

Wojciech Myszka

CodeLite raz jeszcze

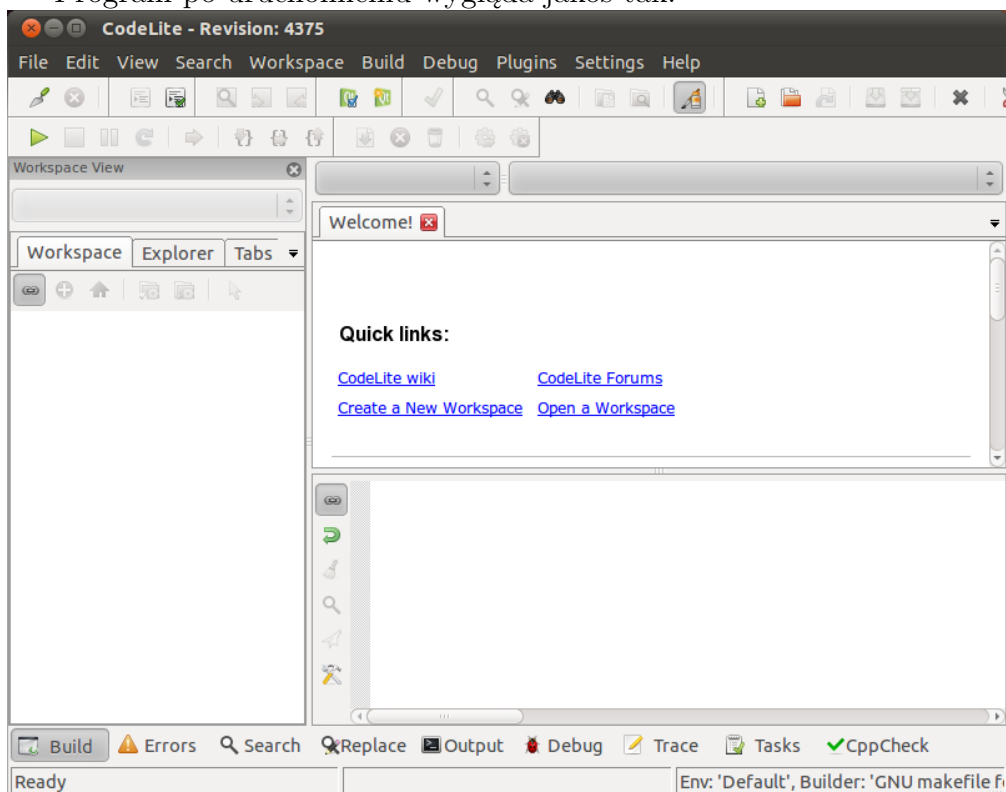
2016-05-07 08:55:02 +0200

1. Uruchomienie

Super-A i piszemy CodeLite.

CodeLite to nieco bardziej zaawansowane IDE. Program jest większy i w efekcie nieco wolniejszy. Aby z niego korzystać trzeba zapoznać się z ideą **projektu**. A nawet z ideą nieco bardziej ogólną: „przestrzeni roboczej” czyli Workspace.

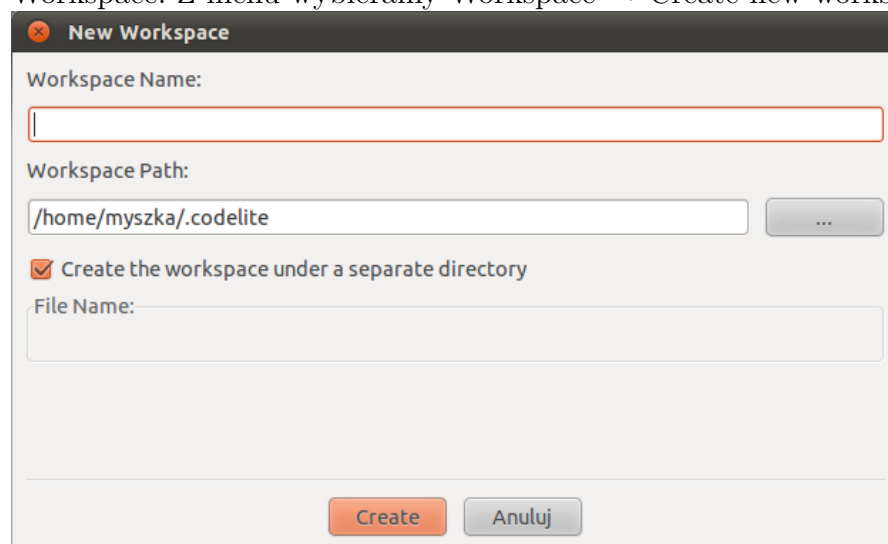
Program po uruchomieniu wygląda jakoś tak:



Nie jest niestety do końca spolszczony, ale nie powinno to być żadnym problemem.

2. Przestrzeń robocza

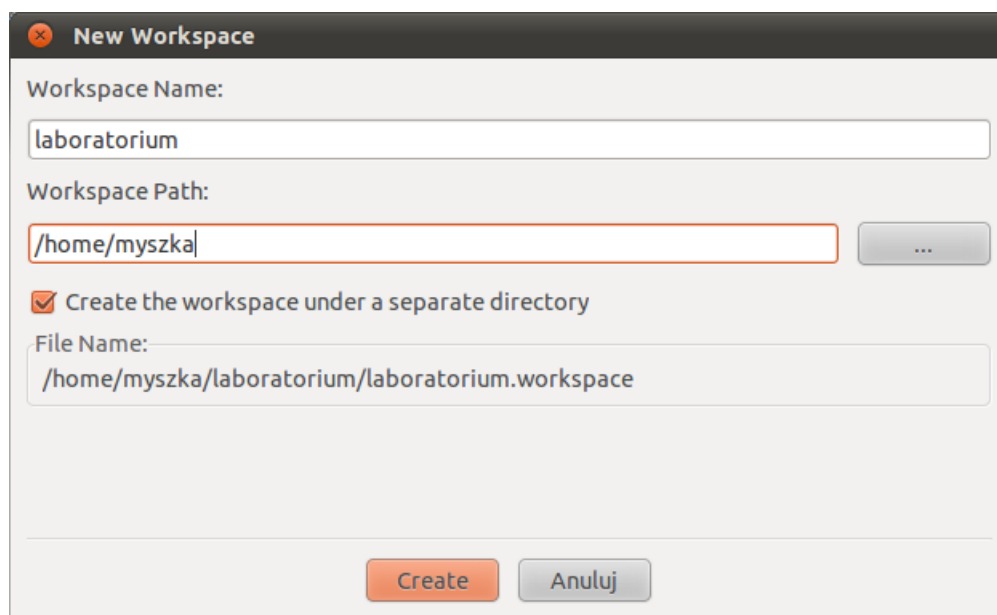
Pracę rozpoczynamy od przygotowania sobie przestrzeni roboczej: tworzymy Workspace. Z menu wybieramy Workspace → Create new workspace:



The screenshot shows a dialog box titled "New Workspace". It has a close button (red 'x') in the top-left corner. The dialog contains the following elements:

- Workspace Name:** A text input field that is currently empty.
- Workspace Path:** A text input field containing the path `/home/myszka/.codelite`, followed by a button with three dots (a file browser icon).
- Create the workspace under a separate directory**
- File Name:** A text input field that is currently empty.
- At the bottom, there are two buttons: **Create** (orange) and **Anuluj** (grey).

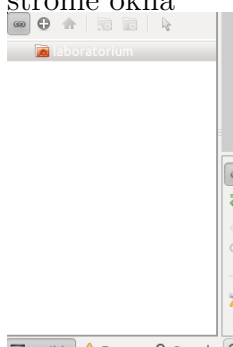
Musimy podać nazwę (na przykład Laboratorium) i zdecydować gdzie kartotekę z naszymi projektami umieścić. Proponuję nie umieszczać tego w kartotece `.codelite`, lepiej umieścić to w kartotece głównej (kasujemy napis `.codelite`)



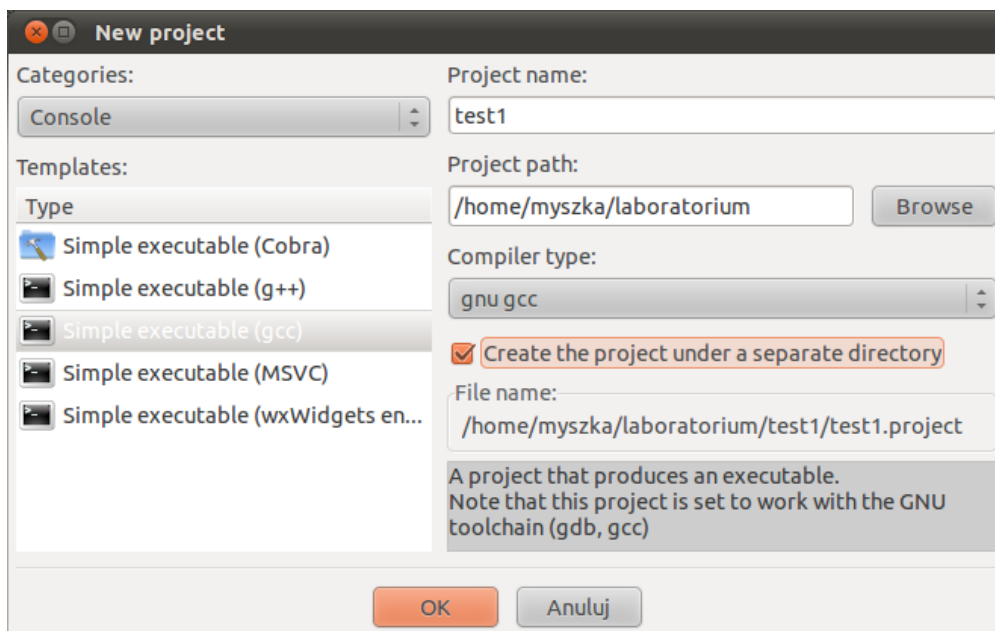
I naciskamy klawisz Create.

3. Projekt

Każde zajęcia laboratoryjne proponuję traktować jako jeden (lub więcej projektów). Aby utworzyć projekt wybieramy Workspace → Create New Project albo klikamy prawym klawiszem myszy na ikonie projektu w przeglądarce po lewej stronie okna

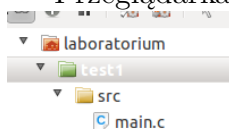


i wybieramy z menu Create New Project



Nie zmieniamy kategorii (Console), typ zmieniamy na Simple executable (gcc). Decydujemy się na nazwę projektu (test1) i wstawiamy ptaszka koło Create the project in separate directory. Na koniec klikamy w klawisz OK.

Przeglądarka po lewej stronie wygląda jakoś tak



Zielony kolor folderka przy nazwie naszego projektu (test1) oznacza projekt aktywny. CodeLite automatycznie utworzyło program źródłowy widoczny po podwójnym kliknięciu na nazwie pliku (main.c) w oknie edytora



Program modyfikujemy tak aby wyprowadzał jakiś napis, który lubimy:

```
#include <stdio.h>
```

```

int main(int argc, char **argv)
{
printf("Ala ma kota\n");
return 0;
}

```

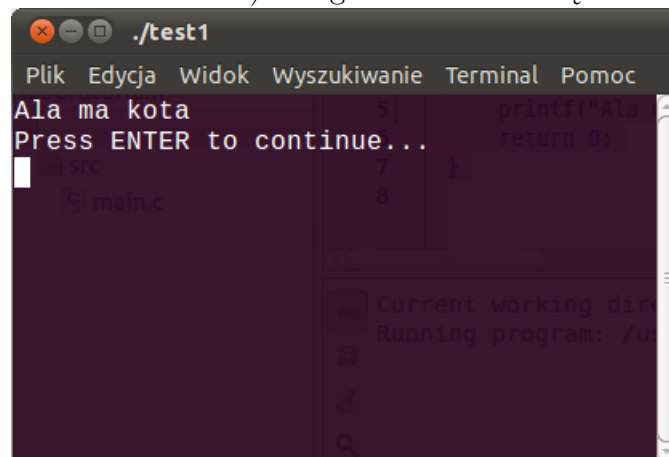
zapisujemy (File → Save albo Ctrl-S), następnie z menu Build wybieramy Build Project (albo naciskamy F7) (pojawienie się w dolnym okienku napisu:

```

/bin/sh -c "'make" -j 2 -f "laboratorium_wsp.mk"'
-----Building project:[ test1 - Debug ]-----
make[1]: Wejście do katalogu `/home/myszka/laboratorium/
gcc -c "/home/myszka/laboratorium/test1/main.c" -g -o
gcc -o ./Debug/test1 ./Debug/main.o "-L."
make[1]: Opuszczenie katalogu `/home/myszka/laboratorium,
-----Build Ended-----
0 errors, 0 warnings

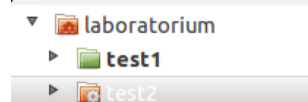
```

świadczy, że wszystko poszło dobrze. Uruchamiamy nasz program (Build → Run albo Ctrl-F5). Program uruchomi się w osobnym terminalu:



Naciskamy Enter aby terminal zlikwidować...

Kolejny projekt tworzymy w sposób podobny:



Ten „zielony jest aktywny” a pomarańczowy nieaktywny. Aby zacząć pracować z projektem test2 wskazujemy go kursorem myszy i prawym klawiszem wybieramy Set as Active

4. Wersja PDF tego dokumentu...

... pod adresem.

Wersja: 50 z **drobnymi modyfikacjami!** data ostatniej modyfikacji 2016-05-07 08:55:02 +0200