

Wydział Nauk Technicznych

UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN
The Faculty of Technical Sciences
POLAND, 10-957 Olsztyn, M. Oczapowskiego 11
tel.: (48)(89) 5-23-32-40, fax: (48)(89) 5-23-32-55
URL: <http://www.uwm.edu.pl/edu/sobieski/> (in Polish)



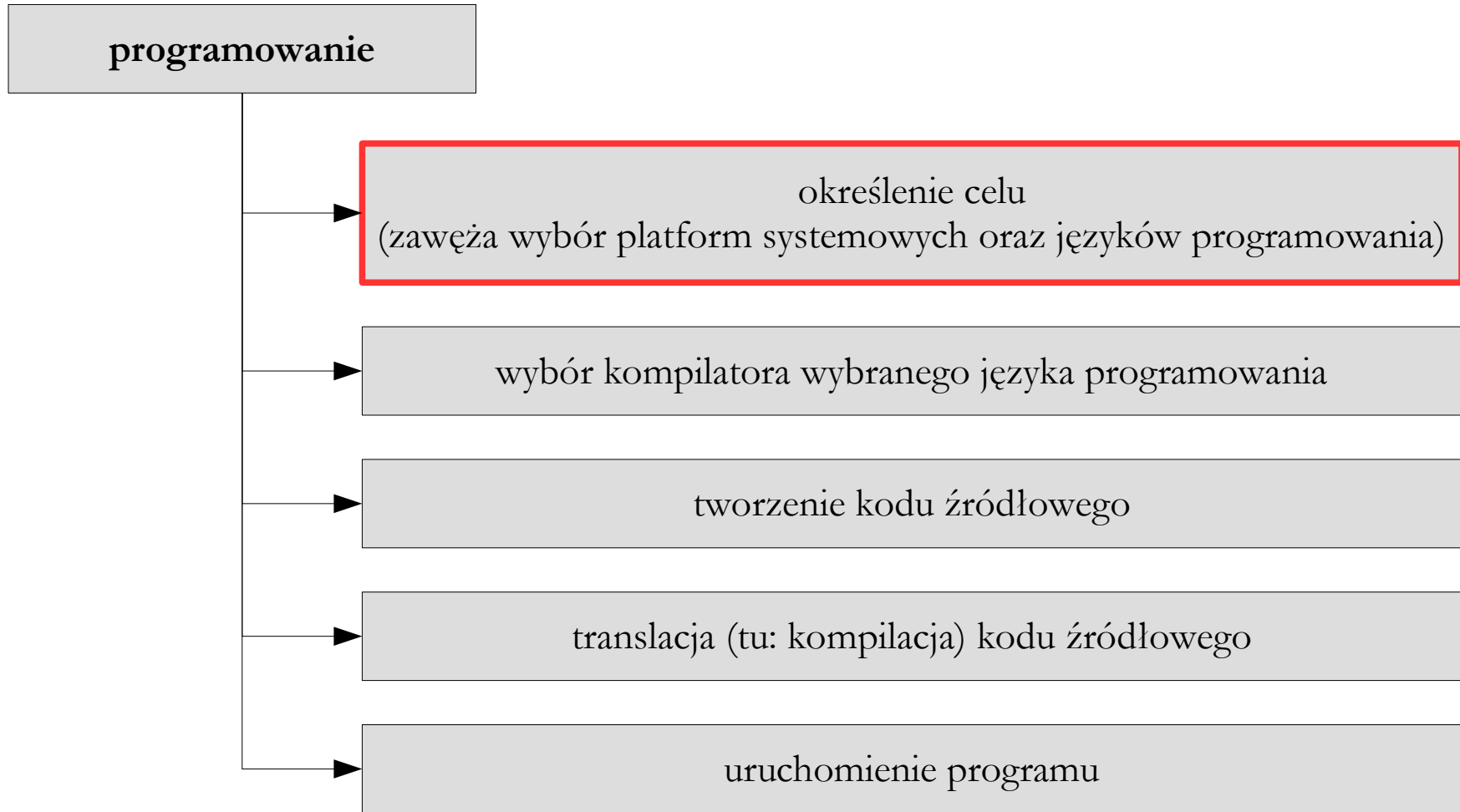
Języki Programowania

Kompilatory Fortranu

Wojciech Sobieski

Olsztyn, 2001-2021

Ścieżka działań



Określenie celu

Określenie celu:

- „łatwe” tworzenie aplikacji w systemach Windows i Linux, służących do przeprowadzania **szybkich obliczeń matematycznych** z możliwością generowania **wykresów 2D i 3D** oraz **profesjonalnych wizualizacji**, a także z możliwością tworzenia wygodnych **nakładek graficznych**.

potrzebny szybki język programowania: Fortran lub C

potrzebna dodatkowa biblioteka lub zewnętrzne środowisko wizualizacyjne

potrzebna dodatkowa biblioteka lub zewnętrzne środowisko wizualizacyjne

potrzebny język typu RAD

Określenie celu

The screenshot displays the PathFinder software interface. On the left, a command prompt window shows the following output:

```
C:\Windows\system32\cmd.exe
tortuosity [m/m] : 1.244451
tortuosity after smoothing [m/m] : 1.160995
tortuosity from Yu and Li formula [m/m] :
volume of the bed (bulk volume) [m^3] :
inner surface of the solid body [m^2] :
specific surface of the porous body (K) [1/m] :
specific surface of the porous body (C) [1/m] :
volume of the porous body [m^3] :
porosity [m^3/m^3] :
Ergun A [1/m^2] :
Ergun 2*B [1/m] :
Kozeny-Carman (C_KC=1) [1/m^2] :
Kozeny-Carman (C_KC=1) after smoothing [1/m^2] :
The time of calculations [s] :
```

Below the command prompt is a gnuplot graph window titled "PathFinder - Final Path spheres surround". The graph shows a 3D plot of a final path (red spheres) on a surface. The axes are labeled x, y, and z. The x-axis ranges from -0.06 to 0.06, the y-axis from -0.06 to 0.06, and the z-axis from 0 to 0.25. The path starts at the origin and moves upwards and outwards, ending at a peak of approximately 0.25 on the z-axis.

On the right, the pathGUI v. 1.1 configuration window is shown. The project name is "pfc3d_18188 18188". The domain geometry is set to "cylinder". The method for calculating the triangle center is "the triangle centre of gravity". The characteristic dimension for the Ideal Location is "average length of the triangle side (l_ave)". The correction method for the Ideal Location is "function (depending on the triangle size)". The parameters are:

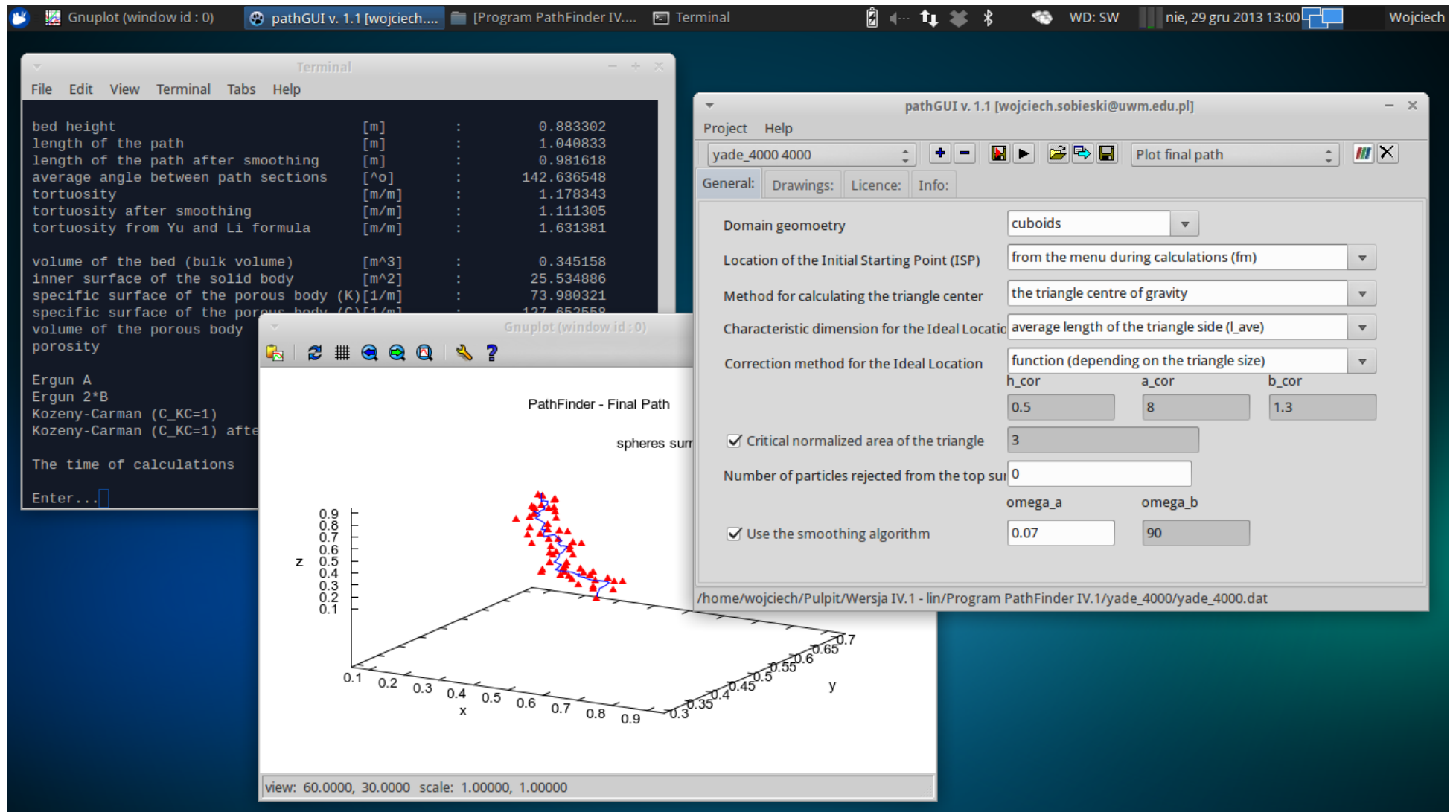
Parameter	Value
h_cor	0.5
a_cor	8
b_cor	1.3
Critical normalized area of the triangle	3
Number of particles rejected from the top surface	16
Use the smoothing algorithm	<input checked="" type="checkbox"/>
omega_a	0.07
omega_b	90

The pathGUI window also shows the file path: C:\Version IV.1\PathFinder IV.1\pfc3d_18188\pfc3d_18188.dat.

PathFinder (Windows)

– raport końcowy wraz ze ścieżką finalną (przykład z PFC^{3D}) oraz narzędzie pathGUI

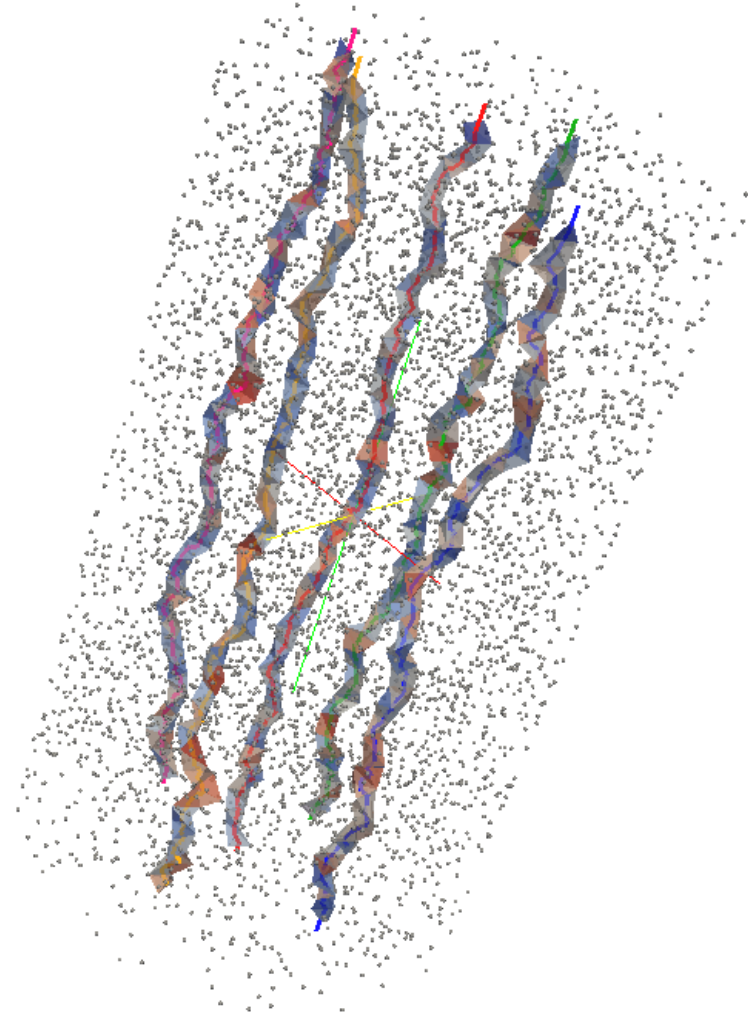
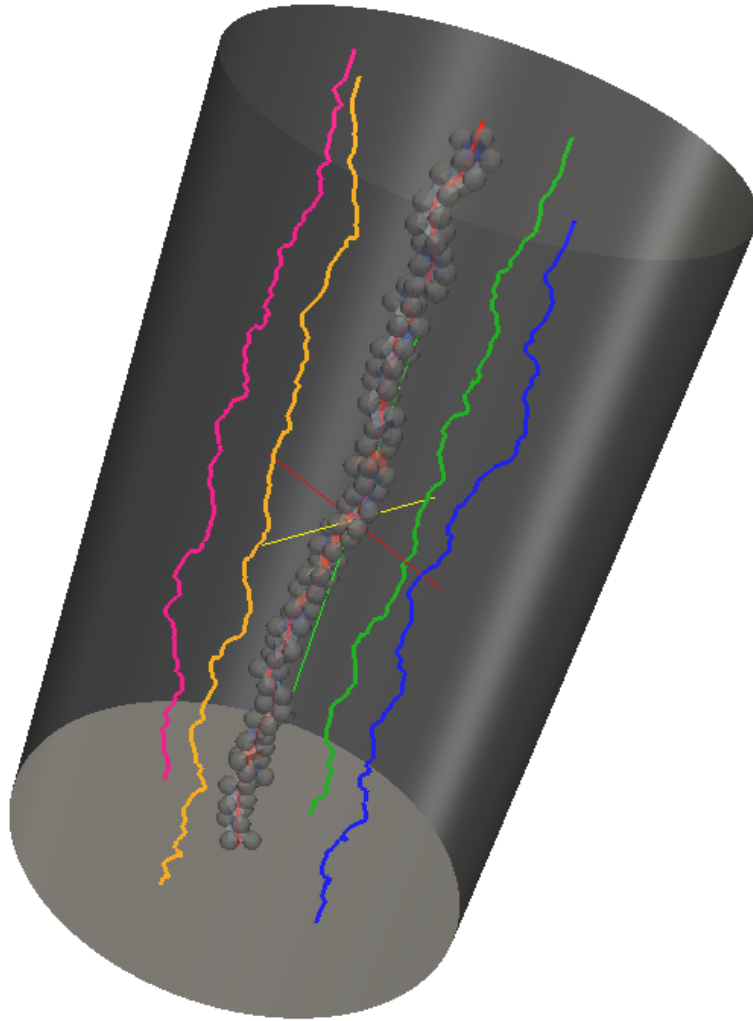
Określenie celu



PathFinder (Ubuntu Linux)

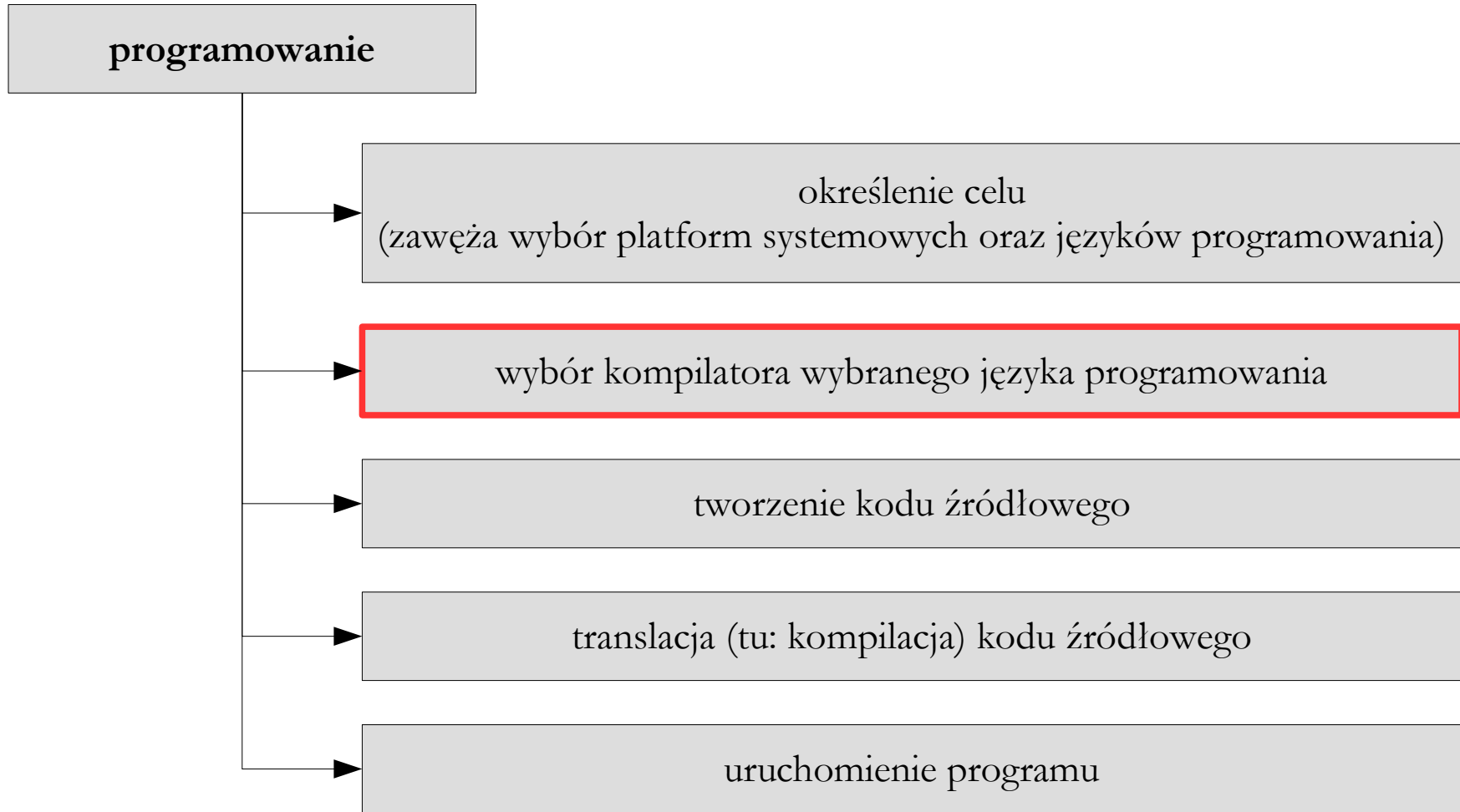
– raport końcowy wraz ze ścieżką finalną (przykład z YADE) oraz narzędzie pathGUI

Określenie celu



Przykłady wizualizacji w ParaView wyników obliczeń uzyskanych w programie PathFinder

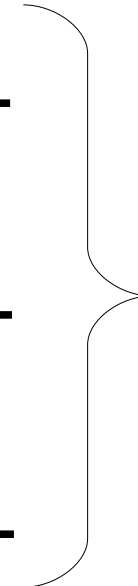
Ścieżka działań



Narzędzia realizacji celu

Narzędzia realizacji celu (wynikające ze stylu programowania):

- język Fortran – część obliczeniowa:
 - kompilatory komercyjne, np.: Intel Fortran
 - kompilatory darmowe, np.: **gfortran** (GPL)
- środowisko RAD – graficzne nakładki:
 - komercyjne, np.: Borland Delphi
 - otwarte, np.: środowisko **Lazarus** (GPL)
- wykresy 2D i 3D:
 - komercyjne, np. MatLab
 - otwarte, np.: **Gnuplot** (GPL)
- wizualizacje:
 - komercyjne, np.: TecPlot
 - otwarte, np: **ParaView** (GPL), MayaVi (GPL)



Fortran

Fortran – język programowania pierwotnie zaprojektowany do zapisu programów obliczeniowych, był niegdyś językiem proceduralnym, obecnie jest nadal rozwijanym językiem ogólnego przeznaczenia. Umożliwia programowanie **strukturalne**, **obiektowe** (Fortran 90/95), **modularne** i **równoległe** (Fortran 2008).

Zastosowaniami Fortranu są, między innymi, obliczenia naukowo-inżynierskie, numeryczne, symulacje komputerowe itp.

Początkowe wersje Fortranu miały mocno ograniczone możliwości, ale dzięki łatwości opanowania, Fortran stał się najpopularniejszym językiem do obliczeń numerycznych.

Fortran

Standard – zestaw cech i właściwości, które muszą być spełnione przez translator danego języka programowania.

Jeden język programowania może mieć (w czasie rozwoju) wiele standardów.

Implementacja standardu (skrótowo: kompilator, assembler lub interpreter) – konkretny program służący do translacji kodów źródłowych danego języka programowania: musi spełniać standardy, ale szczegóły wykonania (np. treść komunikatów generowanych w trakcie procesu translacji) zależą od implementującego (osoby, firmy, instytucji itp.).

Jeden standard języka programowania może mieć wiele implementacji.

Fortran

Standardy/wersje języka Fortran:

- **FORTTRAN**: 1951-55 (początki języka)
- **FORTTRAN II**: 1958
- **FORTTRAN III**: 1958 (wersja nie wydana oficjalnie)
- **FORTTRAN IV**: 1962
- **FORTTRAN 66**: 1966 (pierwszy oficjalny standard)
- **FORTTRAN 77**: 1977
- **Fortran 90**: 1991-92
- **Fortran 95**: 1995 (drobne poprawki)
- **Fortran 2003**: 2003
- **Fortran 2008**: 2008
- **Fortran 2018**: 2018 – zwiększenie zgodności z językiem C
 - <https://wg5-fortran.org/N1901-N1950/N1942.pdf>

Kompilatory języka Fortran

Kompilatory języka Fortran:

- komercyjne (wybrane):
 - Microsoft Power Station
 - Lahey/Fujitsu Fortran
 - Intel Visual Fortran Compiler
 - PGI Visual Fortran
 - PGI CUDA Fortran Compiler
 - Salford FTN95
- otwarte:
 - g77
 - gfortran
 - g95

```
! iteracja w czasie:

do i=1,n

! wybor stalej tlumienia w zalezności od kierunku ruchu:

if (u.gt.0) then
  c=c1
else
  c=c2
end if

! obliczenie pochodnych:

f1=u
f2=-g-(k/m)*(y-w)-c*u*abs(u)/m

! obliczenie nowych wartosci zmiennych:

y=y+f1*dt
u=u+f2*dt
t=t+dt

! przypisanie wartosci do tablic na potrzeby rysunku:

ygraf(i)=y
ugraf(i)=u
tgraf(i)=t

!wyswietlenie wynikow dla i-tego kroku:

!   write(*,'(i6,3f12.4)') i,t,y,u

end do

!wyswietlenie informacji o kolumnach:

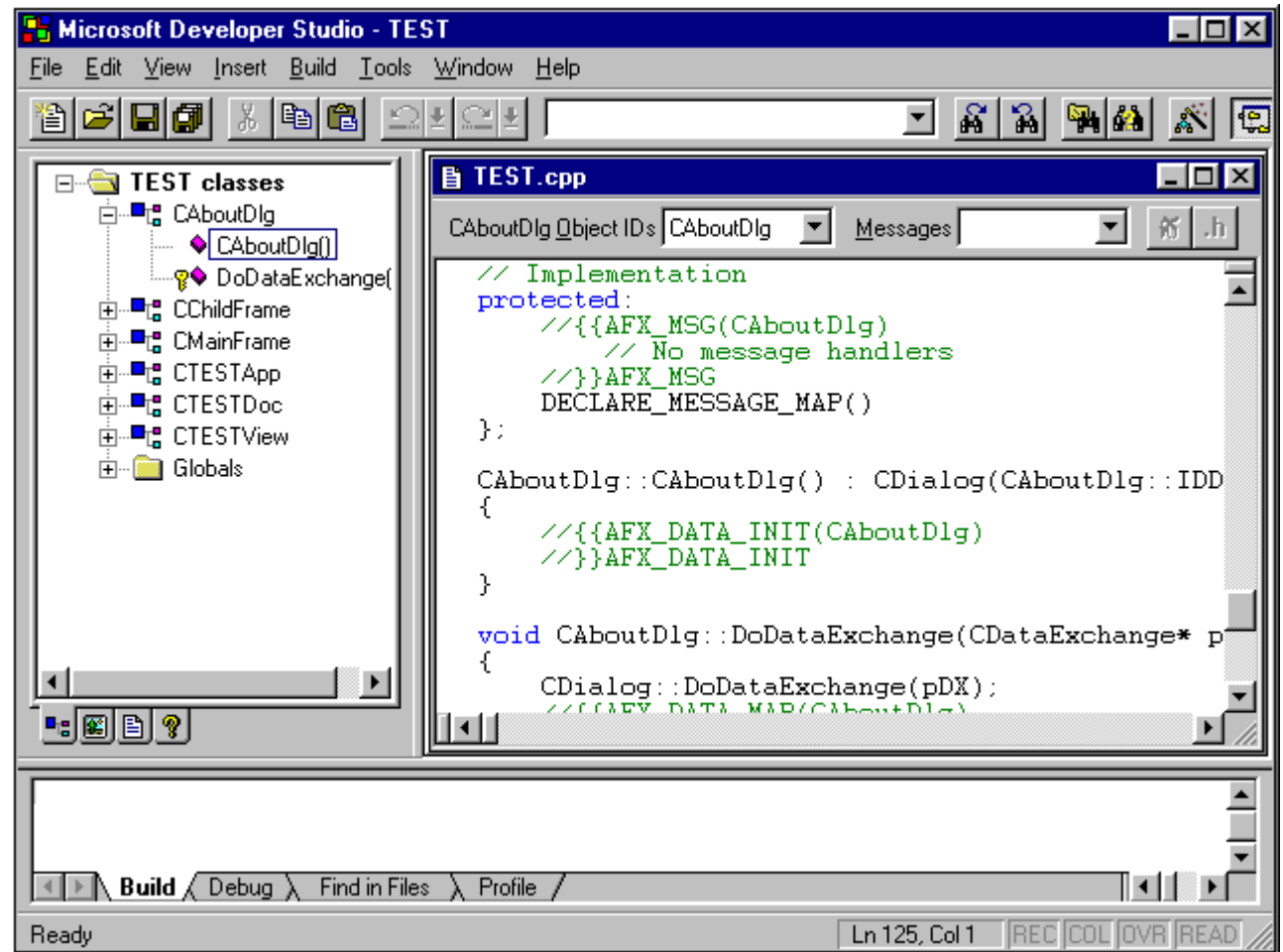
write(*,*) '-----'
write(*,*) ' i:      t:      y:      u:'

! zapis do pliku:

open(1,file='wyniki.txt', status='unknown')
do i=1, n
write(1,*) tgraf(i), ygraf(i), ugraf(i)
end do
close(1)

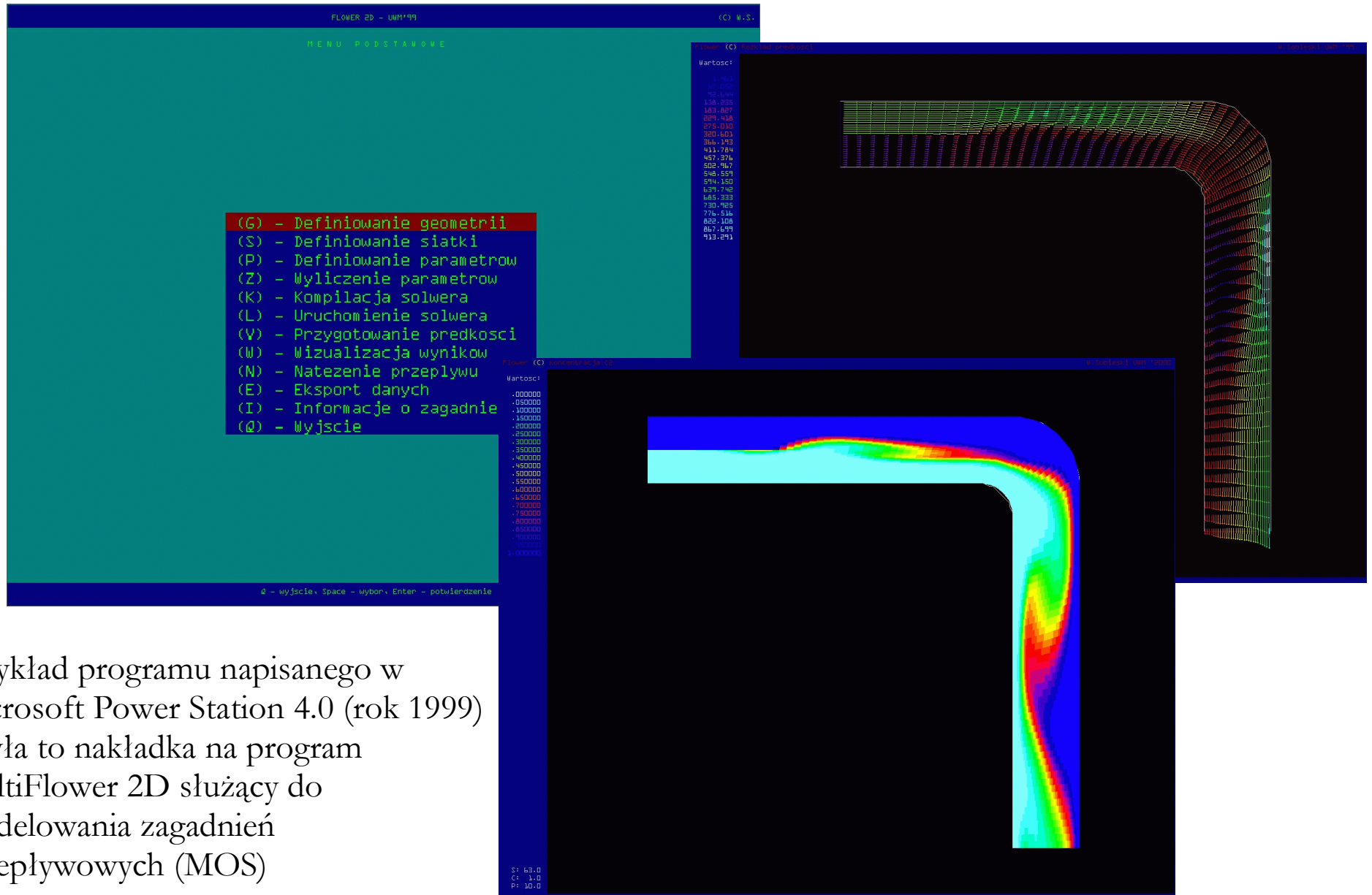
! wywołanie procedury graficznej:
```

Kompilatory języka Fortran



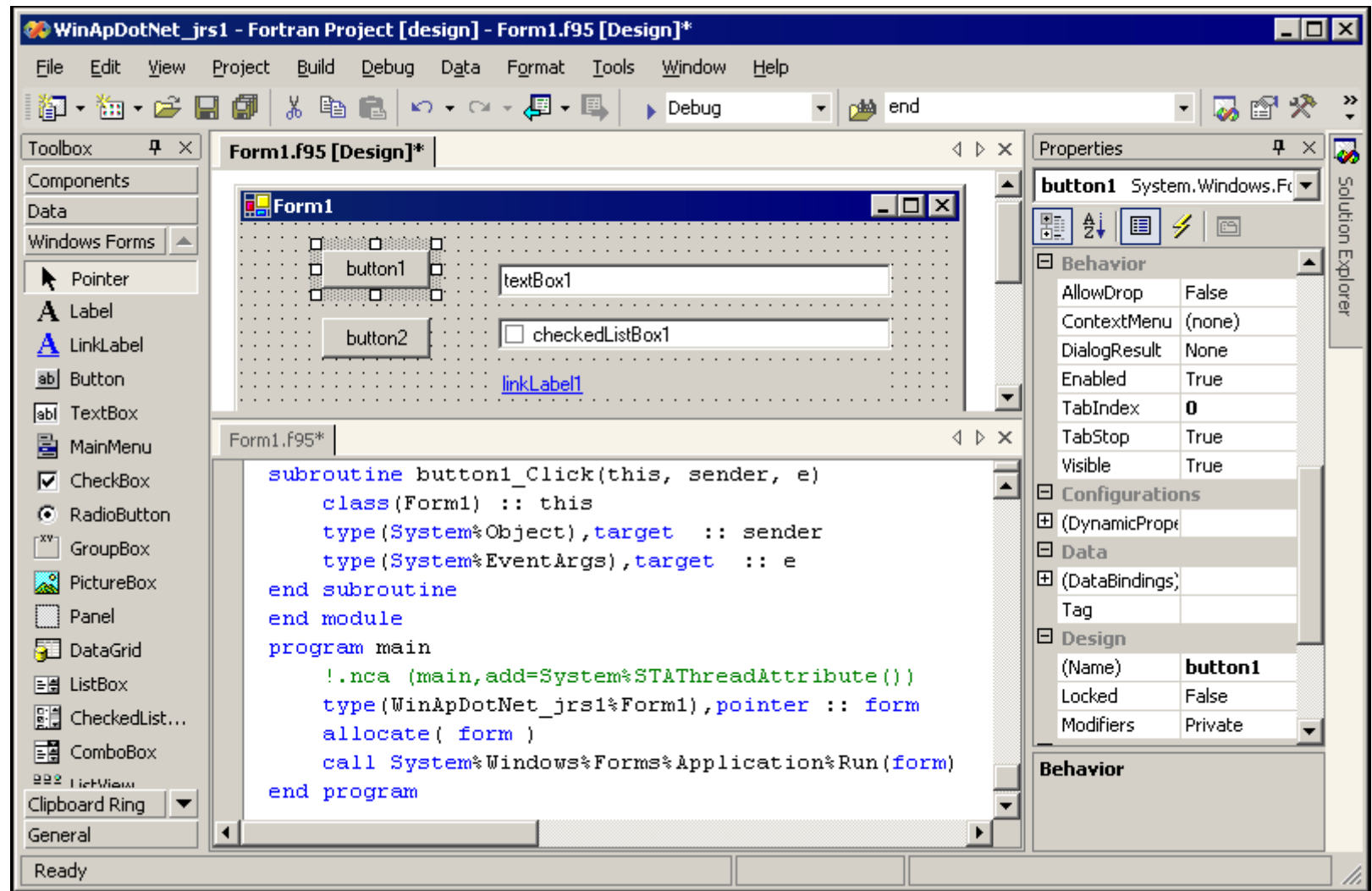
Microsoft Power Station 4.0

Kompilatory języka Fortran



przykład programu napisanego w Microsoft Power Station 4.0 (rok 1999) - była to nakładka na program MultiFlower 2D służący do modelowania zagadnień przepływowych (MOS)

Kompilatory języka Fortran



Lahey Fortran

Kompilatory języka Fortran

The screenshot shows the Microsoft Visual Studio (Administrator) interface during a debug session of a Fortran program named 'pi.f90'. The main window displays the source code, and the bottom panels show the Processes and Breakpoints toolbars.

```
19 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
20 program main
21
22 include 'mpif.h'
23
24 double precision PI25DT
25 parameter      (PI25DT = 3.141592653589793238462643d0)
26
27 double precision mypi, pi, h, sum, x, f, a
28 integer n, myid, numprocs, i, rc
29 ! function to integrate
30 f(a) = 4.d0 / (1.d0 + a*a)
31
32 call MPI_INIT( ierr )
33 call MPI_COMM_RANK( MPI_COMM_WORLD, myid, ierr )
34 call MPI_COMM_SIZE( MPI_COMM_WORLD, numprocs, ierr )
35 print *, 'Process ', myid, ' of ', numprocs, ' is alive'
36
37 sizetype = 1
38 sumtype = 2
39
40 ! Hard-code the number of intervals because stdin is tricky with MPI
41 n = 100000
42
43 call MPI_BCAST(n,1,MPI_INTEGER,0,MPI_COMM_WORLD,ierr)
44
45 ! check for quit signal
46 if ( n .le. 0 ) goto 30
47
48 ! calculate the interval size
49 h = 1.d0/n
```

Processes

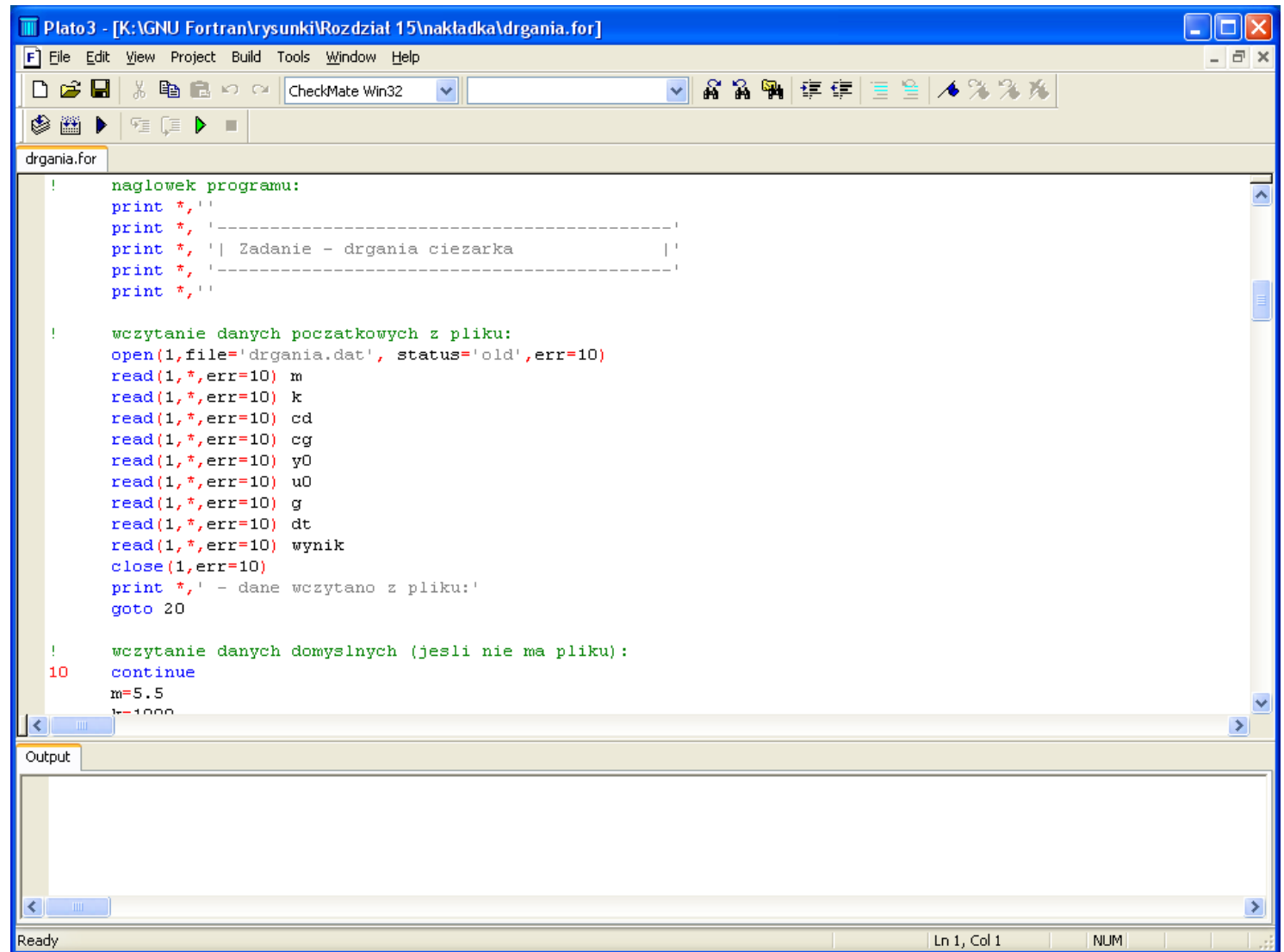
Name	ID	Path	Title
pi.exe	N/A	C:\SHARED\demos\pi\x64\Debug\pi.exe	0 (HPC-NODE06: 2308)
pi.exe	N/A	C:\SHARED\demos\pi\x64\Debug\pi.exe	1 (HPC-NODE06: 2236)
pi.exe	N/A	C:\SHARED\demos\pi\x64\Debug\pi.exe	2 (HPC-NODE07: 2416)
pi.exe	N/A	C:\SHARED\demos\pi\x64\Debug\pi.exe	3 (HPC-NODE07: 496)
pi.exe	N/A	C:\SHARED\demos\pi\x64\Debug\pi.exe	4 (HPC-NODE08: 2924)
pi.exe	N/A	C:\SHARED\demos\pi\x64\Debug\pi.exe	5 (HPC-NODE08: 2164)
pi.exe	N/A	C:\SHARED\demos\pi\x64\Debug\pi.exe	6 (HPC-NODE09: 2804)

Breakpoints

Name	Labels	Condition	Hit Count
pi.f90, line 35 character 1		(no condition)	break always
pi.f90, line 35		(no condition)	break always (currently 1)
pi.f90, line 35		(no condition)	break always (currently 1)
pi.f90, line 35		(no condition)	break always (currently 1)
pi.f90, line 35		(no condition)	break always (currently 1)
pi.f90, line 35		(no condition)	break always (currently 1)
pi.f90, line 35		(no condition)	break always (currently 1)

PGI Visual Fortran

Kompilatory języka Fortran



```
Plato3 - [K:\GNU Fortran\rysunki\Rozdział 15\nakladka\drgania.for]
File Edit View Project Build Tools Window Help
CheckMate Win32
drgania.for
! naglowek programu:
print *, ''
print *, '-----'
print *, '| Zadanie - drgania ciezarka |'
print *, '-----'
print *, ''

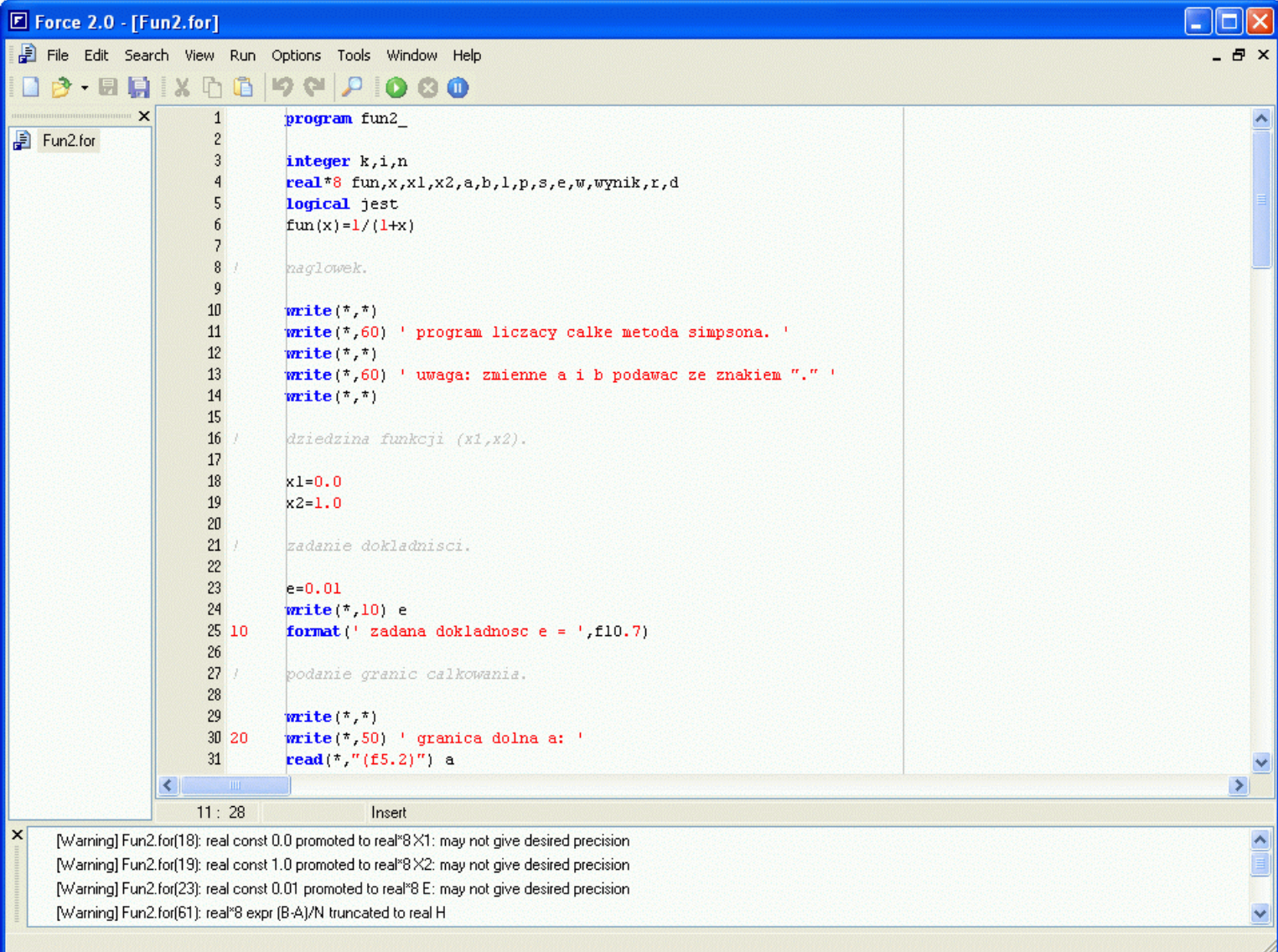
! wczytanie danych poczatkowych z pliku:
open(1,file='drgania.dat', status='old',err=10)
read(1,*,err=10) m
read(1,*,err=10) k
read(1,*,err=10) cd
read(1,*,err=10) cg
read(1,*,err=10) y0
read(1,*,err=10) u0
read(1,*,err=10) g
read(1,*,err=10) dt
read(1,*,err=10) wynik
close(1,err=10)
print *, ' - dane wczytano z pliku:'
goto 20

! wczytanie danych domyslonych (jesli nie ma pliku):
10 continue
m=5.5
k=1000

Output
Ready Ln 1, Col 1 NUM
```

Salford FTN95
Compiler
(wersja Personal)

Kompilatory języka Fortran



The screenshot shows the Force 2.0 Fortran compiler interface. The main window displays a Fortran program named 'fun2.for' with the following code:

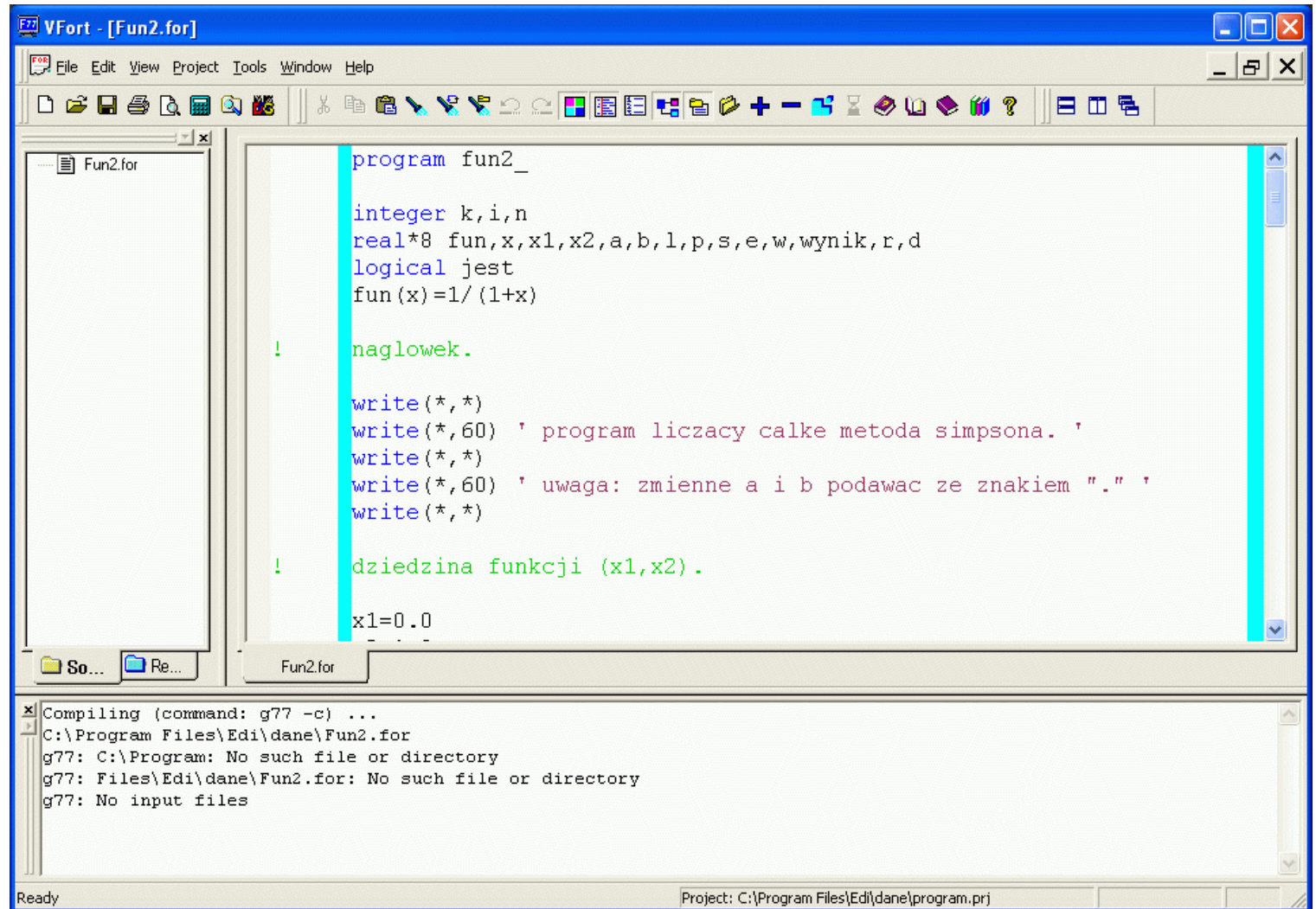
```
1 program fun2_  
2  
3 integer k,i,n  
4 real*8 fun,x,x1,x2,a,b,l,p,s,e,w,wynik,r,d  
5 logical jest  
6 fun(x)=1/(1+x)  
7  
8 ! naglowek.  
9  
10 write(*,*)  
11 write(*,60) ' program liczacy calke metoda simpsona. '  
12 write(*,*)  
13 write(*,60) ' uwaga: zmienne a i b podawac ze znakiem "." '  
14 write(*,*)  
15  
16 ! dziedzina funkcji (x1,x2).  
17  
18 x1=0.0  
19 x2=1.0  
20  
21 ! zadanie dokladnisci.  
22  
23 e=0.01  
24 write(*,10) e  
25 10 format(' zadana dokladnosc e = ',f10.7)  
26  
27 ! podanie granic calkowania.  
28  
29 write(*,*)  
30 20 write(*,50) ' granica dolna a: '  
31 read(*,"(f5.2)") a
```

The status bar at the bottom of the window shows the time 11:28 and the mode 'Insert'. Below the main window, a list of compilation warnings is displayed:

- [Warning] Fun2.for(18): real const 0.0 promoted to real*8 X1: may not give desired precision
- [Warning] Fun2.for(19): real const 1.0 promoted to real*8 X2: may not give desired precision
- [Warning] Fun2.for(23): real const 0.01 promoted to real*8 E: may not give desired precision
- [Warning] Fun2.for(61): real*8 expr (B-A)/N truncated to real H

środowisko
Fortran Force 2.0
(g77)

Kompilatory języka Fortran



The screenshot shows the VFort IDE interface. The main window displays the following Fortran code:

```
program fun2_  
  
integer k,i,n  
real*8 fun,x,x1,x2,a,b,l,p,s,e,w,wynik,r,d  
logical jest  
fun(x)=1/(1+x)  
  
! naglowek.  
  
write(*,*)  
write(*,60) ' program liczacy calke metoda simpsona. '  
write(*,*)  
write(*,60) ' uwaga: zmienne a i b podawac ze znakiem "." '  
write(*,*)  
  
! dziedzina funkcji (x1,x2).  
  
x1=0.0  
_ . _
```

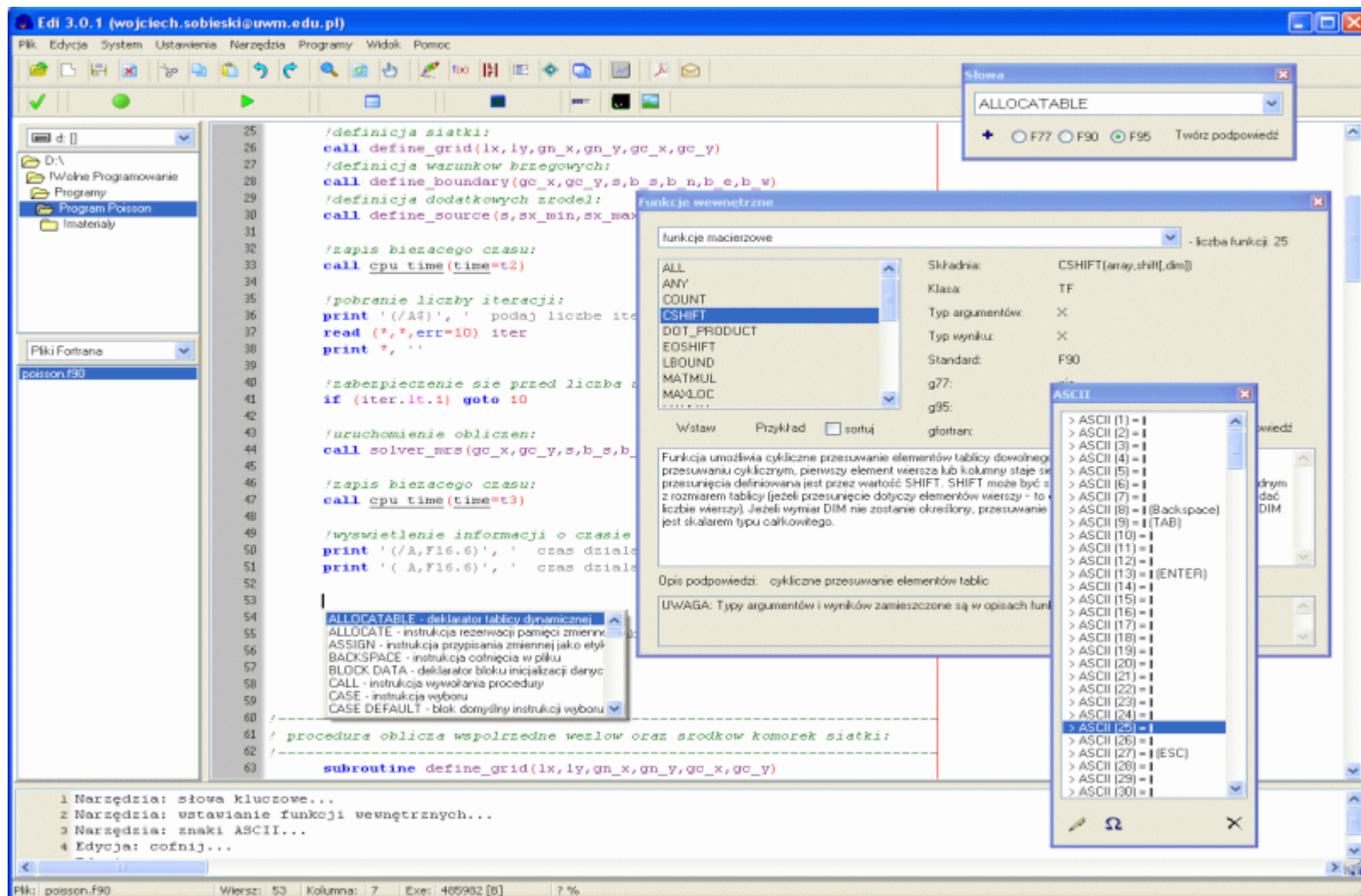
The bottom panel shows the compilation command and error messages:

```
Compiling (command: g77 -c) ...  
C:\Program Files\Edi\dane\Fun2.for  
g77: C:\Program: No such file or directory  
g77: Files\Edi\dane\Fun2.for: No such file or directory  
g77: No input files
```

The status bar at the bottom indicates the project path: Project: C:\Program Files\Edi\dane\program.prj

środowisko
VFort (g77)

Kompilatory języka Fortran



środowisko
Edi v. 3.0.1
(gfortran)

Kompilatory języka Fortran

g77 – kompilator języka Fortran 77 rozwijany w ramach projektu GCC od roku 1987 (aż do wersji GCC 3.4.4 opublikowanej 18 maja 2005 r.). Kompilator ten dostępny jest na systemy operacyjne Unix/Linux, Windows i inne.

gfortran – kompilator języka Fortran 77/95/2003/2008 stanowiący bezpośrednią kontynuację projektu g77. Zmiana nazwy z g77 na gfortran obowiązuje w projekcie GCC od wersji 4.0.1, wydanej oficjalnie 7 lipca 2005. Obecnie kompilator gfortranu jest wiodącym projektem Wolnego Oprogramowania związanym z językiem Fortran. Kompilator ten dostępny jest dla systemów operacyjnych Unix/Linux, Windows i inne.

g95 – kompilator języka Fortran 77 oraz 90/95 rozwijany niezależnie od projektu GCC. Projekt g95 powstał w roku 2000 – obecnie słabo rozwijany.

Kompilatory języka Fortran

W systemach Windows kompilator gfortran dostępny jest poprzez:

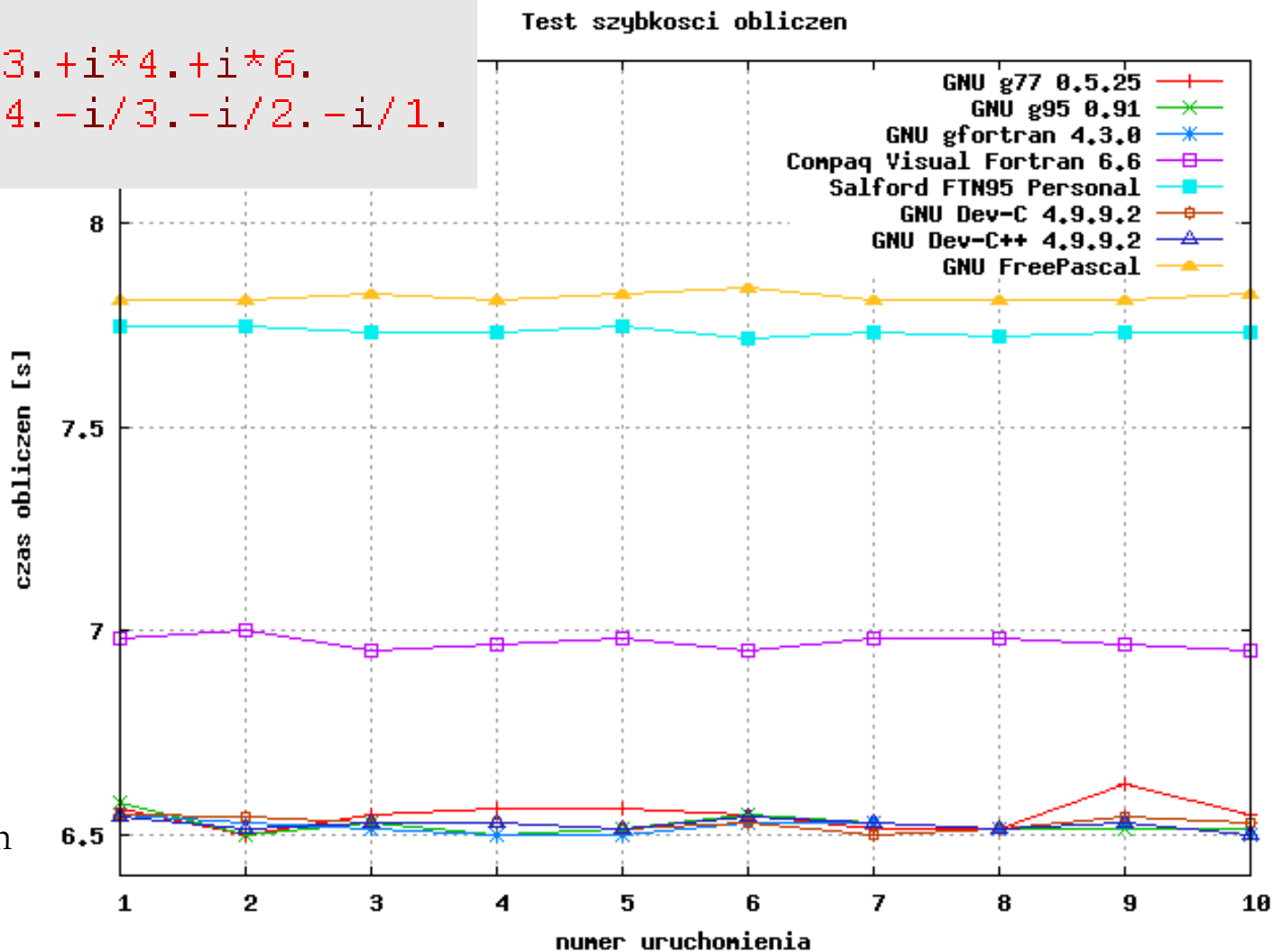
- **MinGW** (Minimalist GNU for Windows) – port GCC (GNU Compiler Collection), dostarczający darmowe i otwarte środowisko oraz narzędzia pozwalające na kompilację natywnych plików wykonywalnych dla platformy Windows. Programy skompilowane z użyciem MinGW do komunikacji z systemem operacyjnym wykorzystują bezpośrednio standardowe biblioteki dynamiczne systemu Windows i nie używają emulacji standardu POSIX (standardu systemu UNIX).
- **Cygwin** – kolekcja narzędzi wolnego i otwartego oprogramowania, które udostępniają programom działającym pod systemem Windows funkcjonalność przypominającą system Linux, a także biblioteka dla systemu Windows standardu POSIX. Cygwin jest rodzajem emulatora systemów Unix/Linux w systemie Windows.

Kompilatory języka Fortran

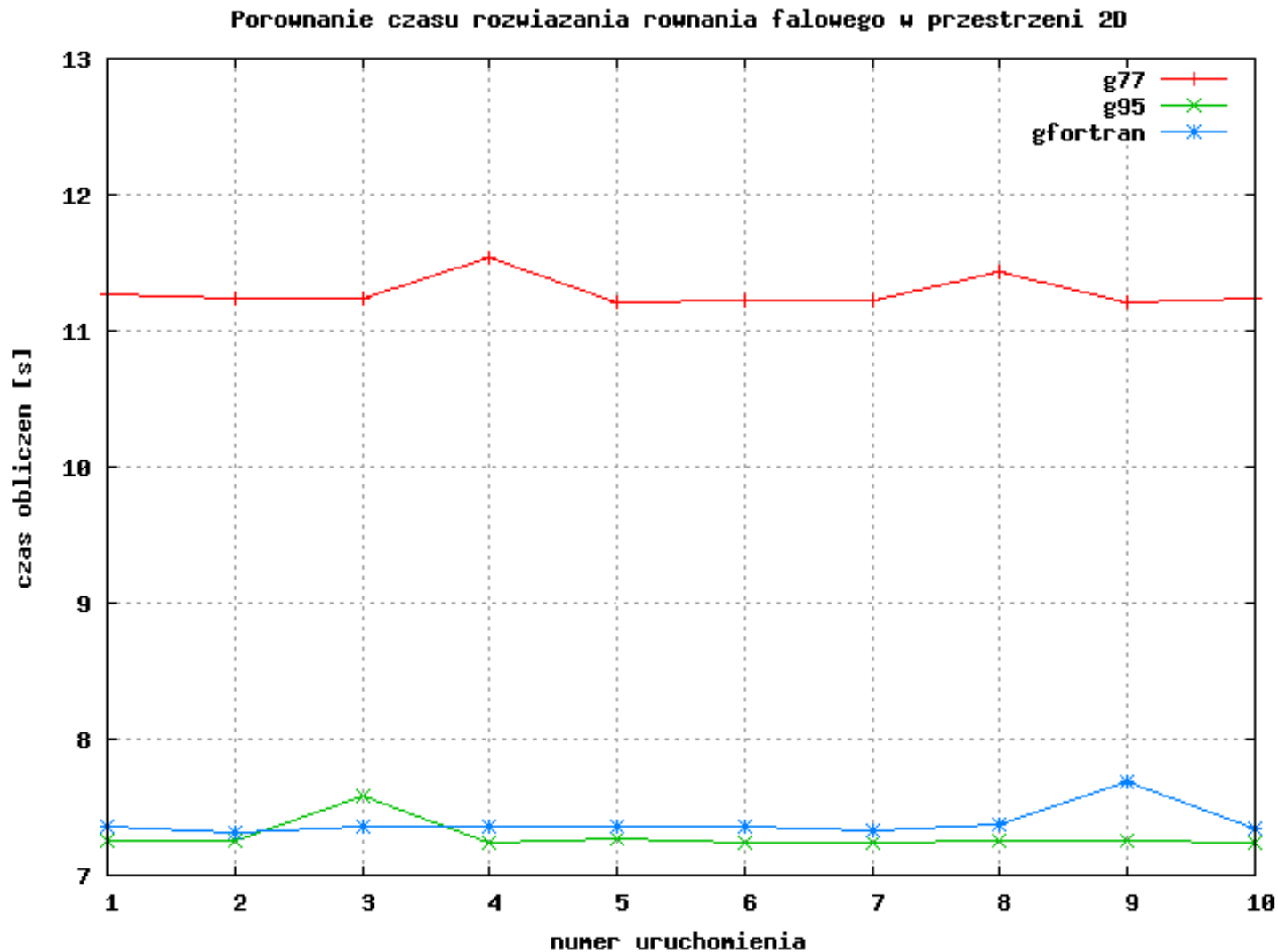
```
do i=1,100000000
  x=i*1.+i*2.+i*3.+i*4.+i*6.
  y=i/6.-i/5.-i/4.-i/3.-i/2.-i/1.
end do
```



porównanie czasu wykonywania się podstawowych operacji matematycznych w programach skompilowanych różnymi kompilatorami



Kompilatory języka Fortran



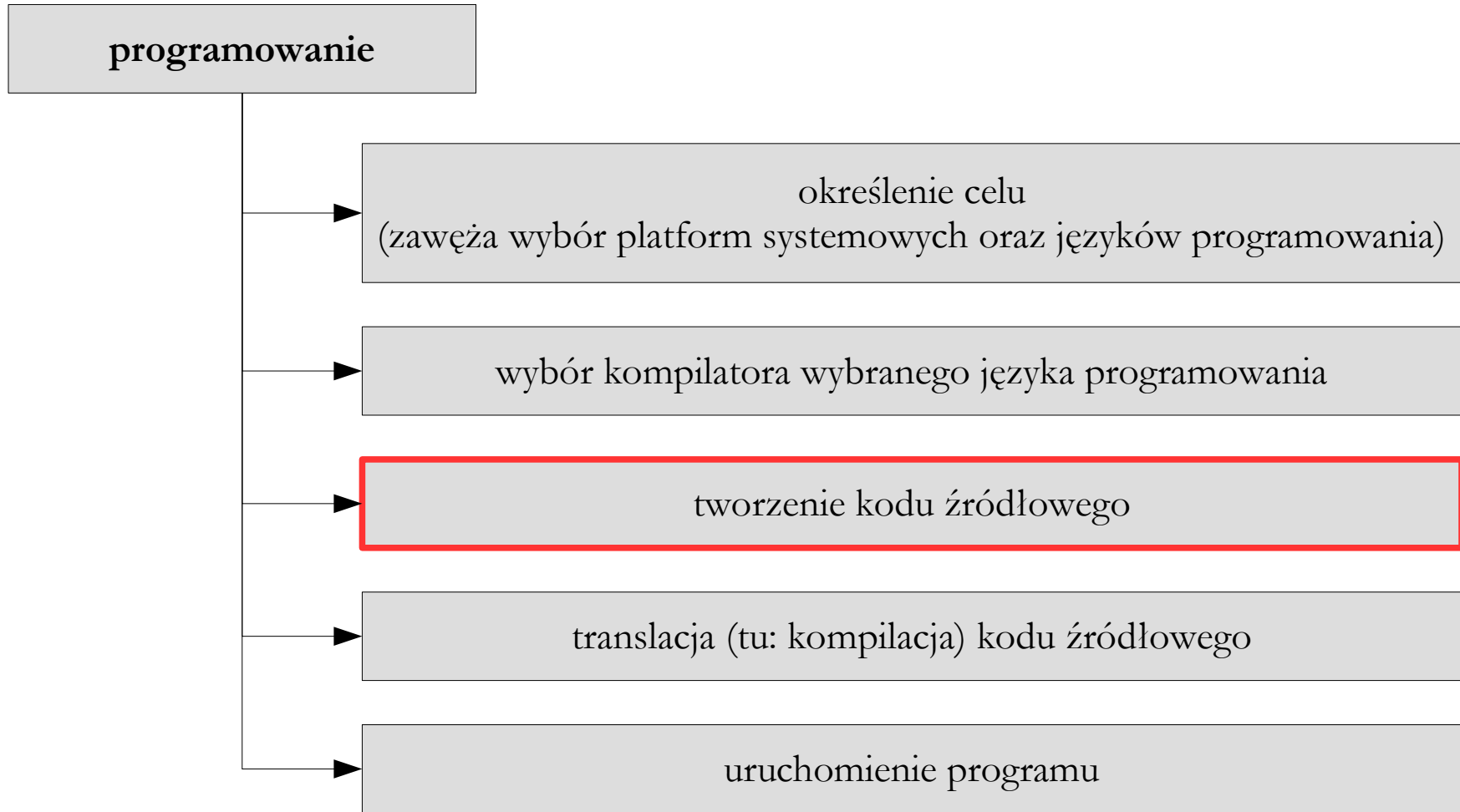
porównanie czasu działania
przykładowego programu
skompilowanego różnymi
kompilatorami GNU
Fortranu

Kompilatory języka Fortran

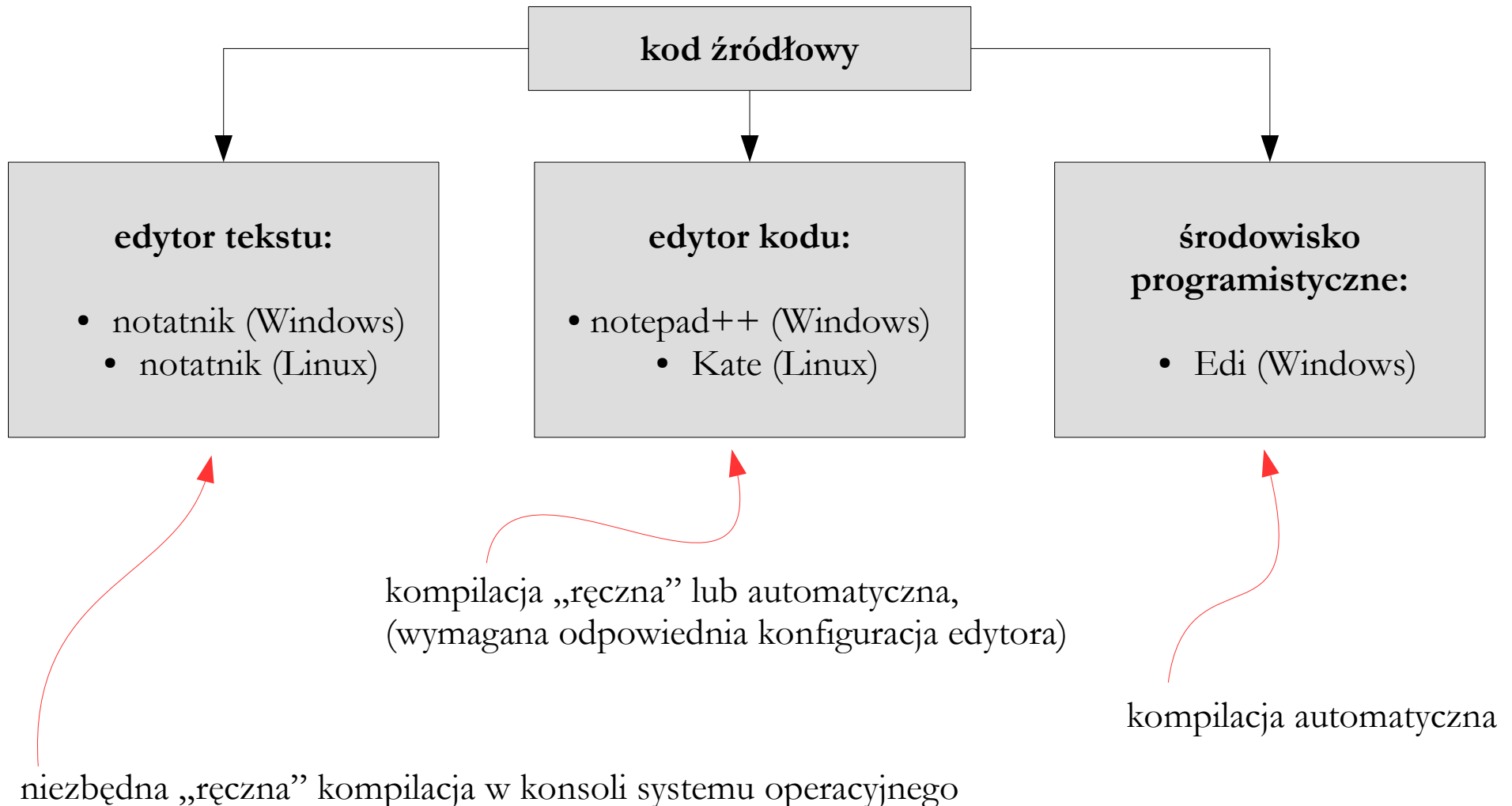
Zalety kompilatora gfortran:

- brak kosztów
- brak problemu aktualizacji
- brak problemu piractwa komputerowego
- łatwy dostęp
- dostępność darmowych środowisk programistycznych i edytorów kodu
- duża stabilność i bezpieczeństwo
- szybkość działania
- dostępność w wielu systemach operacyjnych
- duża liczba instrukcji dodatkowych
(szczególnie do komunikacji z systemem operacyjnym)

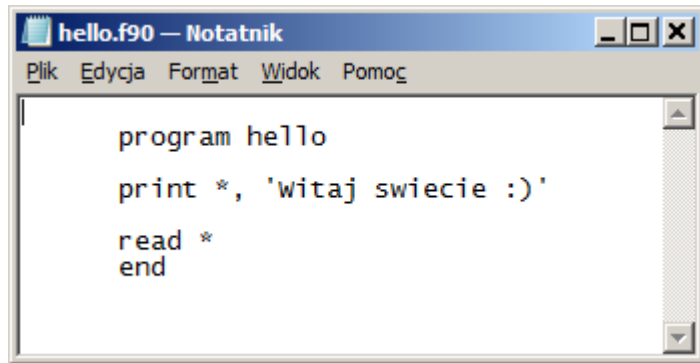
Ścieżka działań



Sposoby edycji kodu źródłowego



Sposoby edycji kodu źródłowego

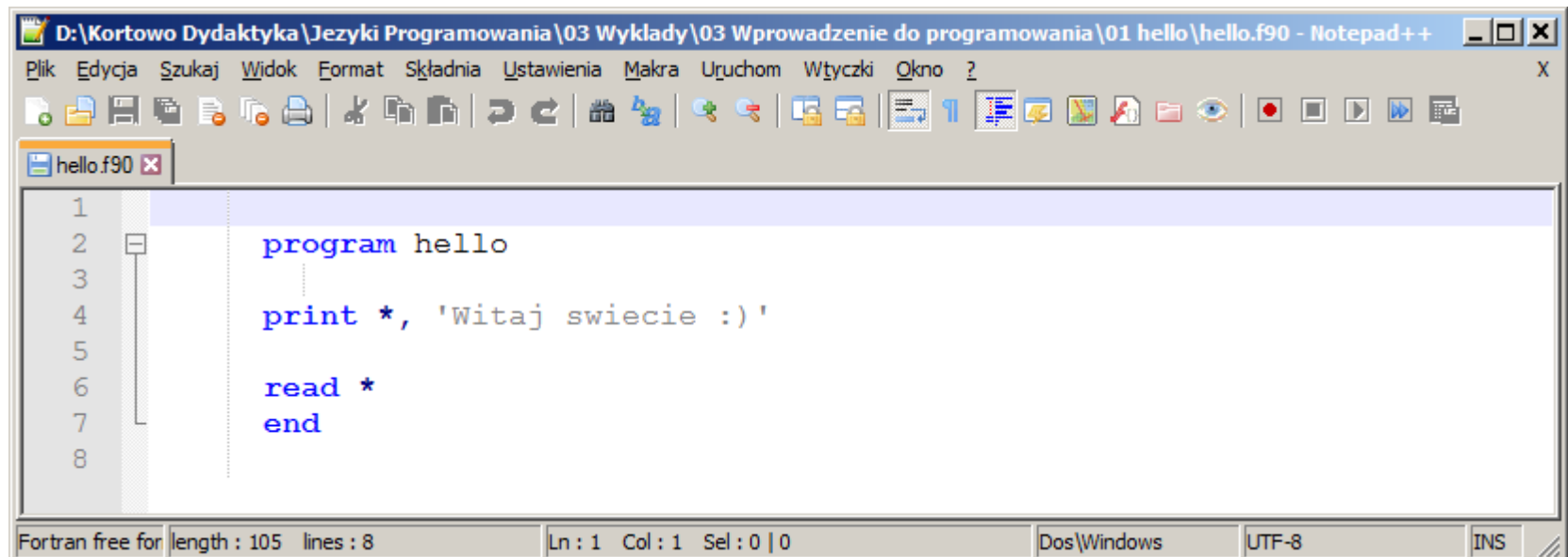


A screenshot of a Notepad window titled "hello.f90 - Notatnik". The window contains the following Fortran code:

```
program hello
print *, 'witaj swiecie :)'
read *
end
```

Windows: tworzenie kodu źródłowego
w edytorze tekstu (Notatnik)

Windows: tworzenie kodu źródłowego
w edytorze kodu (Notepad++)



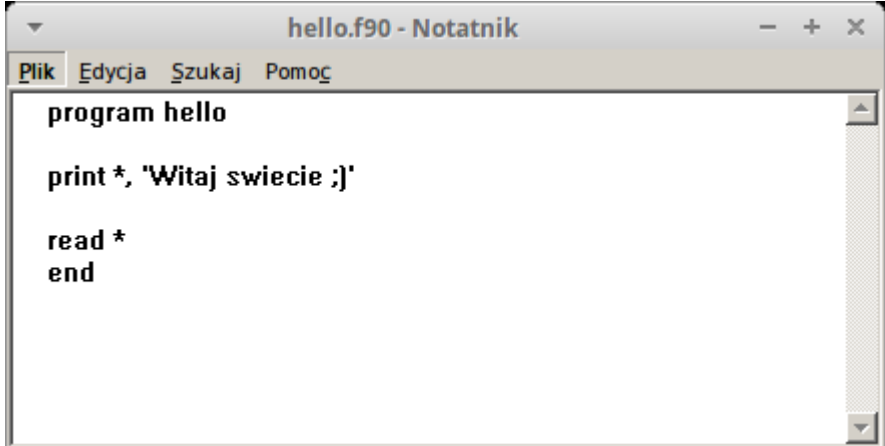
A screenshot of a Notepad++ window titled "D:\Kortowo Dydaktyka\Języki Programowania\03 Wykłady\03 Wprowadzenie do programowania\01 hello\hello.f90 - Notepad++". The window contains the following Fortran code:

```
1 program hello
2
3
4 print *, 'Witaj swiecie :)'
5
6 read *
7 end
8
```

The status bar at the bottom shows: Fortran free for length : 105 lines : 8 Ln : 1 Col : 1 Sel : 0 | 0 Dos\Windows UTF-8 INS

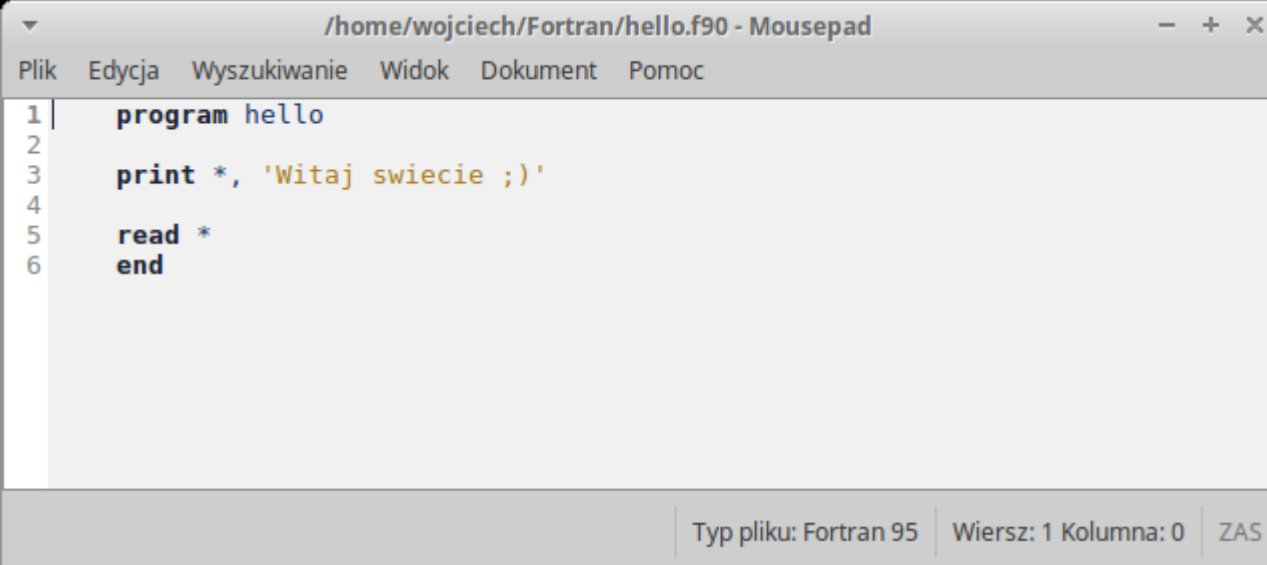
Sposoby edycji kodu źródłowego

Linux: tworzenie kodu źródłowego
w edytorze tekstu (Notatnik)



A screenshot of the Notatnik (Notepad) text editor window. The title bar reads "hello.f90 - Notatnik". The menu bar includes "Plik", "Edycja", "Szukaj", and "Pomoc". The main text area contains the following Fortran code:

```
program hello  
  
print *, 'Witaj swiecie ;)'  
  
read *  
end
```



A screenshot of the Mousepad code editor window. The title bar reads "/home/wojciech/Fortran/hello.f90 - Mousepad". The menu bar includes "Plik", "Edycja", "Wyszukiwanie", "Widok", "Dokument", and "Pomoc". The main text area shows the same Fortran code as the Notatnik window, but with line numbers 1 through 6 on the left side:

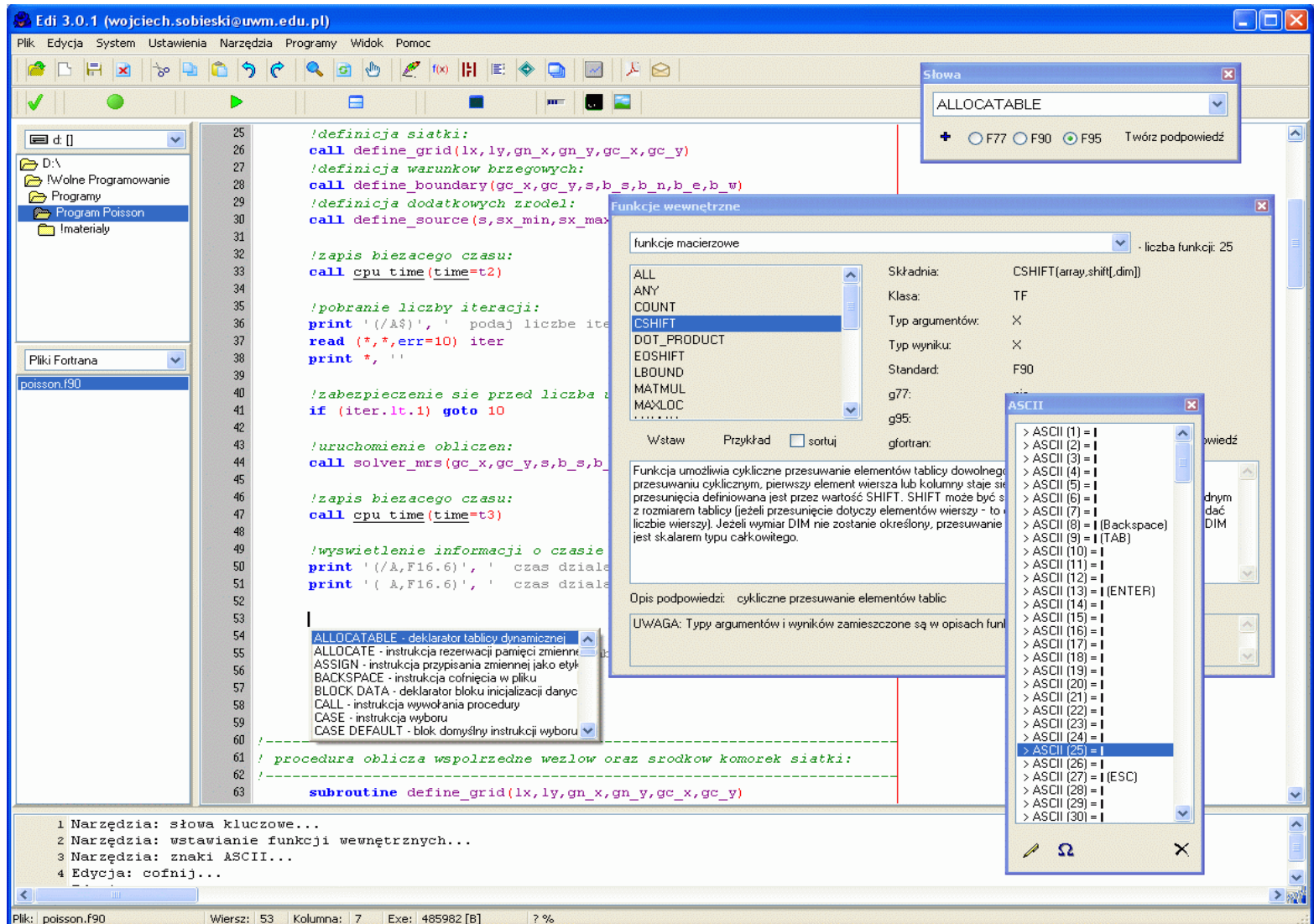
```
1 | program hello  
2 |  
3 | print *, 'Witaj swiecie ;)'  
4 |  
5 | read *  
6 | end
```

At the bottom of the window, a status bar displays: "Typ pliku: Fortran 95 | Wiersz: 1 Kolumna: 0 | ZAS".

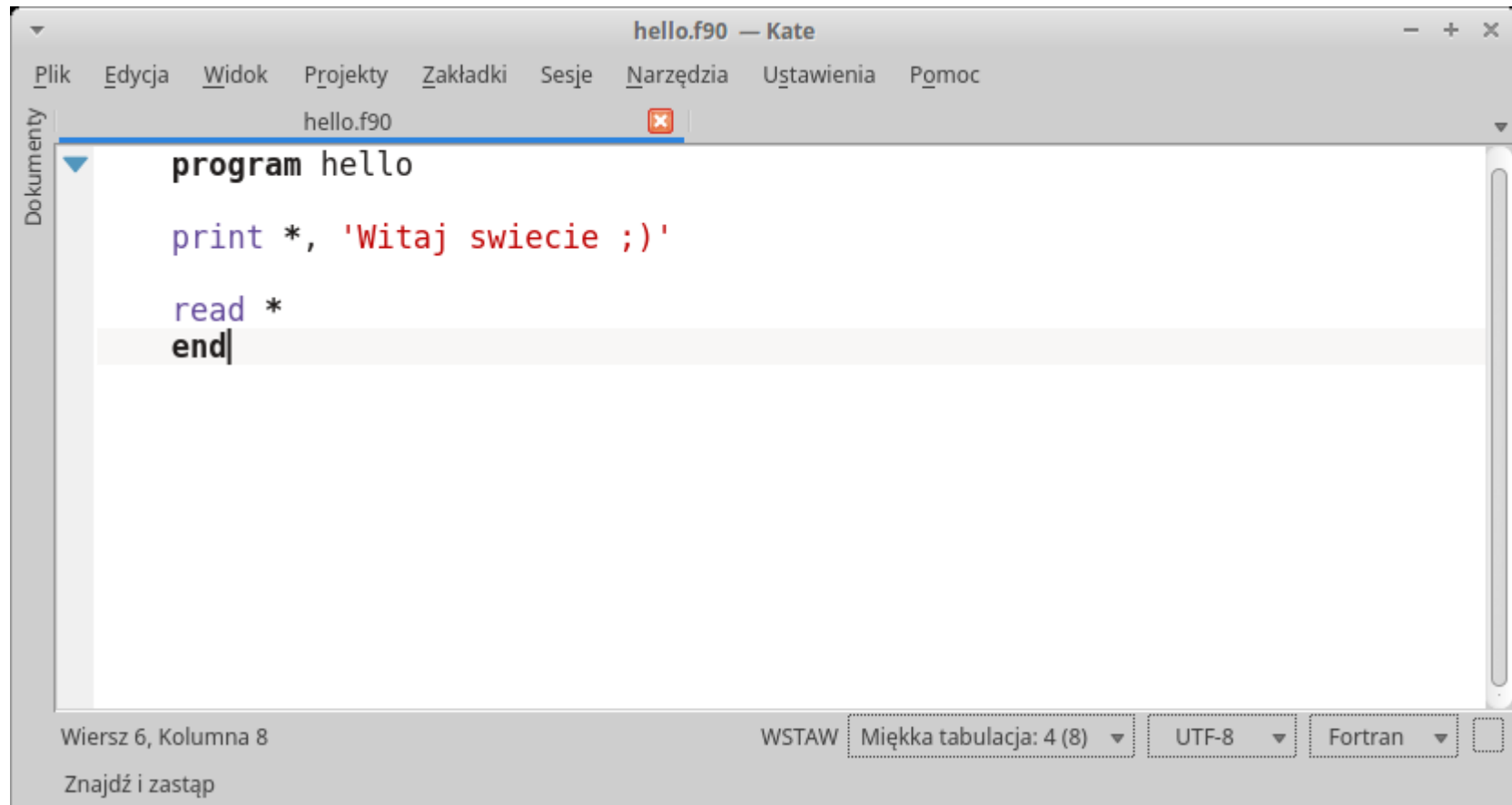
Linux: tworzenie kodu
źródłowego w edytorze
kodu (Mousepad)

Sposoby edycji kodu źródłowego

Windows:
tworzenie
kodu źródłowego
w środowisku
programistycznym
(Edi)



Sposoby edycji kodu źródłowego



```
program hello
  print *, 'Witaj swiecie ;)'
  read *
end
```

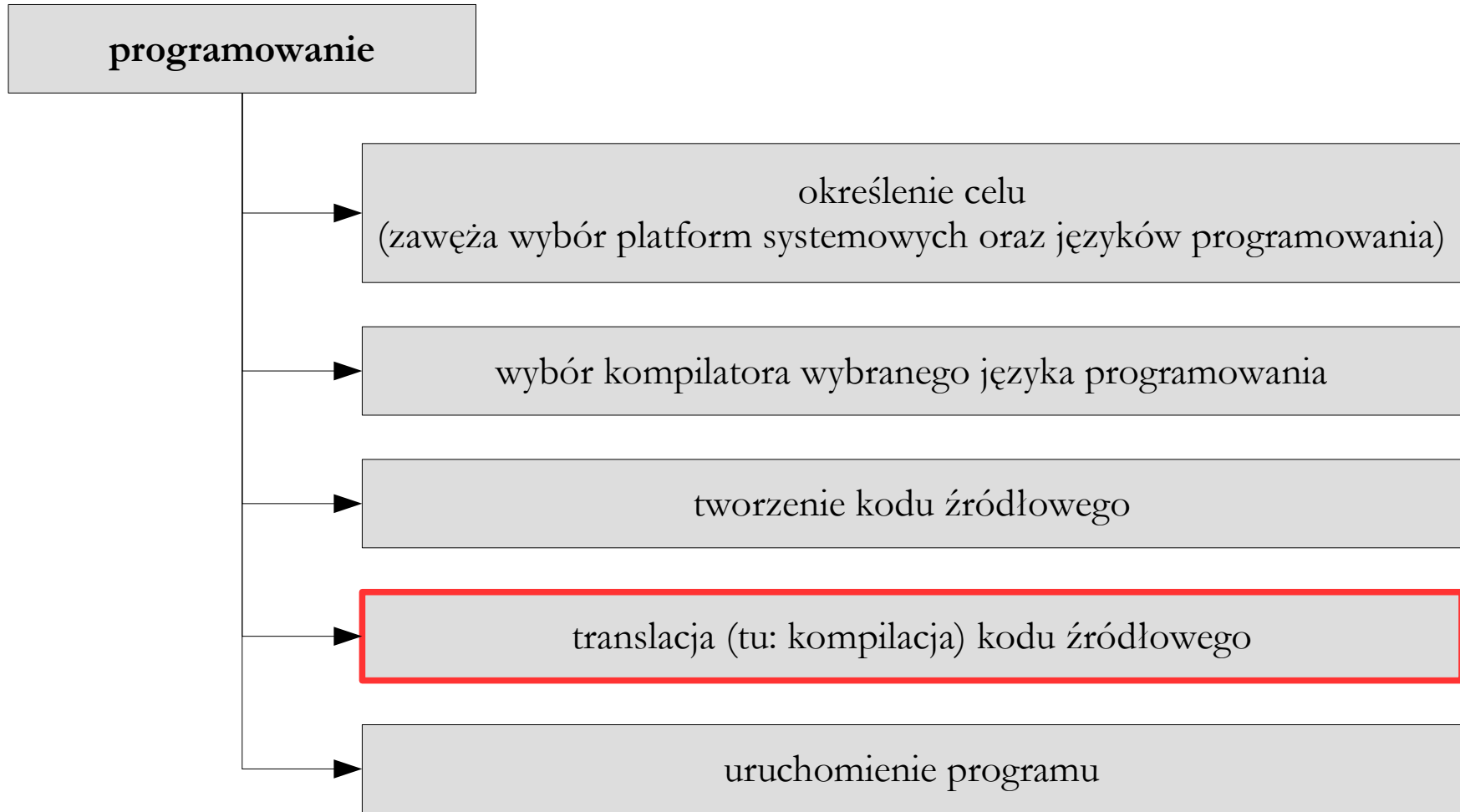
Wiersz 6, Kolumna 8

Znajdź i zastąp

WSTAW Mięka tabulacja: 4 (8) UTF-8 Fortran

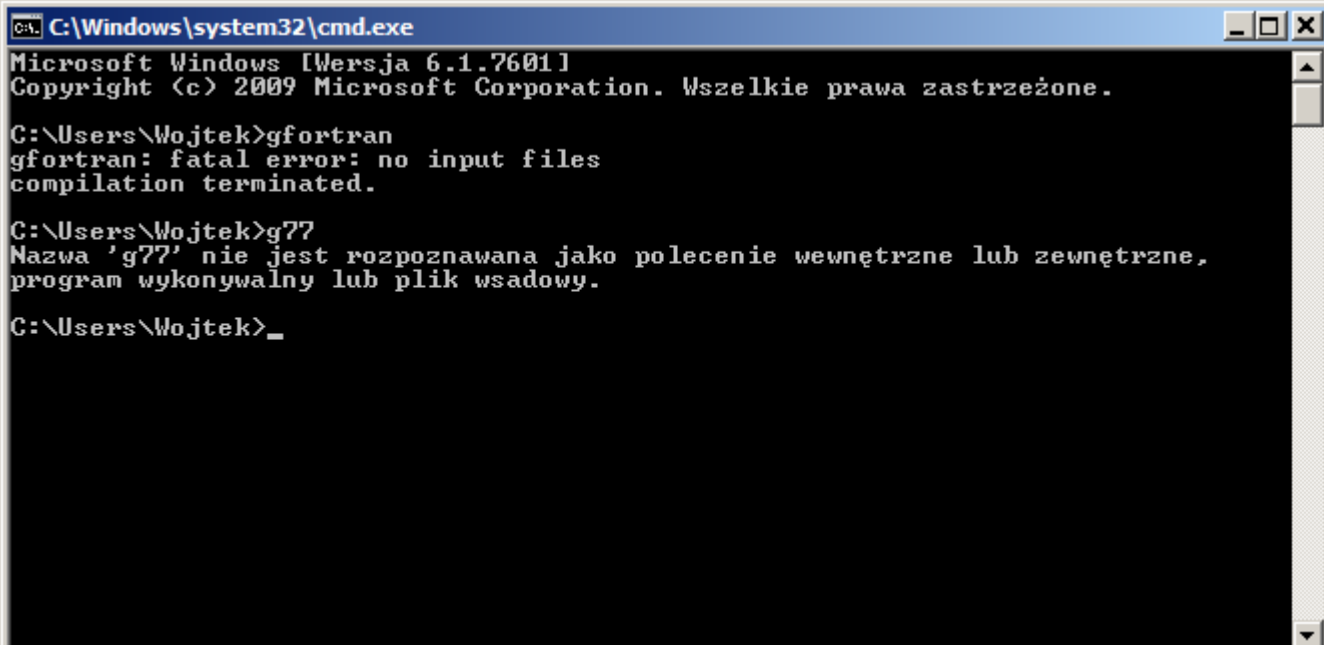
Linux: tworzenie kodu źródłowego w edytorze kodu (Kate)

Ścieżka działań



Zmienne środowiskowe

Każdy system operacyjny posiada konkretny zbiór poleceń możliwych do wywołania z dowolnego miejsca w strukturze katalogów. Przykładem może być polecenie DIR w systemie DOS lub Windows, lub też polecenie LS w systemach Unix/Linux – oba wyświetlają bieżącą listę plików i katalogów. Użycie polecenia spoza tego zbioru spowoduje wyświetlenie komunikatu błędu: „Nazwa 'polecenie' nie jest rozpoznawalna jako polecenie wewnętrzne lub zewnętrzne, program wykonywalny lub plik wsadowy.”



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Wojtek>gfortran
gfortran: fatal error: no input files
compilation terminated.

C:\Users\Wojtek>g77
Nazwa 'g77' nie jest rozpoznawana jako polecenie wewnętrzne lub zewnętrzne,
program wykonywalny lub plik wsadowy.

C:\Users\Wojtek>_
```

Zmienne środowiskowe

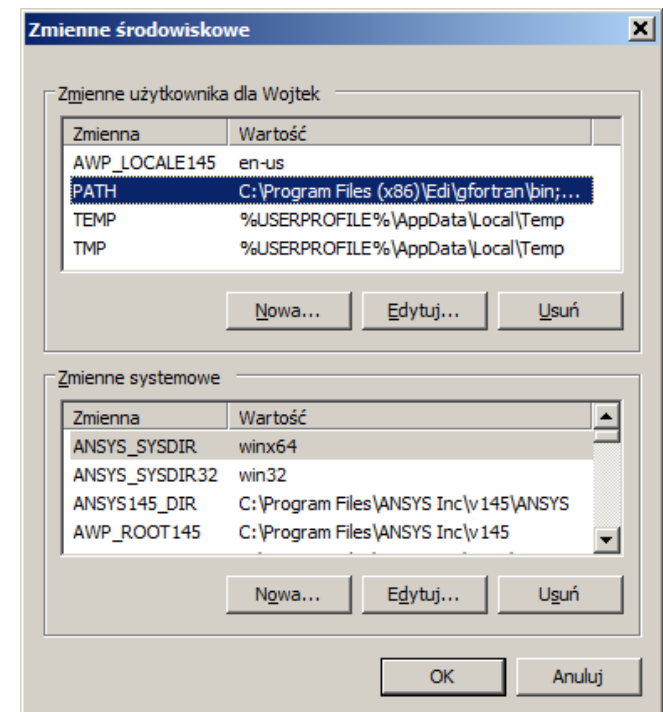
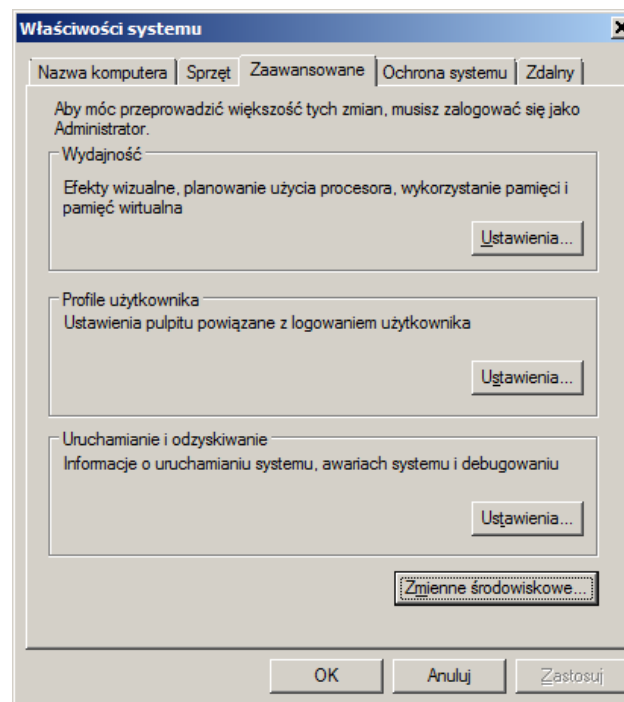
Generalnie istnieją trzy możliwości reakcji systemu operacyjnego na podanie dowolnej komendy:

- jeżeli polecenie jest poleceniem systemowym, system operacyjny wykona je bezpośrednio
- jeżeli polecenie nie jest poleceniem systemowym, system operacyjny przeszuka wszystkie katalogi znajdujące się w specjalnej zmiennej środowiskowej (najczęściej o nazwie **PATH**) – jeżeli znajdzie tam plik wykonywalny lub plik wsadowy powłoki, to go uruchomi
- jeżeli polecenie nie jest poleceniem systemowym, a w katalogach znajdujących się w zmiennej **PATH** nie ma pliku wykonywalnego lub skryptu powłoki o takiej nazwie jak polecenie, system operacyjny zgłosi błąd

Ta sama zasada dotyczy bibliotek, z tym że ich lokalizacja przechowywana jest w innej zmiennej środowiskowej, np. **LIBRARY_PATH**.

Zmienne środowiskowe

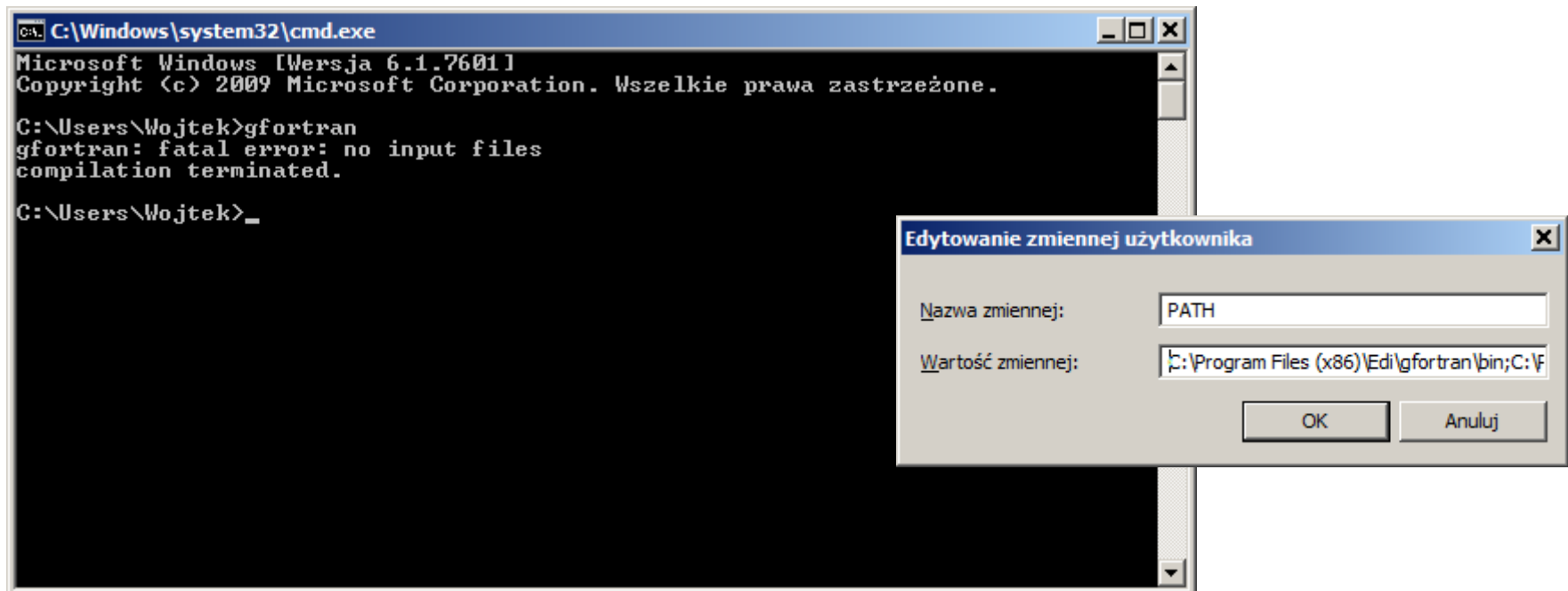
Zmienna środowiskowa – ogólnie dostępna zmienna w systemie operacyjnym, zawierająca informacje przydatne w pracy innych programów. Zmienne środowiskowe mogą być tworzone lub modyfikowane przez system operacyjny, przez instalowane w nim dodatkowe oprogramowanie lub też ręcznie przez użytkownika.



Zmienne środowiskowe

Przykładowa zawartość zmiennej **PATH** (Windows):

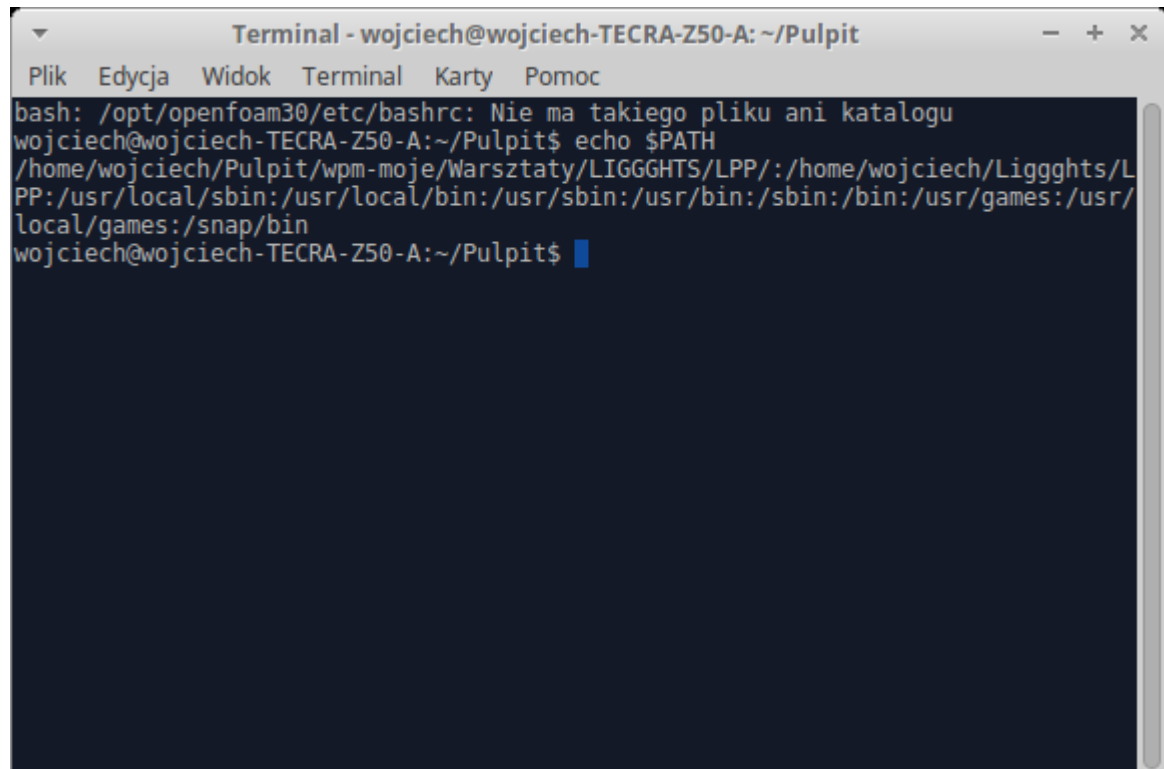
```
C:\Program Files (x86)\Edi\gfortran\bin;C:\Program Files (x86)\  
Microsoft Visual Studio 11.0\VC\bin\x86_amd64;C:\Program Files (x86)\  
Microsoft Visual Studio 11.0\VC\include;C:\Program Files\ANSYS Inc\  
v145\fluent\ntbin\win64
```



Zmienne środowiskowe

Przykładowa zawartość zmiennej **PATH** (Linux):

```
/home/wojciech/Pulpit/wpm-moje/Warsztaty/LIGGGHTS/LPP:  
/home/wojciech/LIGGGHTS/LPP:/usr/local/sbin:/usr/local/bin:/usr/sb  
in:/usr/bin:/sbin:/bin:/snap/bin
```



```
Terminal - wojciech@wojciech-TECRA-Z50-A: ~/Pulpit  
Plik Edycja Widok Terminal Karty Pomoc  
bash: /opt/openfoam30/etc/bashrc: Nie ma takiego pliku ani katalogu  
wojciech@wojciech-TECRA-Z50-A:~/Pulpit$ echo $PATH  
/home/wojciech/Pulpit/wpm-moje/Warsztaty/LIGGGHTS/LPP:/home/wojciech/Liggghts/L  
PP:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/  
local/games:/snap/bin  
wojciech@wojciech-TECRA-Z50-A:~/Pulpit$
```

Zmienne środowiskowe

Przykład definicji zmiennych systemowych w systemach Windows:

```
SET PATH=sciezka_1;sciezka_2;%PATH%
```

```
SET LIBRARY_PATH=sciezka_1;sciezka_2;%LIB%
```

Przykład definicji zmiennych systemowych w systemach Linux:

```
PATH = sciezka_1: sciezka_2: ...: $PATH
```

```
export PATH
```

Zmienne środowiskowe

Dokładna zawartość zmiennej `PATH` zależy od wybranego kompilatora oraz użytego środowiska programistycznego.

Przykłady poprawnych wpisów do zmiennej **PATH** (Windows):

```
SET PATH=C:\F90\BIN;"%PATH%"
```

```
SET INCLUDE=C:\F90\INCLUDE;"%INCLUDE%"
```

```
SET LIB=C:\F90\LIB;"%LIB%"
```

} MS Fortran VS

```
SET PATH=%PATH%;C:\VFort\Bin
```

```
SET LIBRARY_PATH=C:\VFort\Lib
```

} VFort

```
SET PATH=C:\DELPHI\BIN;C:\DELPHI\PROJECTS\BPL;%PATH%
```

} Delphi

```
SET PATH=C:\FreePas\Bin\Win32;%PATH%
```

} Free Pascal

Zmienne środowiskowe

Zapytanie o zmienną systemową **PATH** w systemach Windows:

set	- wszystkie zmienne
set PATH	- tylko zmienna PATH

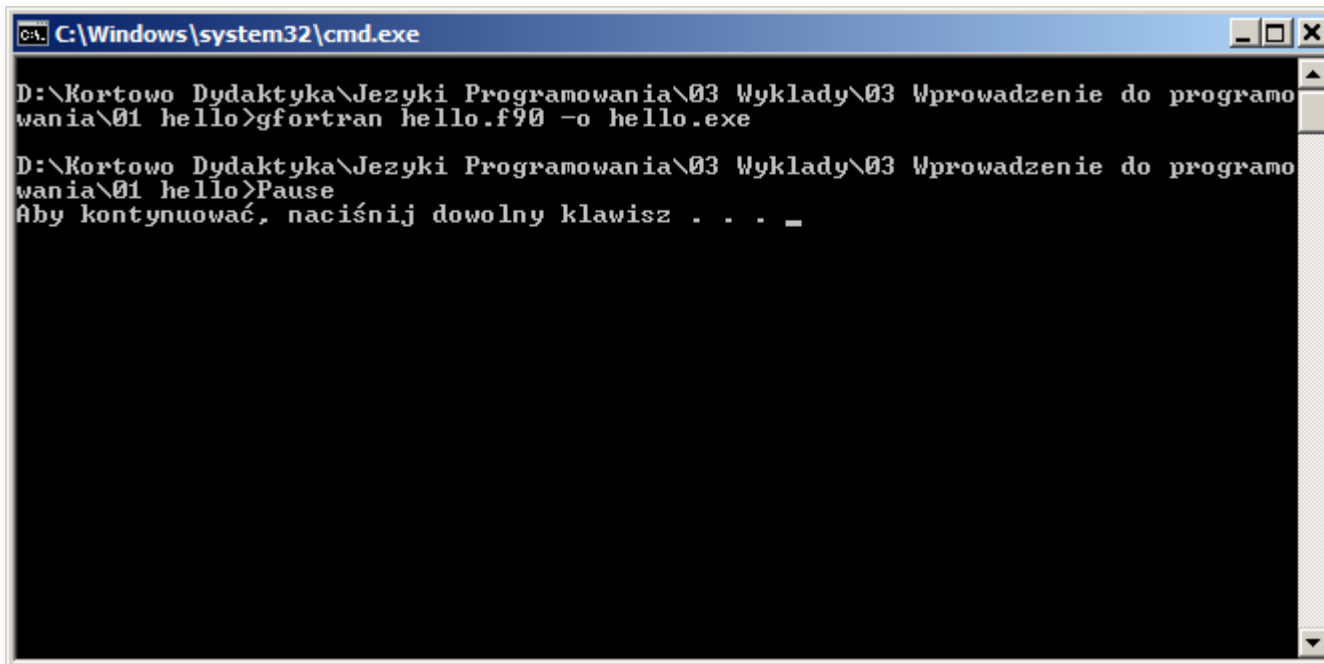
Zapytanie o zmienną systemową **PATH** w systemach Linux:

printenv	- wszystkie zmienne
echo \$PATH	- tylko zmienna PATH

Polecenie kompilacji

Składnia polecenia kompilacji kompilatorem gfortran (Windows):

```
gfortran nazwa.f90 -o nazwa.exe
```

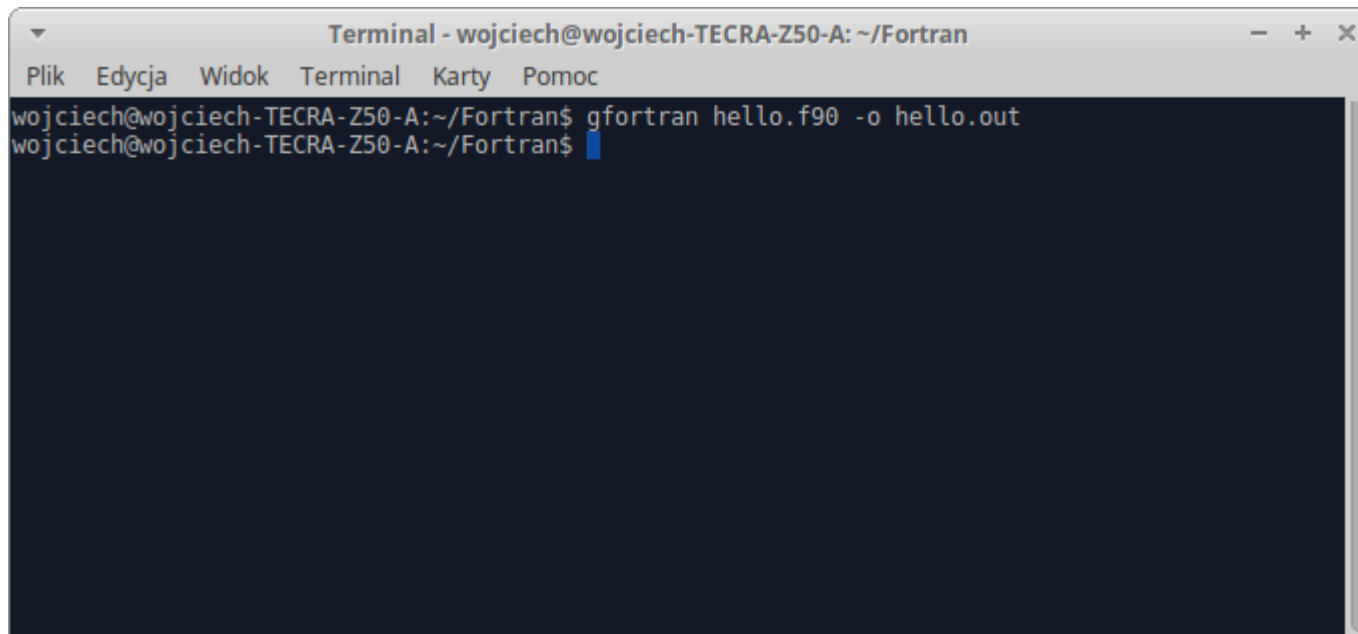


```
C:\Windows\system32\cmd.exe
D:\Kortowo Dydaktyka\Języki Programowania\03 Wykłady\03 Wprowadzenie do programo
wania\01 hello>gfortran hello.f90 -o hello.exe
D:\Kortowo Dydaktyka\Języki Programowania\03 Wykłady\03 Wprowadzenie do programo
wania\01 hello>Pause
Aby kontynuować, naciśnij dowolny klawisz . . . _
```

Polecenie kompilacji

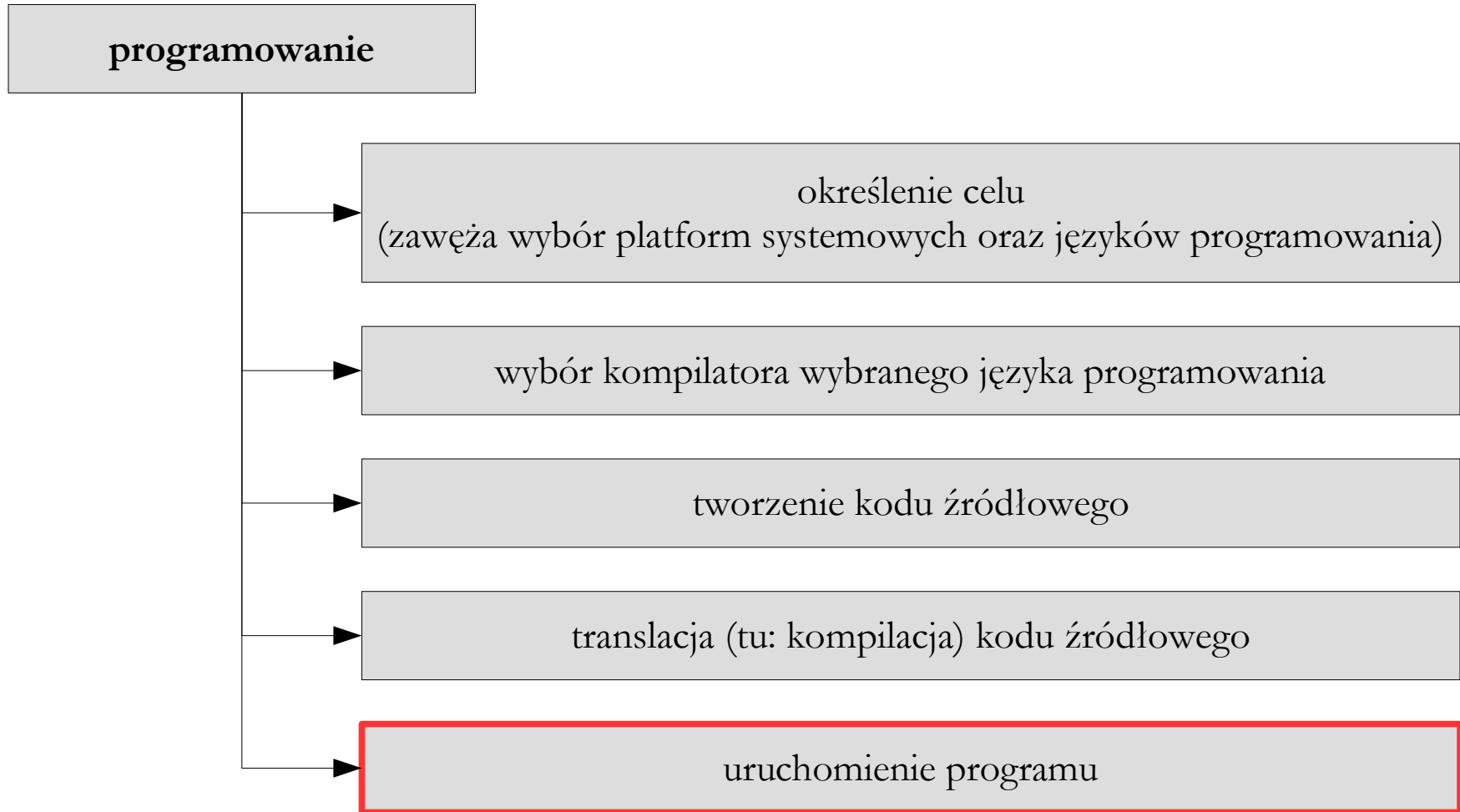
Składnia polecenia kompilacji kompilatorem gfortran (Linux):

```
gfortran nazwa.f90 -o nazwa.out
```



```
Terminal - wojciech@wojciech-TECRA-Z50-A: ~/Fortran
Plik  Edycja  Widok  Terminal  Karty  Pomoc
wojciech@wojciech-TECRA-Z50-A:~/Fortran$ gfortran hello.f90 -o hello.out
wojciech@wojciech-TECRA-Z50-A:~/Fortran$
```

Ścieżka działań



Uruchamianie programu

Uruchamianie skompilowanego programu w systemach Windows:

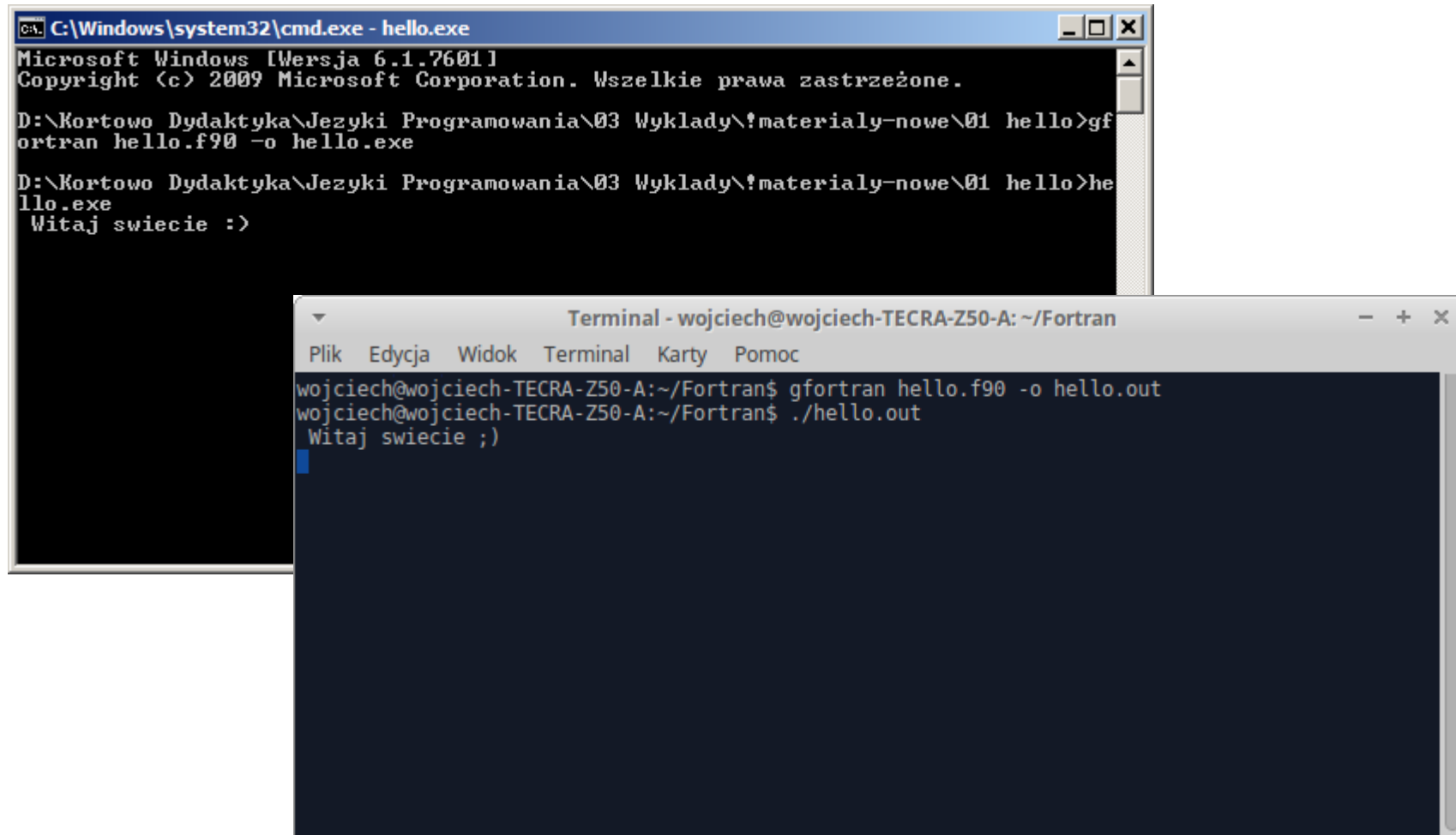
nazwa.exe

Uruchamianie skompilowanego programu w systemach Linux:

./nazwa.out

- w systemach UNIX/Linux istnieje rozróżnienie na małe i wielkie litery
- w systemach UNIX/Linux plik wykonywalny może mieć dowolne rozszerzenie, albo może nie mieć go wcale
- w systemach UNIX/Linux plik wykonywalny nie rozpocznie działania, jeżeli nie będzie miał przyznanych praw do wykonywania
- w systemach UNIX/Linux plik wykonywalny nie rozpocznie działania, jeżeli użytkownik nie będzie posiadał praw do uruchamiania programów w tym katalogu

Uruchamianie programu



The image displays two terminal windows side-by-side, illustrating the execution of a program in different operating systems. The top window is a Windows Command Prompt, and the bottom window is a Linux Terminal.

```
C:\Windows\system32\cmd.exe - hello.exe
Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

D:\Kortowo Dydaktyka\Języki Programowania\03 Wykłady\!materialy-nowe\01 hello>gfortran hello.f90 -o hello.exe

D:\Kortowo Dydaktyka\Języki Programowania\03 Wykłady\!materialy-nowe\01 hello>hello.exe
Witaj świecie :>
```

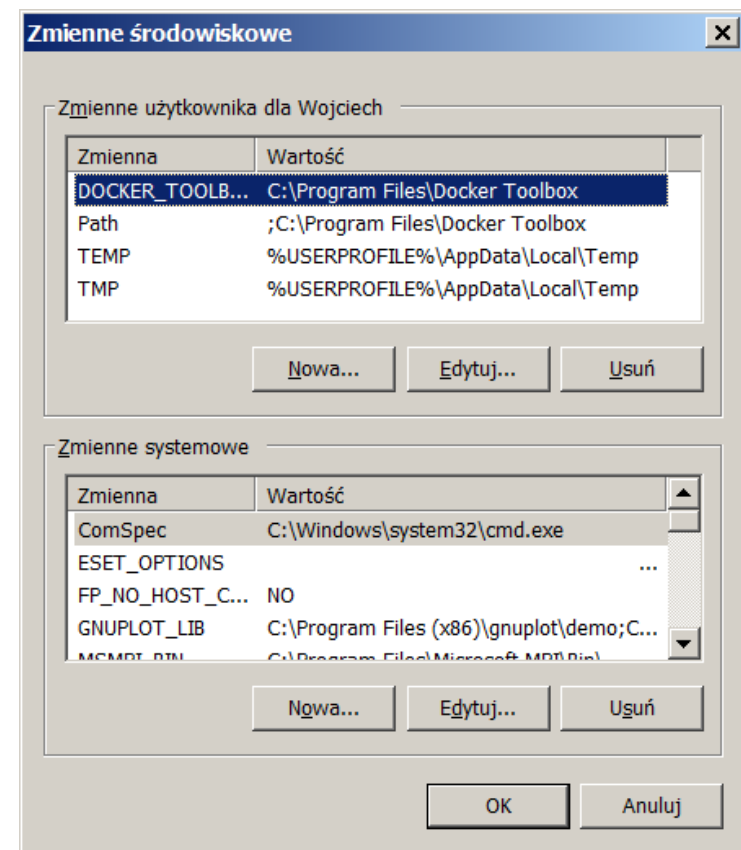
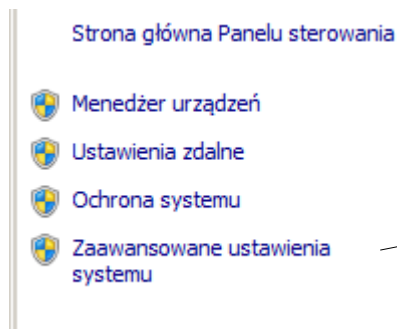
```
Terminal - wojciech@wojciech-TECRA-Z50-A: ~/Fortran
Plik Edycja Widok Terminal Karty Pomoc
wojciech@wojciech-TECRA-Z50-A:~/Fortran$ gfortran hello.f90 -o hello.out
wojciech@wojciech-TECRA-Z50-A:~/Fortran$ ./hello.out
Witaj świecie ;)
```

uruchomienie skompilowanego programu w systemie Windows i Linux

Załącznik

Samodzielna instalacja i konfiguracja narzędzi w środowisku Windows:

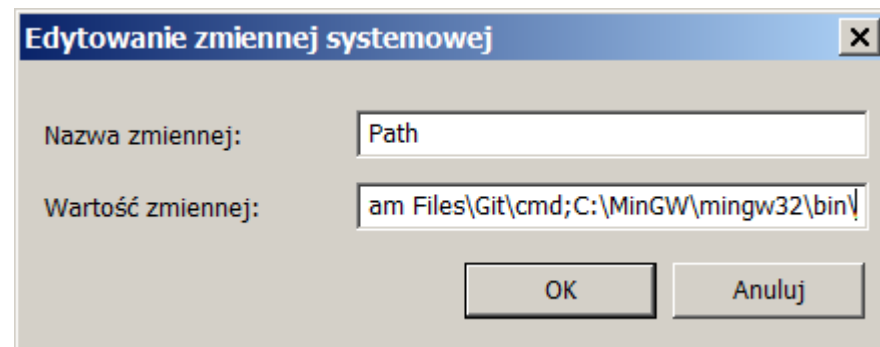
- Pobierz i zainstaluj pakiet **MinGW** (tu: **C:\MinGW**):
(<https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/>)
- Włącz zaawansowane funkcje systemu i uruchom okno zmiennych środowiskowych



Załącznik

Samodzielna instalacja i konfiguracja narzędzi w środowisku Windows:

- Wybierz zmienną PATH (w zmiennych użytkownika), wciśnij klawisz Edytuj i dopisz na końcu, po średniku, ścieżkę do katalogu pakietu MinGW (tu: **C:\MinGW\mingw32\bin**);
UWAGA: nie kasuj innych wpisów, bo coś w systemie operacyjnym może przestać działać



- Uruchom komputer ponownie
- Uruchom konsolę systemu (poleceniem **cmd**)
- Wpisz polecenie gfortran i zatwierdź je klawiszem Enter – jeśli pojawi się komunikat

```
gfortran: fatal error: no input files
```

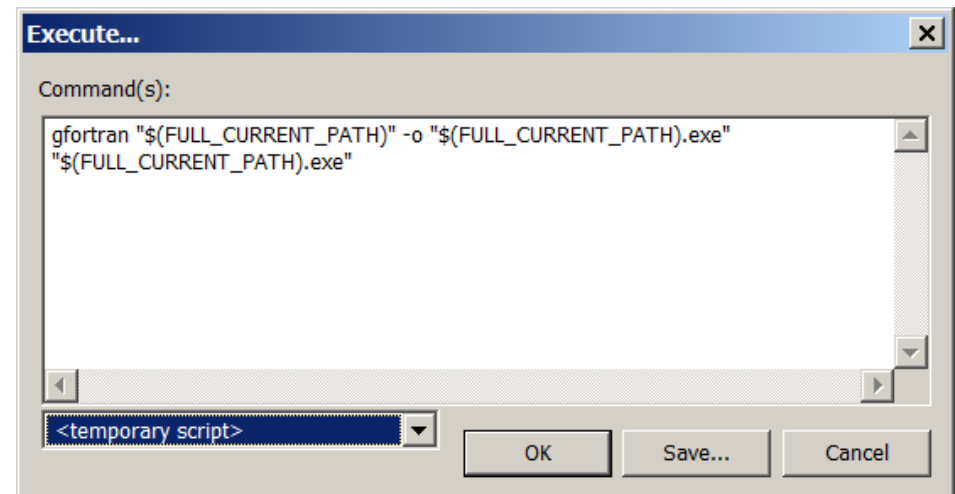
to znaczy, że wszystko jest dobrze: kompilator działa, ale nie znajduje żadnego pliku

Załącznik

Samodzielna instalacja i konfiguracja narzędzi w środowisku Windows:

- Pobierz i zainstaluj program **Notepad++**:
(<https://notepad-plus-plus.org/download/v7.html>)
- Pobierz wtyczkę **NppExec** do programu Notepad++:
(<https://sourceforge.net/projects/npp-plugins/files/NppExec/>)
- Rozpakuj wtyczkę i skopiuj wszystkie katalogi i pliki do katalogu plugin programu Notepad++
- Uruchom program Notepad++ i wciśnij klawisz F6
- W oknie, które się pojawi, wpisz polecenia:

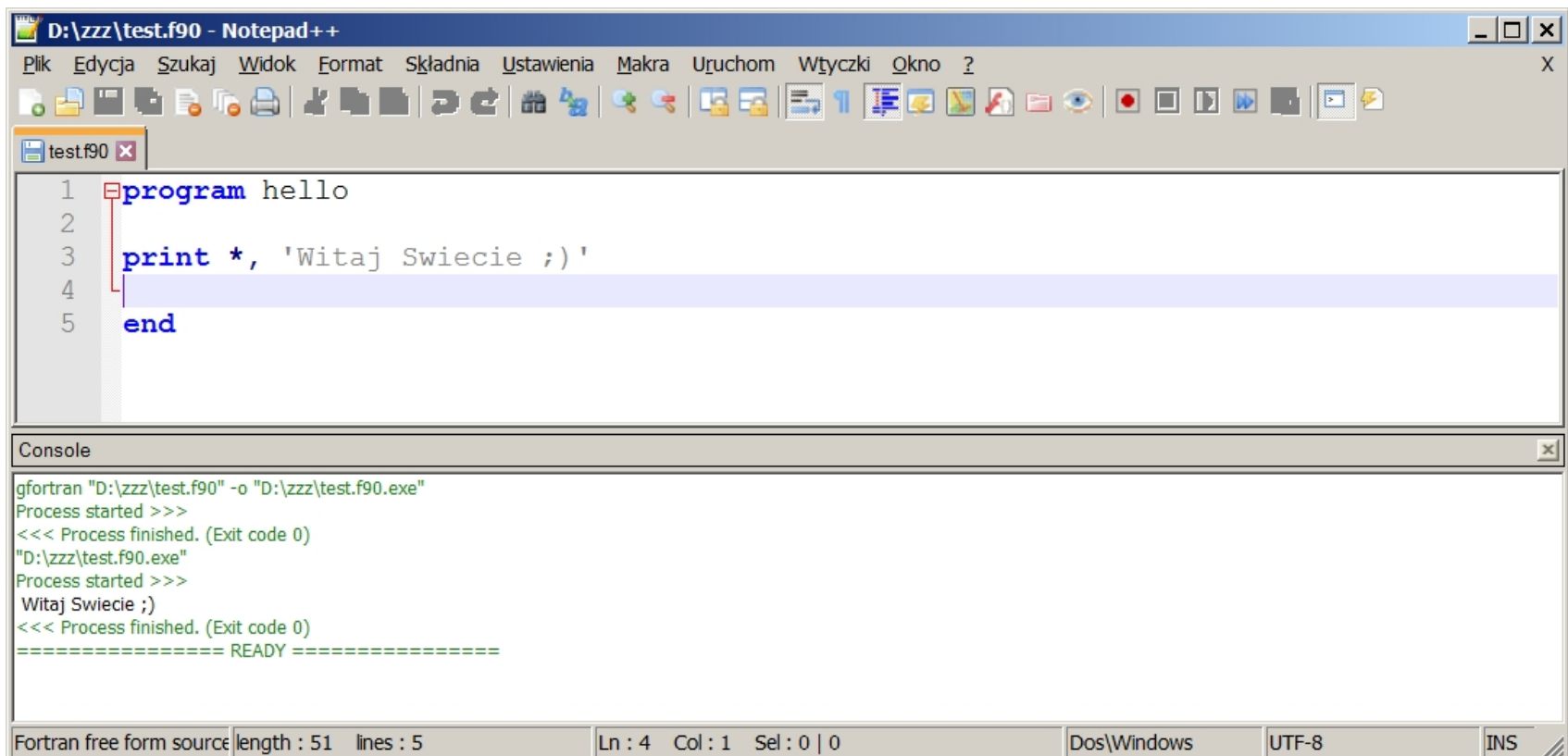
```
gfortran "$(FULL_CURRENT_PATH)" -o "$(FULL_CURRENT_PATH).exe"  
"$(FULL_CURRENT_PATH).exe"
```



Załącznik

Samodzielna instalacja i konfiguracja narzędzi w środowisku Windows:

- Stwórz albo otwórz plik źródłowy (plik źródłowy musi być zapisany na dysku)
- Wciśnij klawisz F6 a następnie klawisz OK – program wykona się w dolnym oknie programu Notepad++ (a nie w oknie konsoli systemu)



The screenshot shows the Notepad++ application window titled "D:\zZZ\test.f90 - Notepad++". The menu bar includes "Plik", "Edycja", "Szukaj", "Widok", "Format", "Składnia", "Ustawienia", "Makra", "Uruchom", "Wtyczki", "Okno", and "?". The toolbar contains various icons for file operations and editing. The main text area shows the following Fortran code:

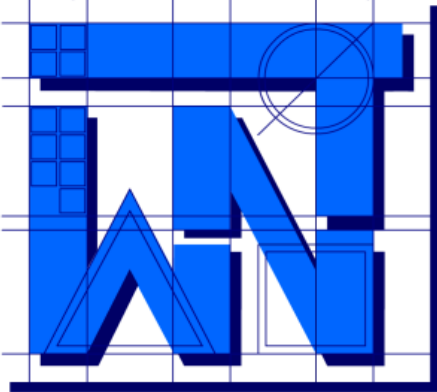
```
1 program hello
2
3   print *, 'Witaj Swiecie ;)'
4
5 end
```

The "Console" window at the bottom displays the execution output:

```
gfortran "D:\zZZ\test.f90" -o "D:\zZZ\test.f90.exe"
Process started >>>
<<< Process finished. (Exit code 0)
"D:\zZZ\test.f90.exe"
Process started >>>
Witaj Swiecie ;)
<<< Process finished. (Exit code 0)
===== READY =====
```

The status bar at the bottom indicates "Fortran free form source", "length : 51", "lines : 5", "Ln : 4", "Col : 1", "Sel : 0 | 0", "Dos\Windows", "UTF-8", and "INS".

Wydział Nauk Technicznych



UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN
The Faculty of Technical Sciences
POLAND, 10-957 Olsztyn, M. Oczapowskiego 11
tel.: (48)(89) 5-23-32-40, fax: (48)(89) 5-23-32-55
URL: <http://www.uwm.edu.pl/edu/sobieski/> (in Polish)



Dziękuję

Wojciech Sobieski

Olsztyn, 2001-2021