



KATEDRA  
INFORMATIKY  
UNIVERZITA PALACKÉHO V OLMOUCI

# Jazyk C

## Ladění

Mgr. Markéta Trnečková, Ph.D.

# Jazyk C

*„If debugging is the process of removing bugs, then programming must be the process of putting them in.“*

(Sam Redwine)

# Chyby v programu

- **Syntaktické chyby**

- **Sémantické chyby**

- Nesprávné použití jazyka
- Chyba algoritmu

# Syntaktické chyby

The screenshot shows the Code::Blocks IDE with a C program in `main.c`. The code is as follows:

```
1 #include <stdio.h>
2
3 int main(){
4
5     int vysledek = 0;
6     for(int i=0; i<10; i++)
7     {
8         vysledek = vysledek + i;
9         printf("%i ",vysledek)
10    }
11
12    return 0;
13 }
14
```

The build log in the 'Logs & others' panel shows the following error:

File	Line	Message
C:\Skola\vyuk...		=== Build: Debug in ladeni (compiler: GNU GCC Compiler) ===
C:\Skola\vyuk...		In function 'main':
C:\Skola\vyuk...	10	error: expected ';' before ')' token
		=== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===

The error message indicates a missing semicolon at the end of the `printf` statement on line 9.

# Sémantické chyby

## Příklad

```
#include <stdio.h>

int main(){
    int a, b, r;
    printf("Zadejte cisla a a b: ");
    scanf("%i %i",&a, &b);
    printf("Vypocet gcd pro %i a %i\n", a, b);
    while(b != 0){
        r = a % b;
        a = b;
        b = r;
        printf("a: %i, b: %i\n", a, b);
    }
    printf("Nejvetsi spolecny delitel je %i", a);
    return 0;
}
```

# Ladění

- Analýza kódu
- Ladění pomocí debuggeru
- Ladící výpisy na `stderr`
- Využití podmíněného překladu
- Self-testy modulů nebo funkcí
- Využití knihovny `assert.h`



# Ladění

Ladící výpisy na stderr

```
fprintf(stderr, text, ...)
```

**Zavření proudu:**

```
fclose(stderr)
```

**Přesměrování proudu:**

```
fw = freopen("ladeni.log", "w", stderr);
```

# Ladění

## Využití podmíněného překladu

### Příklad

```
#ifdef DEBUG /* začátek podmíněného překladu */  
  
/* ladění část */  
  
#endif /* konec podmíněného překladu */
```



# Ladění

## Self-testy modulů nebo funkcí

### Příklad

```
#ifdef SELF
int funkceA(int a);

int main(int argc, char *argv[]){
    if(argc == 1){
        fprintf(stderr, "Nepredany zadny argument.");
        return 1;
    }
    printf("Vysledek = %d", funkceA(atoi(argv[1])));
    return 0;
}
#endif /* konec SELF */

int funkceA(int a){
    /* Definice funkce*/
    return vysledek;
}
```

# Ladění

## Využití knihovny `assert.h`

### Příklad

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

#define VELIKOST 10

int main(){
    int pole[VELIKOST];
    int cislo;
    int index;
    /* Nejaky kod */
    assert(index >=0 && index <= VELIKOST);
    pole[index] = cislo;
    return 0;
}
```

**Odstranění `assert`:** definice symbolické konstanty `NDEBUG`

# Cvičení

- 1 Vyzkoušejte si práci s `assert()`.
- 2 Určete, zda se v následujících kódech vyskytuje chyba a pokud ano, jakého je typu, případně, jak bychom jí opravili.

## Příklad (chyby1.c)

```
#include <stdio.h>

int main(){
    int x = 10;
    int y = 20;

    printf("(%i, %i)", x, y)
    return 0;
}
```

# Cvičení

3 Následující kód vypíše 10. Proč?

Příklad (chyby2.c)

```
#include <stdio.h>

int main(){
    int i;

    for(i = 0; i < 10; i++){
        printf("%i ", i);
    }
    return 0;
}
```

# Cvičení

4

Příklad (chyby3.c)

```
#include <stdio.h>

int main(){
    int i;

    for(i=0, i<10, i++){
        printf("%i ", i);
    }
    return 0;
}
```

# Cvičení

5

Příklad (chyby4.c)

```
#include <stdio.h>

int main()
{
    int pole[3]={1, 2, 3};

    printf("%i ", pole[3]);
    return 0;
}
```

# Cvičení

6

## Příklad (chyby5.c)

```
#include <stdio.h>

int main(){
    int b;

    while(b > 0){
        printf("*");
        b--;
    }
    return 0;
}
```

# Cvičení

7

## Příklad (chyby6.c)

```
#include <stdio.h>

int main(){
    int n = 10;
    int m = 0;

    m = n / 0;

    printf("%i ", m);
    return 0;
}
```



# Cvičení

8

Příklad (chyby7.c)

```
#include <stdio.h>

int main()
{
    int a=0;

    if(a=0){
        printf("a je 0");
    }
    else{
        printf("a není 0");
    }
    return 0;
}
```

# Cvičení

9

Příklad (chyby8.c)

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

```
int main(){  
    int a = 10, b = 20, c;  
    a + b = c;  
    return 0;  
}
```

# Cvičení

10

## Příklad (chyby9.c)

```
#include <stdio.h>
int i;
void vykresli_trouhelnik(int n){
    for(i = 1; i <= n; i++){
        for(int j = 0; j < i; j++){
            printf("*");
        }
        printf("\n");
    }
}
int main(){
    int n = 5;
    for(i = 1; i <= n; i++){
        vykresli_trouhelnik(i);
    }
    return 0;
}
```

# Cvičení

11

Příklad (chyby10.c)

```
#include <stdio.h>

typedef struct{
    int x;
    int y;
    int pole[2];
} struktura;

int main(){
    struktura m;
    m = {10, 2, {10, 2}};
    return 0;
}
```

## Cvičení

12 Následující kód skončí chybou při překladu. Proč tomu tak je?

Příklad (chyby11.c)

```
#include <stdio.h>

int main(){
    char *karty = "JQK";
    char karta = karty[2];

    karty[2] = karty[1];
    karty[1] = karty[0];
    karty[0] = karty[2];
    karty[2] = karty[1];
    karty[1] = karta;

    printf("%s", karty);
    return 0;
}
```

## Cvičení

- 13 Určete, které z následujících kódů se podaří zkompileovat. V případě neúspěchu jaké se v kódu objevují chyby? V případě úspěchu určete, jaký bude výstup a zkuste odhadnout, zda se program chová tak, jak by měl.

### Příklad (chyby12a.c)

```
#include <stdio.h>

int main(){
    int karta = 1;

    if(karta > 1)
        karta = karta - 1;
        if(karta < 7)
            printf("Mala karta");
    else
        printf("Eso");

    return 0;
}
```

# Cvičení

## Příklad (chyby12b.c)

```
#include <stdio.h>

int main(){
    int karta = 1;

    if(karta > 1){
        karta = karta - 1;
        if(karta < 7)
            printf("Mala karta");
    }
    else
        printf("Eso");
}

return 0;
}
```

# Cvičení

## Příklad (chyby12c.c)

```
#include <stdio.h>

int main(){
    int karta = 1;

    if(karta > 1){
        karta = karta - 1;
        if(karta < 7)
            printf("Mala karta");
    }else
        printf("Eso");

    return 0;
}
```



# Cvičení

## Příklad (chyby12d.c)

```
#include <stdio.h>

int main(){
    int karta = 1;

    if(karta > 1){
        karta = karta - 1;
        if(karta < 7)
            printf("Mala karta");
    }
    else
        printf("Eso");

    return 0;
}
```