

Język C — przypomnienie

Systemy Operacyjne 2

Piotr Zierhoffer

8 listopada 2011

Nagłówki!!!:

- `man n NAZWA`, np. `man 2 read`, `man 3 printf`,
`man gcc`

Kompilacja”

- `gcc program.c`
- `gcc program.c -o program`
- **`gcc -Wall program.c -o program`**
- `gcc -g program.c -o program`

Domyślny plik wynikowy:

- `a.out`

Uruchamianie:

- `./a.out`
- `./program`

Debugowanie:

- flaga “-g” przy kompilacji
- `gdb program`
- komendy: `[h]elp`, `[b]reak`, `[p]rint`, `[r]un`, `[c]ontinue`, `[q]uit`

Szkielet programu

```
//nagłówki systemowe
#include <stdio.h>
//nagłówki lokalne
#include "myHeader.h"
//Dyrektywa kompilatora
#define BUFFERMAX 3*5*16
/*Deklaracje/definicje zmiennych globalnych*/
int buffer[BUFFERMAX]; int x=5;
//funkcje
void rockIt(void) {
    printf("Witaj świecie!");
}
int main(int argc, char **argv) {
    //zmienne lokalne
    int zxy = 0;
    while(zxy++ < 6)
        rockIt();
    //wartość zwracana
    return (a + b + c)/zxy;
}
```

```
#include <stdio.h>
int main(int argc, char **argv)
{
    int i;
    for(i = 0; i < argc; ++i)
    {
        printf("Argument[%d]: %s\n", i, argv[i]);
    }
    return 0;
}
```

```
struct my_struct{
    int a;
    char * b;
};
struct my_struct str1;

typedef struct my_struct my_str;
my_str str2;

my_str *p_str = &str2;

str2.b = "asdas";
p_str->a = 4;

typedef struct my_struct{
    int a, b;
} my_struct;
```

```
int a, b;

int *p_a = &a;

a = 3; // *p_a = 3

b = *p_a; //b = 3

*p_a = 6; //a = 6, *p_a = 6, b = 3

char tab[6];

char *same_tab = tab;

char *second_elem = &tab[1];

read(fd, same_tab + b, 2);

read(fd, second_elem, 2);

write(fd, second_elem - 1, 60);
```

```
char buffer[SIZE], outbuffer[LARGERSIZE];
int i = 0;
while(( n = read(fd, buffer + i, 1)) > 0)
{
    write(fd2, buffer + i, 1);
    i++;
}
memcpy( outbuffer + 5, buffer, i); //strncpy?
```



```
#include <stdio.h>
#include <unistd.h>
int main (int argc, char *argv[]){
    int opt;
    opterr = 0;        //disable error messages
    while ((opt = getopt(argc, argv, ":if:lr")) != -1){
        switch (opt){
            case 'i':
            case 'l':
            case 'r':

                printf("Option: %c\n", opt);
                break;
            case 'f':

                printf("Filename: %s\n", optarg);
                break;
            case ':':

                printf("Option %c needs a value\n", optopt);
                break;
            case '?':

                printf("Unknown option: %c\n", optopt);
                break;
        }
    }
    for (; optind < argc; optind++){
        printf("Argument: %s\n", argv[optind]);
    }
    exit(0);
}
```