

Петрозаводский Государственный Университет  
Институт математики и информационных технологий  
Кафедра Информатики и математического  
обеспечения

## Все об информационных ресурсах Института математики и информационных технологий

Заведующий кафедрой Информатики и Математического  
обеспечения, к. т. н., доцент,

Ю. А. Богоявленский  
ybgv@cs.karelia.ru  
<http://cs.karelia.ru>



# Содержание

- 1 Введение
- 2 Математика и информатика
  - Математики и архитектура ЭВМ
  - Математики — лауреаты премии Тьюринга
  - Выводы
- 3 Профессиональные задачи.
- 4 Математическая культура
  - Разделы математики и ИКТ
- 5 Организация учебной работы
- 6 Современные учебные планы
- 7 ИВИ ИМИТ
- 8 ОС UNIX → Linux
- 9 Язык C — на все времена.
  - Язык C — lingua franca
  - Язык C — libc
  - Характеристики glibc
- 10 CLI vs GUI
  - Командный интерфейс: конвейер команд
  - Командный интерфейс: сложная работа с файлами
- 11 От учебы к профессиональной карьере



# Введение. Решение задач.

Новый важный этап интеллектуального развития.

**Цель студента** — сформировать культуру и навыки решения профессиональных **задач**.

- **Мотивация.** Очень хотеть решить задачу.
- **Знания.** Владеть объектами, их свойствами и связями предметной области.
- **Анализ.** Уметь выполнять исчерпывающий анализ условий и требований постановки задачи.
- **Синтез.** Строить путь от исходных данных к решению задачи.



# Введение. Математика и информатика

**Математика** — формирует обобщенную культуру решения задач.

**Крайне полезный ресурс** — **Математическая энциклопедия**

**Центральное понятие** — **Алгоритм**

«Dixit Algorizmi» (Сказал Аль-Хорезми) — из переводов на латынь («De numero Indorum» (Индийский счет), XII век) трудов IX века математика Мухаммеда ибн Муса аль-Хорезми.

1936 г. Теория (стандартизация) алгоритмов. Математики:

- **Алан Тьюринг (Alan Turing)** Машина Тьюринга;
- **Эмиль Пост (Post Emil)** Машина Поста;
- **Алонзо Чёрч (Alonzo Church)**  $\lambda$ -исчисление;
- **Андрей Андреевич Марков (младший)** Нормальные алгоритмы Маркова.

Черная овца в Австралии.

**Очень полезная книга.** Д. Пойа Как решать задачу



# Математики и архитектура ЭВМ

**Джон фон Нейман (John von Neumann, Янош Лайош Нейман)** — аксиоматизации теории множеств и квантовой механики, алгебры фон Неймана, клеточные автоматы, **расчёты сжатия плутониевого заряда до критической массы.**

В работе Burks A. W., Goldstine H. H., Neumann J. Preliminary Discussion of the Logical Design of an Electronic Computing Instrument. —Institute for Advanced Study, Princeton, N. J., July 1946 сформулированы основные принципы логической структура ЭВМ, названные позже **«Архитектура фон Неймана».**

В 1946 г. он возглавил проект по созданию ЭВМ IAS с этой архитектурой

Подавляющее большинство современных ЭВМ реализованы согласно **Архитектуре фон Неймана.**



# 1959 г. ЭВМ Урал-2



# 1959 г. ЭВМ Урал-2. Характеристики

- 5000 целочисленных операций/сек;
- Оперативная память 4096 20-разрядных слов на магнитных сердечниках, возможен доступ к 40-разрядным словам.
- Ввод данных и команд с перфорированной киноленты.
- Вывод на печатающее устройство со скоростью до 20 чисел/с.
- Количество ламп 2100.
- Потребляемая мощность 25 кВт.
- Занимаемая площадь 150—200 м<sup>2</sup>



# Математики — лауреаты премии Тьюринга

Премия Тьюринга (**Turing Award**) — самая престижная премия в информатике, вручаемая с 1966 г. Ассоциацией вычислительной техники (**Association for Computing Machinery ACM**)

- 1977. Джон Бэкус** — руководитель разработки языка **Фортран**, изобретатель формы **Бэкуса — Наура** для определения синтаксиса формальных языков.
- 1983. Деннис Ритчи** — один из авторов языка **C**, соавтор учебника **K&R**, участник разработки **ОС Unix**.
- 1988. Ван Якобсон** — разработал 4 алгоритма работы проткола TCP (в том числе **AIMD**), позволившие предотвращать коллапсы перегрузки Интернет и перейти к ее широкому внедрению.  
См. **M. Allman, V. Paxson, E. Blanton Request for Comments: 5681 TCP Congestion Control**





# Математики — лауреаты премии Тьюринга

- 1999.** **Фредерик Брукс (младший)** — автор бестселлера “**Мифический человеко-месяц**” и “**закона Брукса**”: *Если проект не укладывается в сроки, то добавление рабочей силы задержит его еще больше.* Возглавлял разработки семейства мейнфреймов **IBM System/360** (теперь – IBM zSeries) и их операционной системы **OS/360**.
- 2002.** **Рональд Ривест, Ади Шамир и Леонард Адлеман** — авторы системы шифрования (**RSA Rivest — Shamir — Adleman**, 1977 год), которая широко применяется, в том числе в защищенном веб пртоколе **https**.
- 2004.** **Винтон Серф** — вместе с Робертом Каном под руководством **Леонарда Клейнрока** в 1974 г. разработал семейство протоколов **TCP/IP**.



# Математик — VK (100 млн.) и Telegram (950 млн.)

Николай Валерьевич Дуров. Год рождения 1980.

Сооснователь социальной сети VK и Telegram Messenger.

- Три золотые медали на Чемпионате мира по математике среди школьников в 1996, 1997 и 1998 годах.
- Три серебряные и золотая медали на Международной олимпиаде по информатике для школьников с 1995 по 1998 год.
- Член команды СПбГУ победившей в Международный олимпиаде ACM ICPC в 2000 и 2001 годах.
- Выпускник Математико-механического факультета СПбГУ.
- В 2005 г. под руководством С. В. Востокова защитил диссертацию: “Метод вычисления группы галуа многочлена с рациональными коэффициентами”.
- Работал ведущим разработчиком в команде VK до 2013 года.
- Автор MTProto — криптографического протокола Telegram Messenger на основе симметричного алгоритма шифрования AES с 2048-битными RSA-ключами.



# Выводы

ЭВМ **по сути** есть математические машины так как:

- 1** идейно опираются на абстрактные математические машины (Тьюринг, Пост), разработанные для формализации понятия алгоритма и положившие начало развитию математической теории алгоритмов;
- 2** предназначены для ускорения выполнения самых разных алгоритмов, строящихся на основе численных методов математики;
- 3** создавались под идеейным руководством математиков согласно разработанной ими логической архитектуре (фон Нейман).

Основные достижения по развитию современных ИКТ получены математиками или учеными, владеющими математической культурой.



# Профессиональные задачи.

## ■ Алгоритмы

- ▶ Знать основные классы. Владеть основами оценки сложности.
- ▶ Выбирать, комбинировать, модифицировать, разрабатывать.

## ■ Структуры данных

- ▶ Знать базовые структуры и их свойства.
- ▶ Понимать алгоритмы работы с элементами и их временную сложность.
- ▶ Выбирать, комбинировать, применять.

## ■ Компьютерные сети

- ▶ Знать уровни абстракции протоколов.
- ▶ Разрабатывать сетевые системы.

Компетенции ИКТ = Математическая + Инженерная (технология разработки ПО)



# Профессиональные задачи.

## ■ Web технологии

- ▶ Знать протокол `http`, языки `html`, `css`, `xml`.
- ▶ Владеть навыками разработки на языках `PHP`, `JavaScript`.
- ▶ Применять программные каркасы (frameworks).

## ■ Технология разработки программного обеспечения

- ▶ Знать модели жизненного цикла продукта.
- ▶ Владеть технологиями и инструментами командной разработки.
- ▶ Формировать, моделировать анализировать требования к продукту.
- ▶ Выбирать архитектуру. Оценивать стоимость. Кодировать.
- ▶ Тестировать. Оценивать качество. Сопровождать.

## ■ Информационные системы

- ▶ Владеть теорией СУБД и навыками их применения
- ▶ Выбирать, проектировать архитектуру.
- ▶ Эксплуатировать. Сопровождать.



# Почему математическая культура полезна профессионалам ИКТ

- **Математика** — язык описания и исследования в ИКТ.
  - ▶ Формирует и развивает абстрактное логическое мышление.
  - ▶ Вырабатывает навык регулярного подхода к решению задач.
  - ▶ Приучает не игнорировать “мелочи”.
- Стремительное расширение сферы ИКТ — инвестиции. Калейдоскоп новых парадигм, концепций, инструментов ...
- **Базовые** математические идеи появляются гораздо реже.
- **Математическая культура** — быстрое освоение любых концепций, методов и технологий ИКТ  
⇒ **долговременная карьера.**



# Разделы математики и ИКТ

... доказательство, контрпример, необходимое и достаточное условие

**Математический анализ** — база ВСЕХ направлений математики, формирует математическую культуру и позволяет легко осваивать и применять нужные разделы математики:

- **Дискретная математика** (в т.ч. комбинаторика), линейная алгебра — большинство задач, в т.ч:
  - ▶ Формальные языки — gcc, компиляторы компиляторов.
  - ▶ Структуры данных: графы, деревья, очереди, стеки и др.
  - ▶ Алгоритмы: поиск, сортировка, кратчайшие пути, оптимизация, маршрутизация в компьютерных сетях ... !
  - ▶ Криптография, оптимизация.
- Теория вероятностей, мат. статистика — финансы, Data Mining.
- Геометрия, дифф. уравнения — машинная графика, игры.
- Численные методы — машинное обучение, нейросети, робототехника.



# Определяющая роль математической культуры в ИКТ

## Д. ПОЙА МАТЕМАТИКА И ПРАВДОПОДОБНЫЕ РАССУЖДЕНИЯ

### Математик, физик и инженер.

— Физик верит, — сказал математик, — что 60 делится на все числа. Он замечает, что 60 делится на 1, 2, 3, 4, 5 и 6. Он проверяет несколько других чисел, например 10, 20 и 30, взятых, как он говорит, наугад. Так как 60 делится и на них, то он считает экспериментальные данные достаточными.

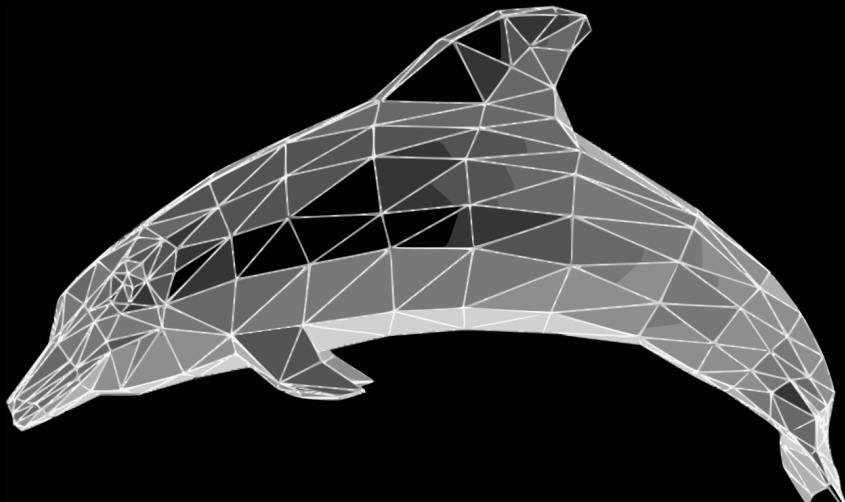
— Да, но взгляни на инженера, — возразил физик. — Инженер подозревает, что все нечётные числа простые. Во всяком случае, 1 можно рассматривать как простое число, доказывает он. Затем идут 3, 5 и 7, все, несомненно, простые. Затем идет 9 — досадный случай; 9, по-видимому, не является простым числом. Но 11 и 13, конечно, простые. Возвратимся к 9, — говорит он, — я заключаю, что 9 должно быть ошибкой эксперимента.





# Полигональная сетка трехмерного изображения дельфина

Учение о полигональных сетках — это большой подраздел компьютерной графики и геометрического моделирования.



# Как организовать учебную работу

Я слышу и забываю,  
Я вижу и запоминаю,  
Я делаю и понимаю.

Конфуций  
Мыслитель и философ  
б в до н.э.

Профессионализм в ИКТ — **только** за счет большого объема работы по решению учебных задач.

- Учебный процесс интенсивный, **много нового**.
- **Работоспособность**: режим дня, сон, питание, физкультура.  
Коты VK.
- **Меньше** прямого контроля, **больше** личной ответственности.
- **Трудолюбие! Усердие!**



# Как организовать учебную работу

## ■ Лекции.

- ▶ Активное восприятие — залог успеха в сессии, следить за ходом лекции, **Задавать вопросы.**
- ▶ Конспект — обязательно, **записывать новое, главное, перечитать** перед следующей лекцией.

## ■ Лабораторные и практические занятия.

- ▶ **Сдавать в срок**, иметь конспекты и учебники.
- ▶ **Иметь план работы** - схему программы, тестовые задачи и т.п.
- ▶ **Настроить среду** (предоставляется кафедрой ИМО) для разработки и отладки программ в месте проживания.

## ■ Прояснять все непонятое, не откладывая, на лекциях, на консультациях у преподавателей (будет расписание) или по эл.почте .



# Современные учебные планы

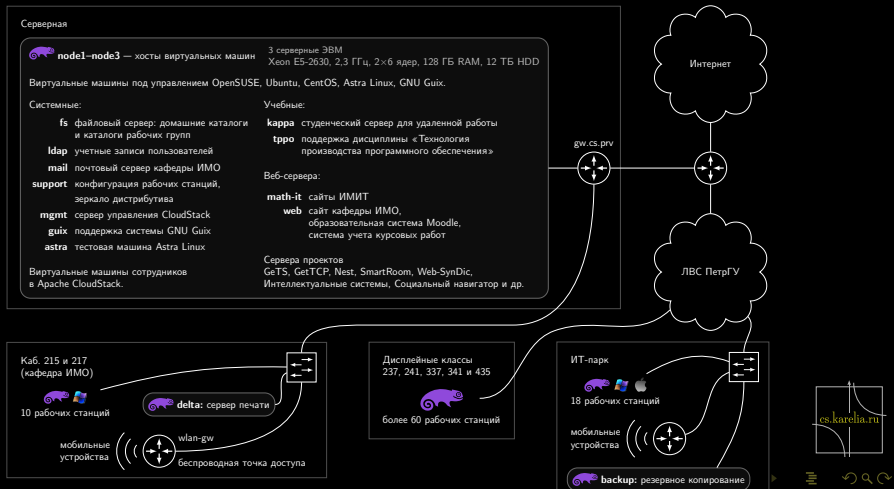
- Бакалавриат с 1993 г..
- Магистратура с 1997 г..
- Аспирантура — давно.
- ФГОС + Computing Curricula, регулярно публикуемые Association for Computing Machinery — ACM и Institute of Electrical and Electronics Engineers Computer Society — IEEE CS.
- Компетенции ИКТ = Математическая + Инженерная (технология разработки ПО).
- ИМИТ поддерживает УП на современном уровне.
- История развития учебных планов. Anatoly Voronin, Iurii Bogoiavlenskii, Vladimir Kuznetsov "Perspectives on the emergence of computing programs propelled by local industry in Russia", ACM Inroads, December 2015, Volume 6, No. 4.



# Информационно-вычислительная инфраструктура ИМИТ (ИВИ)

<http://cs.petrsu.ru/facilities/system.pdf>

Информационно-вычислительная инфраструктура Института математики и информационных технологий  
Поддерживается полный набор инструментов для учебного процесса и НИР/ОКР студентов и сотрудников (около 600 пользователей).



# Информационно-вычислительная инфраструктура ИМИТ (ИВИ)

Нужно изучить — **Веб страница описания ИВИ**

- Развитая, современная.
  - ▶ ОС Linux для студентов — готовый образ — запускается в виртуальной машине (работает как приложение в вашей ОС).
  - ▶ Краткое руководство по Emacs (применяется и как IDE).
  - ▶ Использование отладчика.
  - ▶ Система видео-конференц-связи Jitsi Meet.
- Поддерживается сотрудниками кафедры ИМО — В. А. Пономарев, Е. И. Рыбин, В. М. Димитров, Д. Б. Чистяков.
- Предоставляет современные программные продукты и технологии для учебного процесса, исследований и разработок.
- Кафедрой подготовлены — **Инструменты для студентов**



## ОС UNIX $\Rightarrow$ Linux

- Разработана “Программистами для программистов”.
- Переносимая, многозадачная, многопользовательская.
- Первая ОС с реализацией набора протоколов TCP/IP.
- Первая ОС на ЯВУ. Более 95% кода реализовано на языке C.
- Для переноса на другую архитектуру достаточно разработать кросс компилятор.
- Перенесена на много архитектур - Интернет вещей, котлы ТЭЦ.



# ОС UNIX ⇒ Linux

- Надежная - сервера эл.почты работают без перезагрузки месяцами.
- UNIX → MS DOS → Windows - Subsystem for Linux → Linux от Microsoft.
- Linux — лидер на рынке:
  - ▶ серверных ЭВМ;
  - ▶ смартфонов, планшетов;
  - ▶ высокопроизводительных вычислений (доля 99.4%.);
  - ▶ ОС сетевого оборудования (SOHO-роутеры, энтерпрайз-решения (Cisco));
  - ▶ ОС Аврора, бывшая Sailfish Mobile OS RUS;





# Программирование "в малом". Язык C — на все времена.

(C,libc) — новый подход — пара (Язык, Библиотека).

- Язык лаконичный, компактный - только описание алгоритмов.
- Все операции, например ввод/вывод, управление памятью, файлами и т. п. (потребности растут) реализуются в стандартной библиотеке языка — libc.

Брайан Керниган, Деннис Ритчи Язык программирования C, 1978 — "K&R" (40 изданий в США) — систематическое описание подхода — плодотворный, крайне популярный.

(C,libc) стандартизованы ANSI (1989 г.) и ISO (1990 г.)  
кроссплатформенность.



# Программирование "в малом". Язык C — lingua franca

Языки: C++, C#, Go, Java, JavaScript, PHP и десяткам других унаследовали базовый синтаксис C:

- ограничители { и };
- описание переменных;
- формы операторов for, while, if, switch с параметрами в скобках;
- комбинированные операции ++, --, +=, -= и др.

C — отлично выражает общие идеи программирования.

Конструкция — argc, argv

## Вывод

**Мышление современных программистов существенно C-ориентированно. C — фундаментальная основа квалификации программиста.**



# Программирование "в малом". Язык C — системное программирование.

- Эффективный машинный код. Используется вместо ассемблера.
- Применяется при разработке системного ПО для платформ от суперЭВМ до встроенных.
- Неполный список продуктов, реализованных на C:
  - ▶ Ядра ОС Unix (и всех ее клонов), MS Windows, Linux, OS X, iOS, Android, Windows Phone.
  - ▶ СУБД Oracle Database, MySQL, MS SQL Server, PostgreSQL.
  - ▶ Первые реализации языков Java, Python, Perl и PHP.
  - ▶ CPython — эталонная реализация.
- Реализован для большинства аппаратных платформ и ОС.



# Программирование "в малом". Язык C — libc.

- libc — часть любой из десятков реализации C.
  - ▶ GNU C Library — glibc — для ОС Linux. Содержит около 1500 функций — золотой фонд программирования.
  - ▶ Microsoft C Run-time Library, для ОС Windows;
  - ▶ libSystem, ОС X и iOS компании Apple;
  - ▶ Bionic, от Google для ОС Android.
- glibc обновляется два раза в год.

Многие низкоуровневые компоненты glibc прямо используются в программах на C, C++. Языки C#, Java, Perl, Python, Ruby и многие др. используют их косвенно.

Функции libc — фундаментальная основа всех современных программных продуктов



# Краткое содержание glibc (1270 стр.)

## Table of Contents

1 Introduction	19 Mathematics
2 Error Reporting	20 Arithmetic Functions
3 Virtual Memory Allocation And Paging	21 Date and Time
4 Character Handling	22 Resource Usage And Limitation
5 String and Array Utilities	23 Non-Local Exits
6 Character Set Handling	24 Signal Handling
7 Locales and Internationalization ns	25 The Basic Program/System Interface
8 Message Translation	26 Processes
9 Searching and Sorting	27 Inter-Process Communication
10 Pattern Matching	28 Job Control
11 Input/Output Overview	29 System Databases and Name Service
12 Input/Output on Streams	30 Users and Groups
13 Low-Level Input/Output	31 System Management
14 File System Interface	32 System Configuration Parameters
15 Pipes and FIFOs	33 Cryptographic Functions
16 Sockets	34 Debugging support
17 Low-Level Terminal Interface	35 Threads
18 Syslog	36 Internal probes



# Характеристики glibc

Общая характеристика glibc на 25.04.2021 - **25.07.23** со стр.  
<https://www.openhub.net/p/glibc>

Code Locations: [git://sourceware.org/git/glibc.git](https://sourceware.org/git/glibc.git)

In a Nutshell, GNU C Library...

has had 37,147 - **40307** commits made by 604 - **720** contributors representing 1,302,579 - **1419588** lines of code

is mostly written in C with an average number of source code comments: Across all C projects on Open Hub, 19% of all source code lines are comments. This holds true for GNU C Library as well.

has a well established, mature codebase maintained by a very large development team with increasing Y-O-Y commits

took an estimated 371 - **406** years of effort (COCOMO model) starting with its first commit in February, 1995 ending with its most recent commit 7 months ago



# Командный интерфейс против графического

## Interface: Command Line vs Graphical

- Письменность. Иероглифы против алфавита.
- Command Line Interface — CLI — интерфейс командной строки для взаимодействия с ОС.
- Популярный — `bash` — свободная версия Bourne shell.

Простые примеры:

- Цикл в командной строке `bash`:  
`for d in *.txt ; do mv $d $d.utf8 ; done`
- Простой скрипт в файле `k-m` — монтировать (присоединить) домашний каталог на удаленном сервере `karra` в каталог `KAPPA` на локальной ЭВМ):

```
#!/usr/bin/sh
cd
sshfs karra.cs.karelia.ru: KAPPA
```



# Командный интерфейс: конвейер команд

Символ | — соединение в конвейер.

Конвейер (pipe) — однонаправленный канал взаимодействия команд

```
bash du -k -c -d 1|sort -n -r|less
```

- **du - disk usage**: **-k**: блоки по 1k,, **-c**: получить суммарный объем каталога, **-d 1**: для подкаталогов ниже на 1 уровень;
- **sort - сортировать строки**: **-n**: числовая сортировка, **-r**: в обратном порядке.

```
2890548 total
2890548
2765996  ./Lectures
84592   ./Intel-guides
9736    ./GCC-guide
5528    ./My-tests
4100    ./GDB
28      ./Coding-Style
12      ./Voprosy-k-zachetu
8       ./Profilers
lines 1-10/10 (END)
```





# Командный интерфейс: сложная работа с файлами

Дано: 150 УМКД дисциплин кафедры, каждый в 5 файлах lowriter.

Нужно: каждый из них перевести в pdf и объединить в один pdf.

```
#!/bin/bash
# M. A. Kryshen. Getting pdf files of UMKD
# Target directory
base="$(pwd)"
# For each directory
find "$source" -type f | xargs -d '\n' -n 1 dirname | sort -u | \
  while read name; do
    echo "$name"
    dest="$base${name:${#source}}"
    mkdir -p "$dest"

    libreoffice --headless --convert-to pdf --outdir "$dest" "$name"/*

    # Title sheet to the begin
    mv "$dest"/*titul_list*.pdf "$dest"/000_titul_list.pdf
    # Adding empty page after title
    ln empty.pdf "$dest"/001_empty.pdf
    pdffunite "$dest"/*.pdf "$dest".pdf
    rm -f "$dest"/*.pdf
    rmdir "$dest"
done
```



# От учебы к профессиональной карьере

- <https://imit.petrso.ru/page/structure>  
Структура ИМИТ.
- [https://imit.petrso.ru/files/2018/f\\_248\\_dubna.pdf](https://imit.petrso.ru/files/2018/f_248_dubna.pdf)  
А. В. Воронин Развитие ИТ - технологий в Петрозаводском государственном университете.
- <https://imit.petrso.ru/employers>  
Работа в сфере ИКТ и ПО. Компании партнеры ИМИТ.
- <https://imit.petrso.ru/page/vacancy>  
Предложения компаний-партнеров.

<http://cs.petrso.ru/studies/index.php.ru>  
Кафедра ИМО, страницы дисциплин.

<http://cs.petrso.ru/staff/index.php.ru>  
Кафедра ИМО, сотрудники, элпочта.



# Магистры выпуска 2011 г.



Спасибо за внимание.

"... Важнейшим из всех искусств является для нас искусство программирования на языке Ассемблера..."

Ю.А.Богоявленский  
"О грядущих катастрофах"

**ДРАМА**

**НА**

**СИСТЕМНОЙ**

**МАГИСТРАЛИ.**

(с) зр.22302, 1991г.

