Warsztaty C cz. 1

Zuzanna Krawczyk
Bartek Chaber
Mariusz Zaborski

O czym będzie dzisiejsze spotkanie?

- len
- GIT
- vim
- clang
- zmienna długość list argumentów

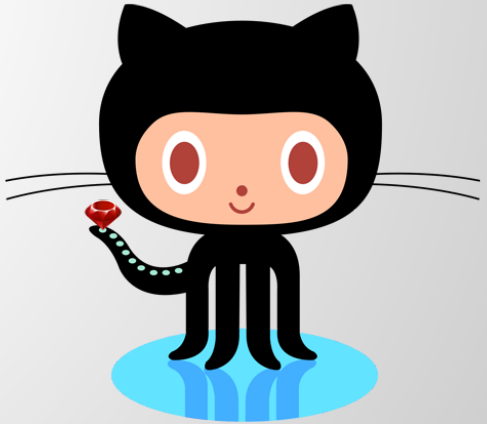
Zapraszamy po konto na lenie

ssh len

git na przykładzie github :)

Czym jest git?

- Rozproszony* system** kontroli*** wersji**** oprogramowania*****,
- Ułatwia pracę programisty.



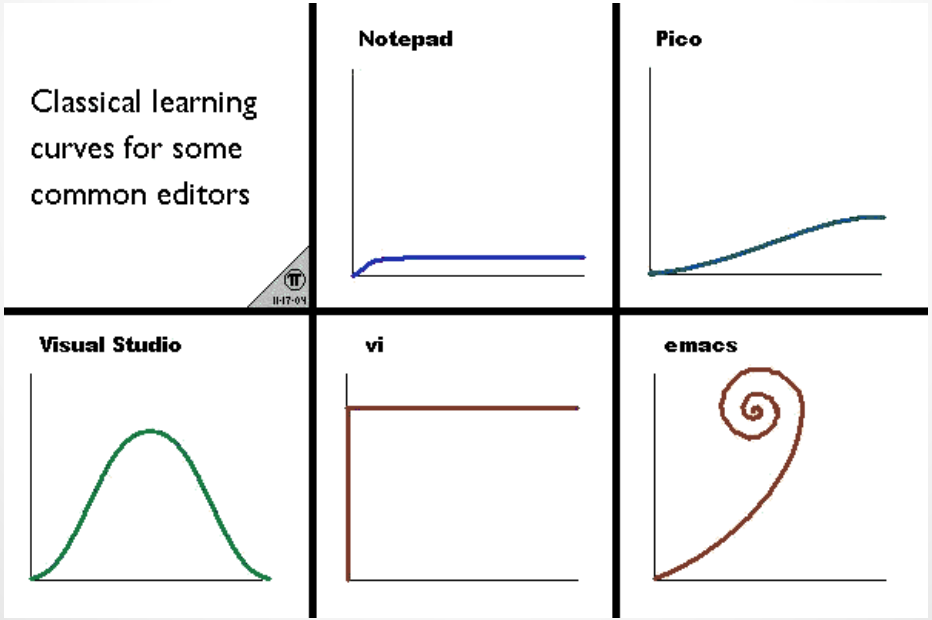
GitHub

- https://github.com/
- fork
- git clone https://github.com/NAME/C.git
- git add README
- git commit -m “Hello world”
- git push origin master
- git pull

vi na przykładzie vim :)

Czym jest vim?

- Przyspiesza pracę programisty.



Vim

/sth or **?sth** (wyszukuje w przód/w tył)

dd (usuwa bieżącą linię)

p (wkleja tekst ze schowka Vima)

:20 (przechodzi do linii 20)

:1,20y or **:1,20d** (kopiuje/usuwa linie od 1 do 20)

:2,4s/foo/bar/g (zastępuje wszystkie wystąpienia *foo* z *bar* w liniach od 2 do 4)

Vim

gg=G (automatyczne wcinanie całego pliku)
C-W C-V | **:vsplit** (dzieli ekran na pół w pionie)
:! pwd (wywołuje polecenie pwd)
!! pwd (wstawia wynik pwd do pliku)
% (przeskakuje pomiędzy pasującymi {}/()/[])
`` (przeskakuje do ostatnio edytowanej linii)
:make (wywołuje polecenie make)

Vim
vim -u vimrc integral.c

dw (usuwa najbliższe słowo)

r*znak* (zastępuje bieżący znak przez *znak*)

xp (zamienia kolejność dwóch następnych znaków)

u | C-R (undo/redo)

C-p | C-n (dopełnianie słów)

kompilator

twój przyjaciel clang

Czym jest clang?

- Alternatywa dla gcc
- kompatybilny z gcc
- bardziej czytelne komunikaty błędów

alias cc=’clang’

Zmienna liczba argumentów

Wreszcie C!

Przykłady funkcji:

- printf(“%s %s\n”, “Hello”, “world”)
- scanf(“%d”, &x)
- Py_BuildValue(“[i,i]”, 123, 456)

Jak to działa?

```
#include <stdio.h>

void fun();

int main(int argc, char *argv[])
{
    int a = 1, b = 2, c = 3;
    fun(a, b, c);
    return 0;
}
```

```
void fun(int args)
{
    int *pargs = &args;
    printf("%d\n", pargs[0]);
    printf("%d\n", pargs[1]);
    printf("%d\n", pargs[2]);
}
```

Zalecany sposób (ANSI C)

- # include <stdarg.h>
- va_start
- va_arg
- va_end

Przykład...

```
#include <stdarg.h>
double average( int count, ... )
{
    va_list ap;
    int i;
    double av = 0;
    va_start(ap, count);
    for(i = 0; i < count; i++)
        av += va_arg(ap, double);
    va_end(ap);
    return av / count;
}
```

Zadanie

Napisz funkcję, która zaalokuje pamięć dla struktury nieznanego typu wypełni ją wartościami podanymi jako argumenty opcjonalne, a następnie zwróci wskaźnik do zaalokowanej pamięci.

void *alloc(const char *fmt, ...);

Wywołanie:

```
struct test {
    int i1, i2;
    double d;
    int i3;
};

int main(){
    struct test *p = alloc("iidi", 33, 44, 3.14, 55);
    printf("%d %d %f %d\n", p->i1, p->i2, p->d, p->i3);
    return 0;
}
```

Dziękujemy za uwagę!