

premennej je uvedená rovnakým spôsobom ako jej definícia, len je pred ňou uvedené kľúčové slovo `extern` (a v prípade inicializovanej premennej sa v deklarácii inicializátor neuvádza).

Bolo by možné uviesť deklarácie všetkých globálnych premenných, ktoré sú v nejakom module použité, „manuálne“ na začiatku tohto modulu. Toto by však bolo prácne a ťažko udržiavateľné, s rizikom, že pri zmene skutočnej definície premennej sa táto zmena nepremietne do všetkých zdrojových súborov, kde je táto premenná deklarovaná. Preto je dobrým zvykom deklarácie globálnych premenných z určitého modulu (zdrojového súboru `.c`), ktoré majú byť navonok viditeľné, zhrnúť do hlavičkového súboru prislúchajúceho k danému zdrojovému súboru (pozri kapitolu 12). Tento hlavičkový súbor je potom vložený pomocou `#include` do ostatných zdrojových súborov projektu. Hlavičkové súbory teda slúžia na „zviditeľnenie“ či akýsi „export“ globálnych premenných a funkcií modulu navonok, aj keď tento postup je vlastne daný zvyklosťami a nie je nevyhnutný pre preklad, ani vyžadovaný normou.

### 33. Programátorské paradigmy VI.

*Makrá*

```
FILE uart_str = FDEV_SETUP_STREAM(uart_putchar, uart_getchar,  
_FDEV_SETUP_RW);  
FILE lcd_str = FDEV_SETUP_STREAM(lcd_putchar, NULL,  
_FDEV_SETUP_WRITE);
```

`static void`

Makro je prostriedok preprocesora C, ktoré umožňuje **doslovnú náhradu** určitého reťazca (mena makra) v zdrojovom texte iným reťazcom. Táto náhrada je definovaná pomocou kľúčového slova `#define`, nasleduje meno makra, oddeľovač (*whitespace*), a za ním nasleduje text, použitý ako *náhrada*. Text môže obsahovať už aj *whitespace* znaky a použije sa pri nahradzovaní makra celý až do konca riadka (resp. po kombináciu znakov označujúcu začiatok komentára, ak sa také na tomto riadku nachádzajú). Text môže obsahovať iné, predtým definované makrá, ktoré budú tiež nahradené, t.j. makrá je možné navzájom **vnárať**.

Makrá sa teda dajú použiť nielen na definíciu resp. „pomenovanie“ konštant, ale aj na definovanie akéhokoľvek textu, napríklad

na zjednodušenie či skrátenie zápisu „hlavnej časti“ programu, alebo na zjednodušený zápis opakujúceho sa textu.

Keďže pri definícii makra je koniec riadku ukončovacím znakom textu predstavujúceho náhradu, makrom sa nedajú nahrádzať viacriadkové texty; to však nevadí, keďže zariadkovanie v samotnom jazyku C nemá väčší význam ako akýkoľvek iný *whitespace* znak. Makrá sa však dajú zapísať spôsobom, ktorý je z „estetického“ hľadiska podobný viacriadkovému textu, pomocou znaku \ (spätného lomítka) na konci riadku, vo funkcii znaku pre „pokračovanie riadku“:

```
#define set_sleep_mode(mode) \
do { \
    _SLEEP_CONTROL_REG = \
        ((_SLEEP_CONTROL_REG & ~(_BV(SM0) | _BV(SM1))) | (mode)); \
} while(0)
```

Preprocesor znak \ na konci riadku odstráni spolu so zariadkovaním, t.j. ku riadku, ktorého posledný znak pred zariadkovaním je \, pripojí nasledujúci riadok. Tento proces sa udeje ešte pred vyhodnotením direktív preprocesora<sup>34</sup>, vrátane direktívy #define, takže *náhrada* v takto definovanom makre už zariadkovanie neobsahuje.

Uvedená ukážka ilustruje aj rozšírenú prax, keď sa makrá, ktoré predstavujú ucelené výrazy, uzatvárajú do cyklu do{} while(0). Tento cyklus sa vykoná len raz, a teda je funkčným ekvivalentom samotného bloku {}. Dôvod na tento postup je ten, že takéto makrá sa používajú výzorovo zhodne s volaniami funkcií, t.j. ako výraz, za ktorým sa píše bodkočiarka; to však v kladnej vetve príkazu if pred else spôsobí chybu, pretože tam môže byť prítomný len jeden výraz (jednoduchý, ukončený bodkočiarkou, alebo zložený, t.j. samotný blok, za ktorým už nesmie nasledovať bodkočiarka, lebo taká kombinácia je pokladaná už za dva výrazy). Toto konkrétne makro obsahuje len jeden výraz a teda by bolo možné ho zapísať bez uzavretia do bloku a bez bodkočiarky, ale bol by to neobvyklý zápis, a u makier obsahujúcich viacero výrazov takéto riešenie aj tak nie je možné.

Keďže správanie makier je v podrobnostiach odlišné od správania premenných, funkcií a iných „natívnych“ prvkov jazyka, aby nedošlo pri použití makier k omylu, niektoré štýlové predpisy vyžadujú, aby **mená makier** neobsahovali malé písmená, t.j. aby boli tvorené len **veľkými písmenami**, číslicami a špeciálnymi znakmi.

---

34 C99, §5.1.1.2 - táto kapitola predpisuje jednotlivé fázy prekladu