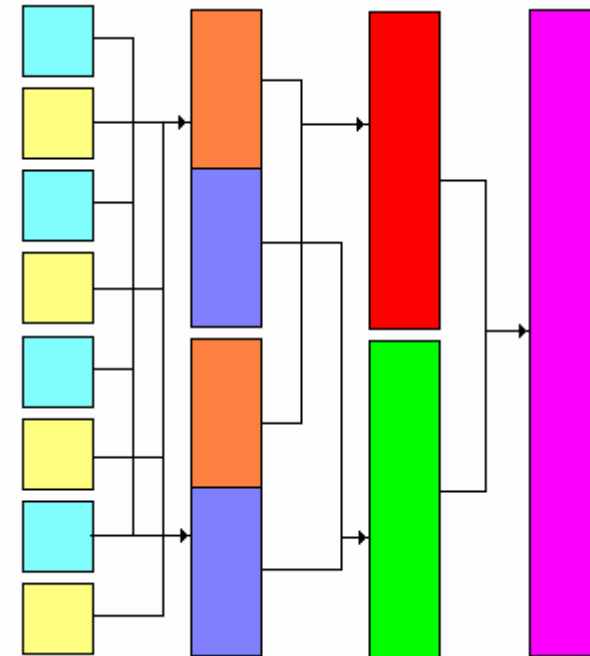
$$= y(\nu) = A_0 + \sum_{m=0}^N \frac{A_m}{2} [e^{jm\omega\nu + j\varphi_m} + e^{-jm\omega\nu - j\varphi_m}] =$$

$$= y(\nu) = A_0 + \sum_{m=0}^N C_m e^{-jm\omega\nu}.$$

В исходной форме  $N \times N = N^2$  умножений.

В БПФ  $N + N/2 + N/4 + \dots + 1$  умножений.

Выигрыш:  $N/\log_2(N)$  раз.

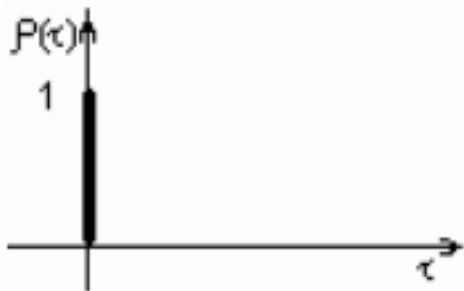


## Корреляционные функции сигнала

**ВКФ:** 
$$R(x, y) = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y}$$

**АКФ:** 
$$\rho(k) = \frac{1}{\sigma_x^2 (N - k)} \sum_{t=1}^{N-k} x_t x_{t+k}$$
 для всевозможных  $0 < k < N$

АКФ белого шума



АКФ реального сигнала



Пусть  $S(t)$  – спектр сигнала, тогда  $\rho(t) = |S(t)|^2$  = «энергетический» спектр.



**MISSION**

**ACCOMPLISHED**