

Životnost/viditelnost proměnných

- Z hlediska životnosti (kdy proměnná fyzicky existuje) a viditelnosti (lze ji použít v programu) rozlišujeme proměnné na globální a lokální.
- Globální proměnná je definována mimo funkce a leží v globální paměti. Existuje (žije) po celou dobu programu a je dosažitelná ze všech modulů. Je možné ji použít od deklarace dále.
- Použití globálních proměnných se doporučuje jen v nejnútnejších případech (kód s těmito proměnnými se nedá sdílet- pro více řešení, mezi vlákny ...)
- funkce je vždy globální
- Lokální proměnné jsou proměnné, které existují pouze v bloku, ve kterém byly definovány. Jejich viditelnost/možnost použití je stejná – od místa deklarace do konce bloku ve kterém byly deklarovány. Proměnná (v C) může být definována pouze na začátku bloku. Lokální proměnné se tvoří na zásobníku. Lokálními parametry jsou i parametry funkce.

Rozdělte lokální a globální proměnné. Kde začíná a končí jejich platnost

```
extern double bb ;  
int pocet ;  
extern int cc ;  
  
int main(int argc, char *argv[])  
{  
    int a ;  
    // příkazy  
    {  
        double b ;  
        // příkazy  
    }  
    // příkazy  
    return 0 ;  
}
```

```
extern double bb ; // deklarace - oznamuji existenci  
                // proměnné, fyzicky je v globální paměti  
int pocet ;      // definice - oznamuji jméno, podle typu  
                // se zabere paměť v globální paměti  
extern int cc ;  // oznámení jména cc a jeho typu int
```