

# Zaawansowane metody programowania

**Dr Zbigniew Koziol - wykład**  
**Mgr Mariusz Woźny - laboratorium**

**Wykład I**

**Ogólnie o przedmiocie**

## Kontakt:

**softquake@gmail.com**

**mobile: 500 685 477**

**stationary at home: 17 229 8940**

**Treści wykładu będą dostępne pod:**

**<http://www.nanophysics.pl>**

## Warunki zaliczenia i oceny:

- zaliczenie zajęć laboratoryjnych
- obecność na wykładzie, aktywność
- przygotowanie i zaprezentowanie przed wszystkimi prezentacji multimedialnej (PowerPoint, etc) na temat wybrany z zaproponowanych lub własny (10-15 minut)

O tworzeniu prezentacji w PowerPoint:

<http://nanophysics.pl/teaching/prezentacje-HowTo.pdf>

## Systemy operacyjne

- **Windows, Linux, Solaris...** dziesiątki, setki i dziesiątki lub setki wersji każdego z nich

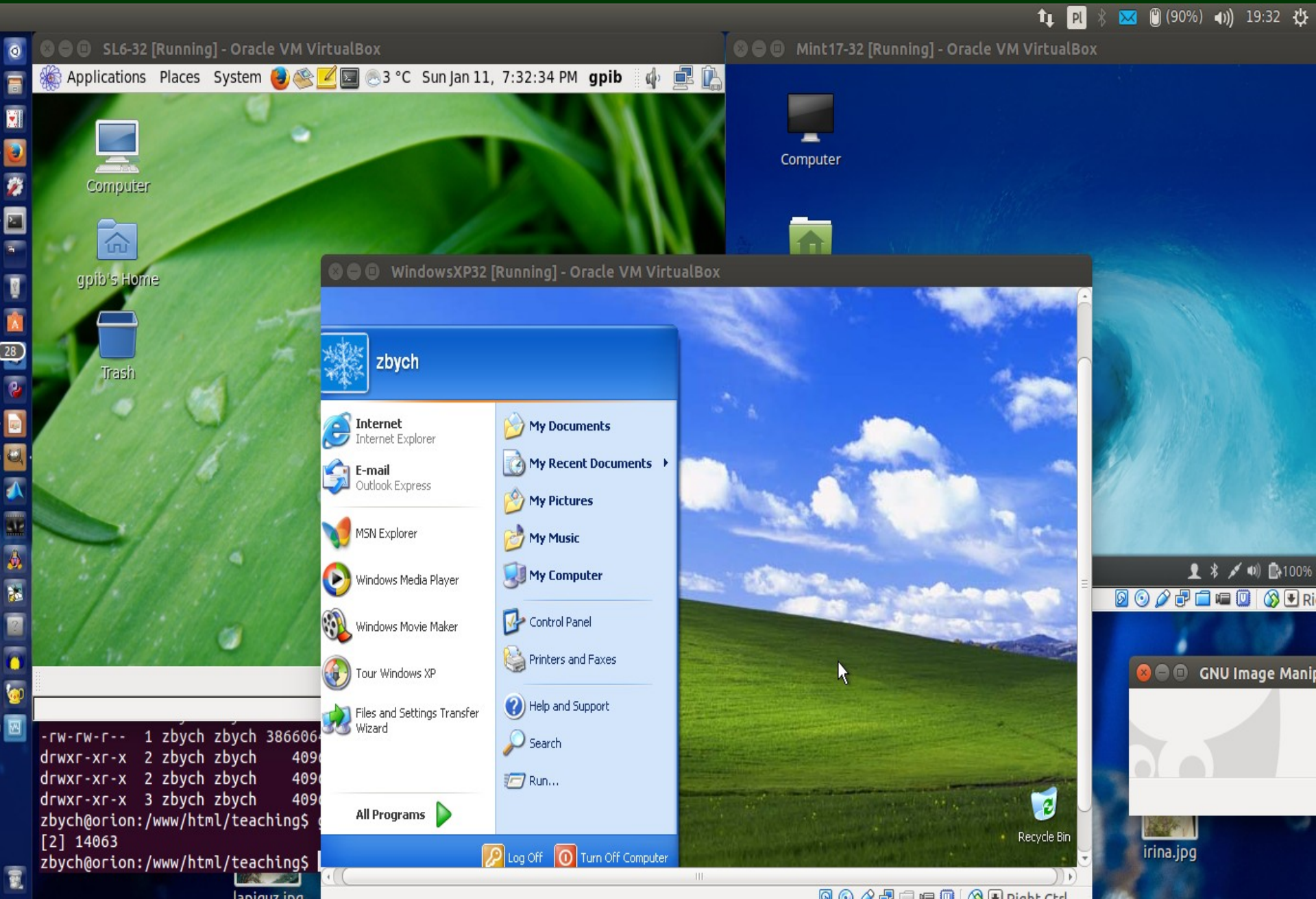


## VirtualBox

- Instalacja Linuxa na pen-drive:  
<http://nanophysics.pl/linux-pen-drive.php>
- **VirtualBox** <http://www.virtualbox.org>

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8), DOS/Windows 3.x, Linux (2.4, 2.6 and 3.x), Solaris and OpenSolaris, OS/2, and OpenBSD.



# Języki programowania

- **assambler** (MOVE, POP; elementarne operacje na bajtach i rejestrach procesora; dla człowieka niemal nieczytelny)
- **C/C++**
- **C#**
- **Fortran**
- **Java** (Sun Microsystems)
- **PHP** (Pretty Home Page, <http://www.php.net>)
- **Perl** (wspaniały do obróbki tekstów i danych w formacie ASCII) <http://www.perl.com>
- **Pascal / Free Pascal** <http://www.freepascal.org>
- **Basic / Free Basic** <http://www.freebasic.org>
- **Python** <http://www.python.org>
- **JavaScript (Netscape) (przykład?)**
- **SQL** (standart raczej, nie język, do komunikacji z bazami danych i manipulacji na danych)
- **bash** (Linux i inne systemy uniksowe, do pracy z terminala i dla skryptów systemowych)

**Zajęcia Lab będą polegały na nauce oprogramowania na przykładzie, głównie, LabView.  
Na wykładzie zaś jako przykłady używany będzie kod w różnych językach.**



# Języki programowania

**Dobry programista zna większość z wymienionych języków oraz jeszcze i inne.**

**Beznadziejne... ? Nie!**

**Gdy poznać dobrze choćby jeden język, wtedy łatwo zrozumieć i inne!**

## Oprogramowanie specjalistyczne: komercyjne i swobodne

- **Comsol Multiphysics**
- **LabView**
- **Maple**
- **Synopsys' Sentaurus**

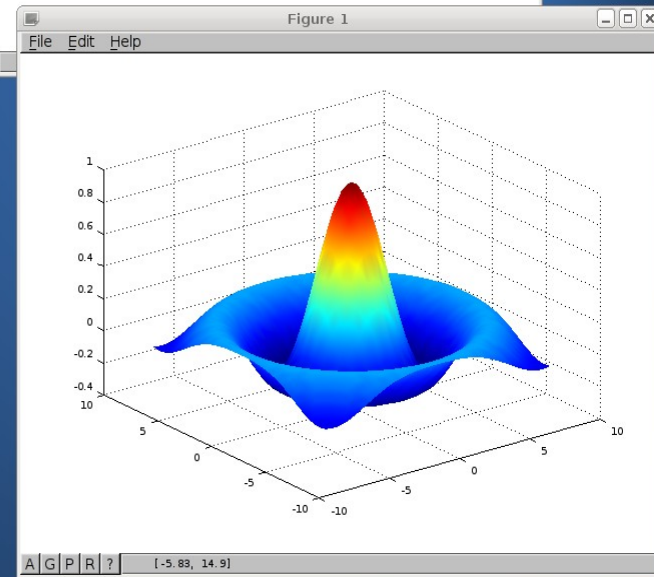
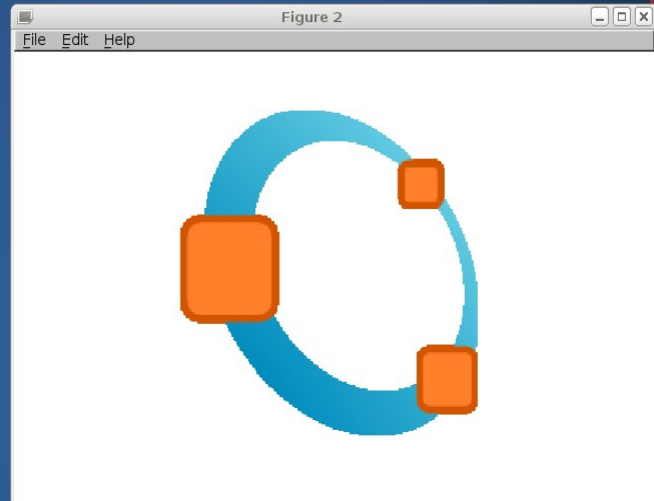
**Elmer**

**Octave** (<https://www.gnu.org/software/octave/>)

**Aestimo**

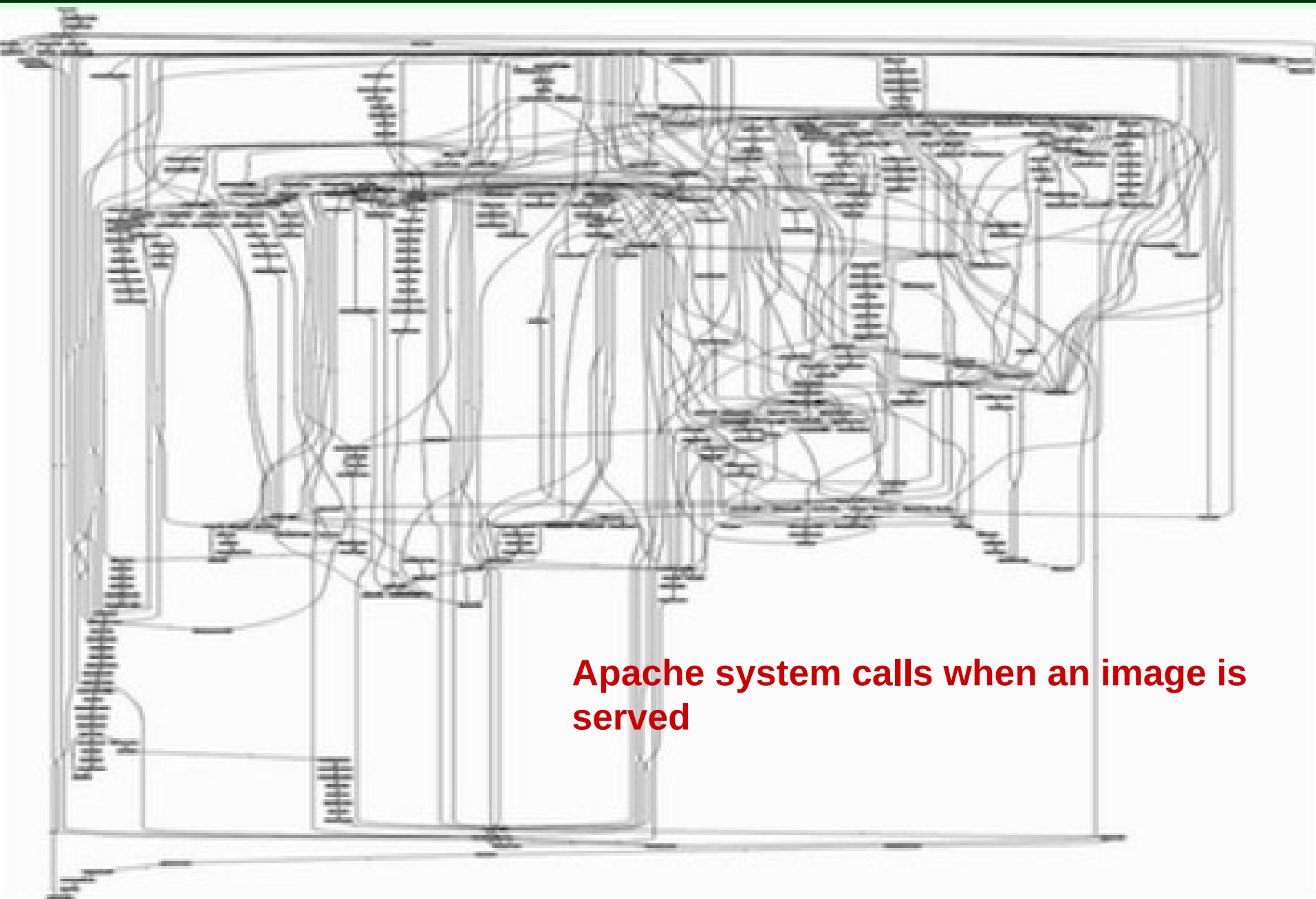


```
emacs@coredump  
  
narg = 1;  
while (narg <= numel (varargin))  
  arg = varargin{narg++};  
  if (isnumeric (arg))  
    if (numel (arg) == 2 || isempty (arg))  
      display_range = arg;  
    elseif (columns (arg) == 3)  
      indexed = true;  
      colormap (arg);  
    elseif (! isempty (arg))  
      error ("imshow: argument number %d is invalid", narg+1);  
    endif  
  elseif (ischar (arg))  
    switch (arg)  
    case "displayrange";  
      display_range = varargin{narg++};  
    case {"truesize", "initialmagnification"}  
      warning ("image: zoom argument ignored -- use GUI features");  
    otherwise  
      warning ("imshow: unrecognized property %s", arg);  
      narg++;  
    endswitch  
  else  
    error ("imshow: argument number %d is invalid", narg+1);  
  endif  
endwhile  
  
## Set default display range if display_range not set yet.  
if (isempty (display_range))  
  display_range = [min(im(:)), max(im(:))];  
end  
imshow.m 51% L119 (Octave Abbrev Fill)--17:39-----
```



```
Terminal  
  
Please contribute if you find this software useful.  
For more information, visit http://www.octave.org/help-wanted.html  
  
Read http://www.octave.org/bugs.html to learn how to submit bug reports.  
  
For information about changes from previous versions, type `news'.  
  
octave:1> graphics_toolkit fltk  
octave:2> sombrero  
octave:3> shading interp  
octave:4> logo = imread ("logo.png");  
warning: your version of GraphicsMagick limits images to 8 bits per pixel  
octave:5> size (logo)  
ans =  
    200    200     3  
  
octave:6> class (logo)  
ans = uint8  
octave:7> figure  
octave:8> imshow (logo)  
octave:9> edit imshow  
octave:10>
```

**D**laczego struktura kodu, zaprojektowanie są ważne?



**Apache system calls when an image is served**



**IIS (Microsoft) system calls when an image is served**

## **Programowanie jest jak projektowanie skomplikowanej konstrukcji architektonicznej albo urządzenia elektronicznego**

- budujemy program z istniejących klocków, należy je odpowiednio dobrać i poskładać
- warto stworzyć plan już na początku, diagram przedstawiający powiązania logiczne między elementami/blokami
- plan posłuży również do komunikacji z innymi programistami: złożone projekty programistyczne mogą polegać na pracy w dużych grupach osób, przez lata, gdzie każda osoba wykonuje tylko element konstrukcji

**Dalej:**

## **Elementy struktury kodu**

- **dyrektywy**
- **biblioteki i moduły zewnętrzne**
  
- **pętle FOR**
- **WHILE**
- **IF**
- **LABEL i GOTO**
- **funkcje i procedury**



## Jeszcze dalej:

- operacje wejścia/wyjścia
- grep, przekierowywanie danych, perl i bash, gnuplot
- STD INPUT, STD OUTPUT, STD ERROR
- | (pipe)

## A jeszcze potem:

- niektóre algorytmy używane często