

Wojciech Myszka

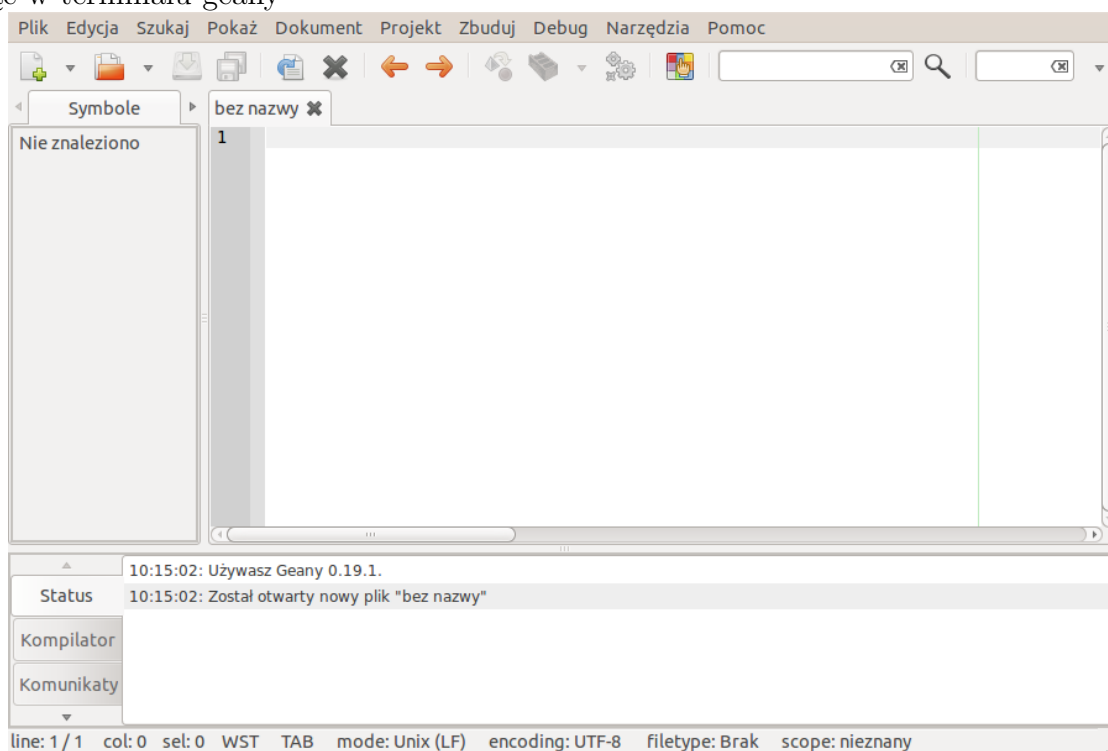
# Geany

2016-05-07 08:55:45 +0200

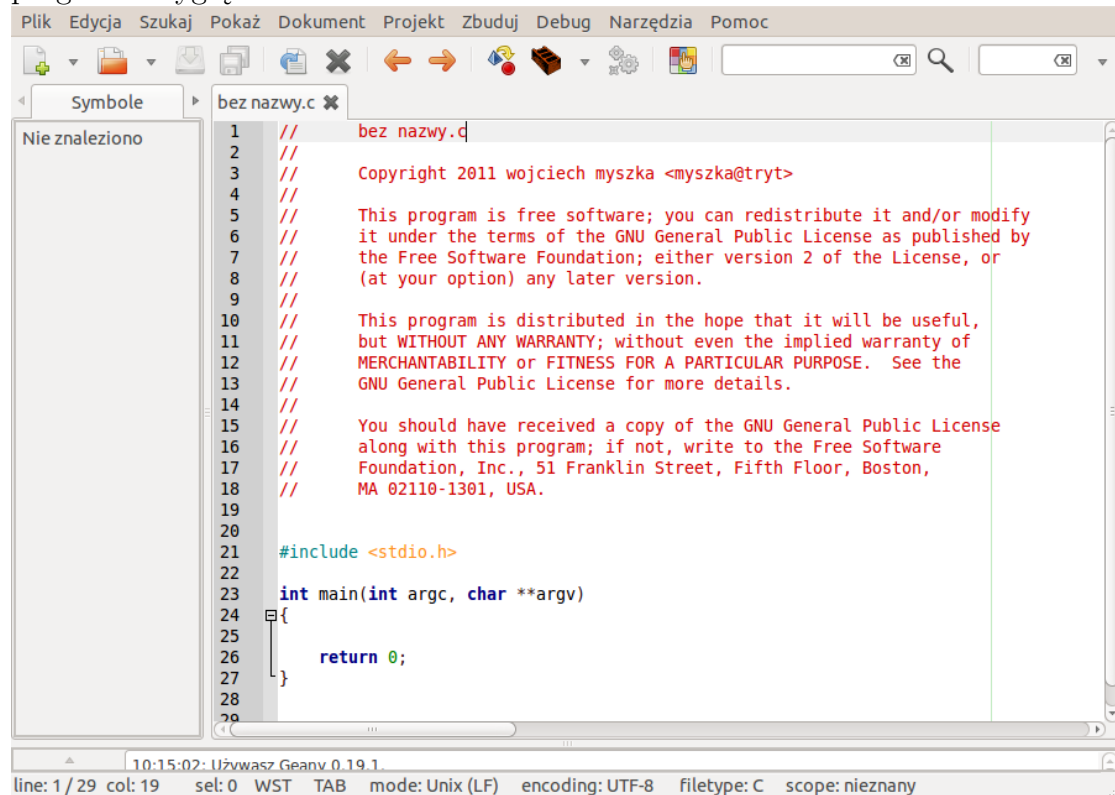
To chyba najprostsze środowisko. Można z niego korzystać do tworzenia większych projektów, ale można też używać do budowy programów „ad-hoc”.

## 1. Program ad hoc

Uruchamiamy program albo wybierając odpowiednią pozycję w menu albo pisząc w terminalu geany



Następnie z menu wybieramy Plik → Nowy (z szablonu) a tam main.c. Okno programu wygląda teraz tak:



```
1 // bez nazwy.c
2 //
3 // Copyright 2011 wojciech myszka <myszka@tryt>
4 //
5 // This program is free software; you can redistribute it and/or modify
6 // it under the terms of the GNU General Public License as published by
7 // the Free Software Foundation; either version 2 of the License, or
8 // (at your option) any later version.
9 //
10 // This program is distributed in the hope that it will be useful,
11 // but WITHOUT ANY WARRANTY; without even the implied warranty of
12 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 // GNU General Public License for more details.
14 //
15 // You should have received a copy of the GNU General Public License
16 // along with this program; if not, write to the Free Software
17 // Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
18 // MA 02110-1301, USA.
19 //
20
21 #include <stdio.h>
22
23 int main(int argc, char **argv)
24 {
25
26     return 0;
27 }
28
29
```

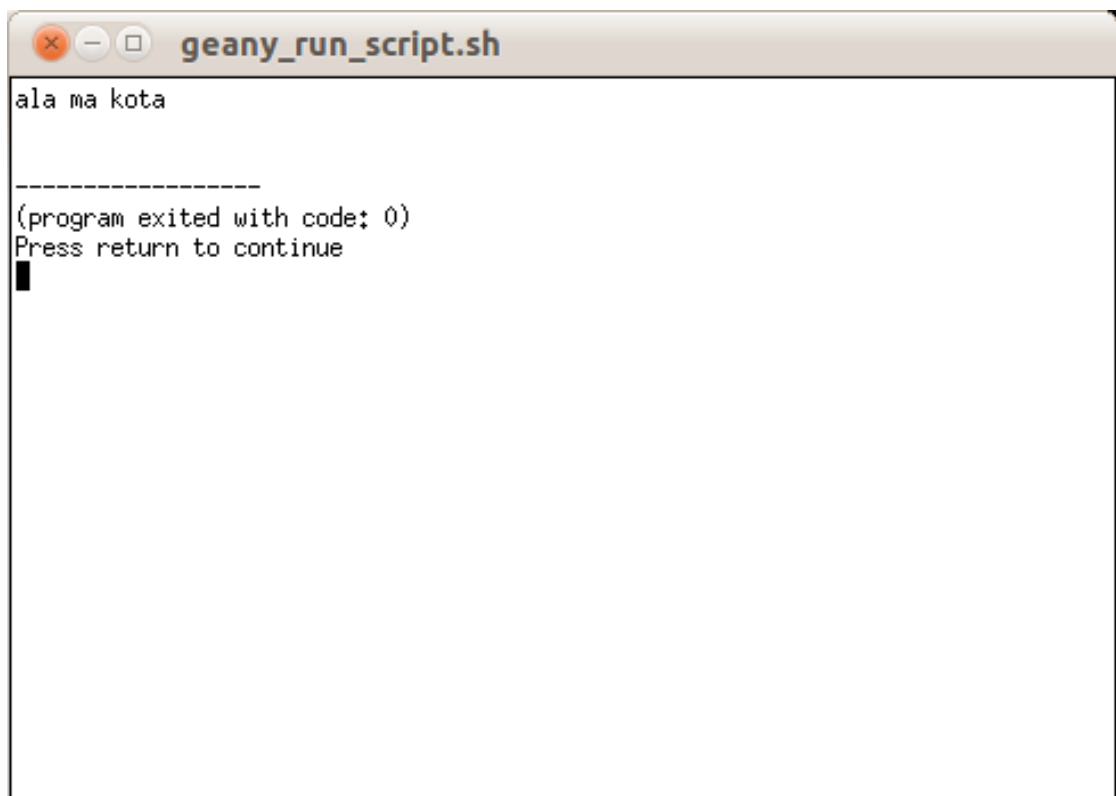
i możemy rozpocząć pisanie programu, czyli wstawiać zawartość przed poleceniem return. Dopiszmy tam

```
printf("ala ma kota\n");
```

Po dokonaniu zmian powinniśmy utworzony program zapisać Plik → Zapisz jako i podajemy jakąś nazwę. Jeżeli mogę coś doradzić, to najlepiej będzie jeżeli nazwa nie będzie zawierała żadnych odstępów (zamiast nich można stosować znak podkreślenia) i polskich liter. Powinna też mieć rozszerzenie c (czyli po kropce powinna występować literka c). Na przykład ala.c

Teraz możemy program skompilować i uruchomić. Z menu Zbuduj wybieramy Zbuduj (albo naciskamy klawisz F9). Aby uruchomić z menu Zbuduj wybieramy Execute (albo naciskamy klawisz F5).

Na potrzeby wykonania otwiera się osobne okienko terminala:



```
ala ma kota

-----
(program exited with code: 0)
Press return to continue
█
```

Naciśnięcie klawisza Enter powoduje zakończenie wykonania i zamknięcie okienka.

## 2. Funkcje matematyczne

Aby w programie móc skorzystać z funkcji matematycznych należy skonfigurować środowisko pracy. Polega ono na wykonaniu dwu dodatkowych czynności:

1. Po pierwsze zadeklarować w nagłówku, że z funkcji matematycznych będziemy korzystali, umieszczając linię

```
#include <math.h>
```

2. Po drugie skonfigurować środowisko IDE. W przypadku programu **geany** polega to na otwarciu okienka Zbuduj → Zdefiniuj polecenie budowania i w polu „C Poleceń” dla Build dodać na końcu wiersza, (po odstępach) „-lm”

```
gcc -Wall -o "%e" "%f"
```

żeby uzyskać

```
gcc -Wall -o "%e" "%f" -lm
```

Oraz kliknąć OK

Item	Etykieta	Polecenie	Working directory	Clear
<b>C source file Commands</b>				
1:	Skompiluj	gcc -Wall -c "%F"		<input type="button" value="X"/>
2:	Zbuduj	c -Wall -o "%e" "%f" -lm		<input type="button" value="X"/>
3:				<input type="button" value="X"/>
Error Regular Expression:				<input type="button" value="X"/>
<b>Non-Filetype Commands</b>				
1:	Make	make		<input type="button" value="X"/>
2:	Zbuduj przy użyciu własnych parametrów	make		<input type="button" value="X"/>
3:	Zbuduj obiekt	make %e.o		<input type="button" value="X"/>
4:				<input type="button" value="X"/>
Error Regular Expression:				<input type="button" value="X"/>
Note: Item 2 opens a dialog and appends the response to the command.				
<b>Execute Commands</b>				
1:	Execute	./%e"		<input type="button" value="X"/>
2:				<input type="button" value="X"/>
%d, %e, %f, %p are substituted in command and directory fields, see manual for details.				
				<input type="button" value="Anuluj"/> <input type="button" value="OK"/>

### 3. Geany pod Windows

Można próbować zainstalować Geany i kompilator C w środowisku Windows. Opisane to zostało w osobnym dokumencie.

### 4. Projekt

Geany jest jednym z prostszych IDE, więc nie daje zbyt wielu możliwości. Wydaje się jednak, że w każdym przypadku należy korzystać z projektów. Aby utworzyć projekt postępujemy w następujący sposób:

1. Z menu **Projekt** wybieramy **Nowy**. Wpisujemy nazwę projektu. Najlepiej aby dla każdego zajęcia był osobny projekt.

The dialog box titled "Nowy projekt" contains the following fields and controls:

- Nazwa:** An empty text input field.
- Nazwa pliku:** A text input field containing the path `/home/myszka/projekty/`.
- Katalog podstawowy:** A text input field containing the path `/home/myszka/projekty/`.
- Buttons for **Anuluj** and **Utwórz** are located at the bottom right.

2. Z menu **Plik** wybieramy **Nowy według szablonu: main.c**. (Uwaga, wybranie `main.cxx` pozwoli bez problemu korzystać z różnych konstrukcji dostępnych w języku C++<sup>1</sup>).
3. Plik zapisujemy (menu **Plik** → **Zapisz** i wybieramy jakąś nazwę. Może to być `main.c`, albo cokolwiek innego).

## 5. Wersja PDF tego dokumentu. . .

. . . pod adresem.

Wersja: 42 z **drobnymi modyfikacjami!** data ostatniej modyfikacji 2016-04-24 14:32:08 +0200

---

<sup>1</sup> Ale i tak zajęcia poświęcone są językowi C!