



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Techniki Informatyczne

Pierwszy program w ANSI C

Laboratorium - 03

Stanisław Flaga

Nazwa wydziału: WIMiR

Nazwa katedry: KAP

Kraków

http://home.agh.edu.pl/flaga_st -> „Dla studentów”

Na podstawie: materiały własne, Internet, literatura podawana na bieżąco

- 1. Określenie celu programu.**
- 2. Projektowanie programu.**
3. Pisanie kodu.
4. Kompilacja.
5. Uruchomienie programu.
6. Testowanie i usuwanie błędów.
- 7. „Pielęgnowanie” i modyfikacja programu**

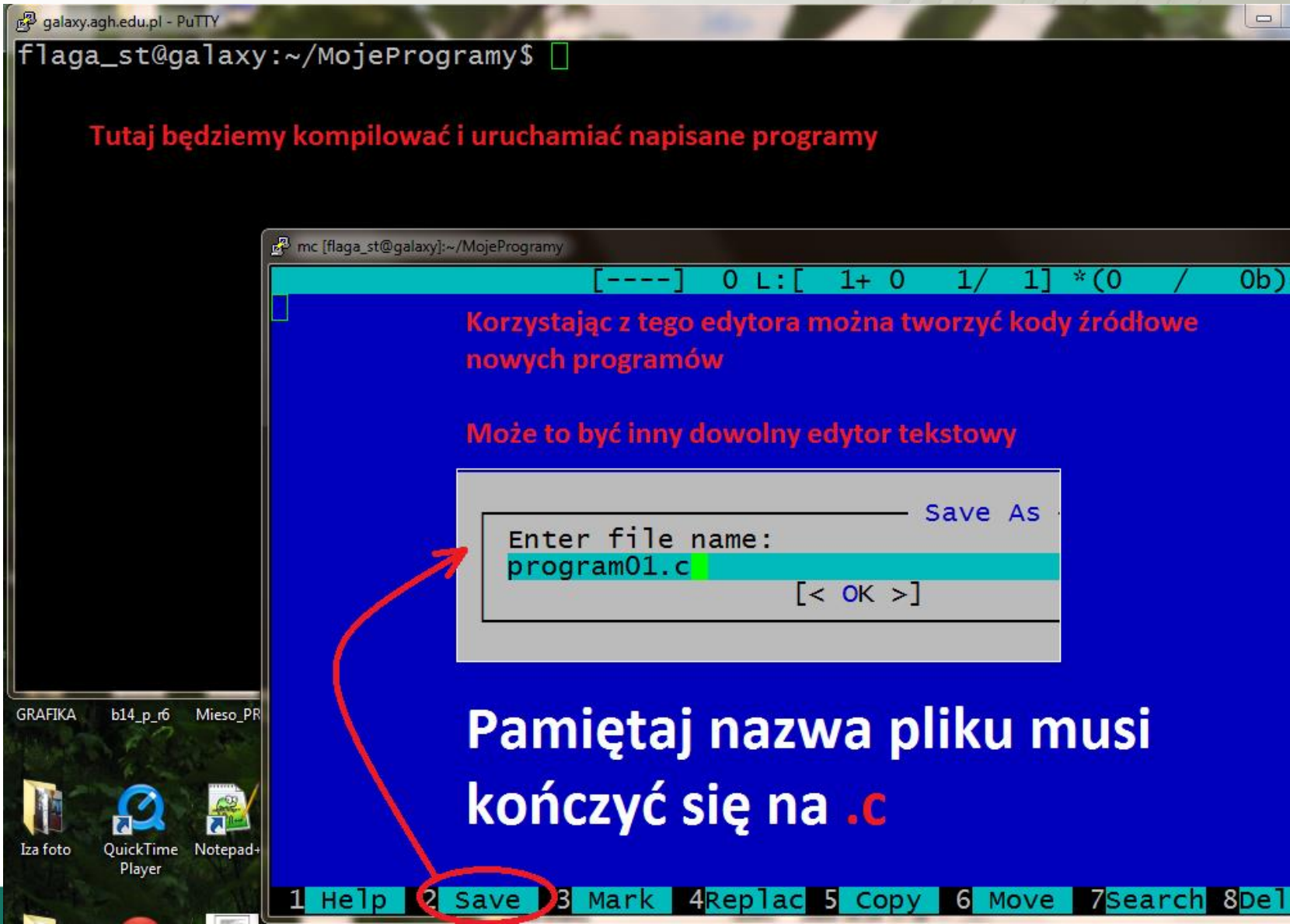
Akcje które należy wykonać aby stworzyć program w języku C

1. Korzystając z dowolnego edytora tekstu (np. **edytor programu mc** po stronie serwera lub **notepad++** po stronie komputera lokalnego) napisz kod źródłowy programu.
2. Skompiluj program przy pomocy kompilatora **gcc**
3. Jeżeli wystąpią błędy kompilacji popraw kod źródłowy i skompiluj go ponownie.
4. Uruchom program którego kod wynikowy powstał skutkiem działania kompilatora.

...i kolejny program, aż do osiągnięcia zamierzonych efektów...

1. Uruchom dwa razy sesję terminalową korzystając z programu *putty*.
2. Korzystając z odpowiedniej komendy systemu Linuks utwórz katalog **MojeProgramy**
3. W obydwu sesjach terminalowych uczynić katalog **MojeProgramy** katalogiem bieżącym.
4. W jednej z sesji uruchom program **mc**
5. W sesji z uruchomionym programem **mc** naciśnij kombinację klawiszy **SHIFT-F4** co uruchomi edytor tekstów.
6. Naciśnij klawisz **F2** wywołując akcję zapisu pliku tekstowego – nazwij plik **program01.c**
7. Wpisz kod źródłowy programu (*następna strona*).

Przykładowe środowisko pracy



galaxy.agh.edu.pl - PuTTY
flaga_st@galaxy:~/MojeProgramy\$

Tutaj będziemy kompilować i uruchamiać napisane programy

mc [flaga_st@galaxy]:~/MojeProgramy

[----] 0 L:[1+ 0 1/ 1] *(0 / 0b)

Korzystając z tego edytora można tworzyć kody źródłowe nowych programów

Może to być inny dowolny edytor tekstowy

Save As
Enter file name:
program01.c
[< OK >]

Pamiętaj nazwa pliku musi kończyć się na .c

1 Help 2 Save 3 Mark 4 Replac 5 Copy 6 Move 7 Search 8 Del

Struktura programu w języku C

```
program01.c [-M--] 0 L:[ 1+17 18/ 18] *(381 / 381b)= <EOF>
/*-----*/
/* To jest komentarz - tego nie "czyta" kompilator */
/* Warto tutaj wpisac dane autora */
/* oraz date ostatniej modyfikacji */
/*-----*/
#include<stdio.h>
int main(void)
{
    printf("\n\nTo moj pierwszy program \n\n");
    return 0;
}
```

Komentarz pomiędzy / i */*

Dyrektywa preprocesora dołączająca nagłówek biblioteki standardowej obsługi wejścia i wyjścia

Nagłówek funkcji głównej (początek programu)

Kod poprawnego zakończenia programu - koniec funkcji głównej

Ciało funkcji głównej

Pierwszy program – *program01.c*

```
program01.c      [-M--]  0 L:[ 1+17 18/ 18] *(381 / 3
/*-----*/
/* To jest komentarz - tego nie "czyta" kompilator */
/* Warto tutaj wpisac dane autora */
/* oraz date ostatniej modyfikacji */
/*-----*/

#include<stdio.h>

int main(void)
{
    printf("\n\nTo moj pierwszy program \n\n");

    return 0;
}
```

Pierwszy program - wyjaśnienia

/* - początek komentarza,

***/** - koniec komentarza,

- na prawo od znaku dyrektywa preprocesora,

"" - w cudzysłów wpisujemy łańcuch znaków (napis),

; - operator instrukcji (na lewo od niego instrukcja do wykonania)

\n - powoduje przejście do nowej linii,

printf() - funkcja wypisująca na konsoli,

return - słowo kluczowe języka ANSI C - zwraca wartość funkcji

Kompilacja programu

.agh.edu.pl - PuTTY

```
a_st@galaxy:~/MojeProgramy$ gcc -ansi program01.c -o program01
```

Nazwa kompilatora

Określenie standardu

Nazwa pliku z kodem źródłowym

Opcja -o powoduje, że plik wynikowy nie nazywa się a.out tylko tak jak chcemy

Poczytaj o innych opcjach kompilatora **gcc**

Jeżeli kod programu nie zawiera błędów to po uruchomieniu kompilacji otrzymamy

```
flaga_st@galaxy:~/MojeProgramy$ gcc -ansi program01.c -o program01  
flaga_st@galaxy:~/MojeProgramy$
```

Uruchomienie programu

W wyniku kompilacji powstał plik wykonywalny o atrybutach **755**

```
flaga_st@galaxy:~/MojeProgramy$ ls -l
total 16
-rwxr-xr-x 1 flaga_st imirgrp 9159 2011-11-02 21:16 program01*
-rw-r--r-- 1 flaga_st imirgrp  377 2011-11-02 19:52 program01.c
flaga_st@galaxy:~/MojeProgramy$
```

Uruchomić plik wykonywalny *program01* poprzez wywołanie jego nazwy

```
flaga_st@galaxy:~/MojeProgramy$ program01

To moj pierwszy program

flaga_st@galaxy:~/MojeProgramy$
```

Najczęstsze studenckie błędy kompilacji

Brak średnika

```
#include<stdio.h>

int main(void)
{
    printf("\n\nTo moj pierwszy program \n\n")
    return 0;
}
```

Numer linii

```
flaga_st@galaxy:~/MojeProgramy$ gcc -ansi program01.c -o program01
program01.c: In function 'main':
program01.c:13: error: expected ';' before 'return'
flaga_st@galaxy:~/MojeProgramy$
```

Najczęstsze studenckie błędy kompilacji

Brak zamykającego nawiasu falistego }

```
#include<stdio.h>

int main(void)
{
    printf("\n\nTo moj pierwszy program \n\n");

    return 0;
```

```
program01.c: In function 'main':
program01.c:13: error: expected declaration or statement at end of input
```

Zadania do wykonania

1. Uruchom program ***program01.c***
2. Wyszukaj informacje na temat funkcji **printf**
3. Napisz i uruchom nowy program ***program02.c*** wypisujący na konsoli różne zmienne stałoprzecinkowe z wykorzystaniem funkcji **printf**
4. Wyszukaj informacje na temat funkcji **scanf**