

STIL DE PROGRAMARE

Stilul de programare se definește ca un set de reguli utilizat în scrierea codului sursă pentru programe de calculator, având drept scop facilitarea citirii și înțelegerii rapide a acestuia de către toți dezvoltatorii. Acest articol se adresează în principal celor care nu și-au definit până în acest moment un stil bun de programare, dar totodată și cei experimentați ar putea găsi câteva lucruri interesante și folositoare.

În următoarele câteva rânduri vor fi prezentate principalele elemente de stil propuse de Linus Torvalds, creatorul kernel-ului linux, și totodată cel care conduce în prezent echipa de dezvoltare a acestuia.

1. INDENTARE ȘI SPATIERE

- indentarea se utilizează pentru a defini în mod clar unde începe un bloc de control și unde se încheie
- pentru a realiza indentarea codului se recomandă folosirea tab-urilor și nu a spațiilor (mărimea recomandată pentru tab este 4)
- puneți câte un spațiu după fiecare variabilă, operator, paranteză. Codul trebuie să fie cât se poate de **AERISIT**.

Urmăriți cu atenție exemplul :

```
int power( int a, int b )
{
    int c = 1;

    while ( b > 0 ) {
        c = c * a;
        b = b - 1;
    }

    return c;
}
```

2. AMPLASAREA ACOLADELOR

- stilul recomandat în această privință este amplasarea acoladei deschise ultima, pe linie cu instrucțiunea către care face referință, iar acolada închisă va fi plasată pe linie separată după cum se observă în exemplul de mai jos.

```
if ( x is true ) {
    we do y
}
```

- Exista si o exceptie de la regula de mai sus in cazul definirii functiilor. Acolada deschisa va fi plasata la inceputul liniei urmatoare conform exemplului de mai jos.

```
int count( int x )
{
    body of function
}
```

- Atentie ! Acolada inchisa va fi singura pe linie cu exceptia cazurilor cand este urmata de continuarea aceleiasi declaratii. Urmariti exemplul de mai jos !

```
do {
    body of do-loop
} while (condition);

SAU

if (x == y) {
    ..
} else if (x > y) {
    ...
} else {
    ....
}
```

3. STABILIREA NUMELOR

- Un programator C nu va da nume frumoase variabilelor cum ar fi *ThisVariableIsATemporaryCounter*, o va denumi pur si simplu *tmp*.
- Variabilele globale(vor fi folosite doar daca sunt absolut necesare) trebuie sa aiba un nume descriptiv, precum si functiile globale. De exemplu, daca o functie numara utilizatorii activi ea va trebui denumita *count_active_users()* si NU *cntusr()*.
- Numele variabilelor locale trebuie sa fie scurt si la obiect. De exemplu un contor de ciclu se va denumi simplu *i* si nu *loop_counter* pentru ca este foarte neproductiv.

4. FUNCTII

- functiile trebuie sa fie scurte si sa faca un singur lucru. Marimea unei functii nu trebuie sa depaseasca unul sau doua ecrane (ISO/ANSI marime ecran 80x24).
- lungimea unei functii trebuie sa fie invers proportionala cu nivelul de indentare si complexitatea acesteia. Astfel daca avem o functie alcatuita dintr-un CASE foarte mare in interiorul caruia se fac operatii simple atunci este in regula sa avem o functie mai lunga.
- Un alt aspect care trebuie avut in vedere in cazul functiilor este numarul de variabile locale. Daca acesta este mai mare de 5-10 atunci functia in discutie trebuie reorganizata si impartita in bucati mai mici.

5. COMENTARIII

- Comentariile sunt un lucru bun, dar exista riscul unei supra-comentari. Niciodata nu incercati sa explicati cum functioneaza codul; daca codul este scris clar atunci acest lucru va fi evident.
- In general comentariile descriu ce face un cod, si nu cum face acel lucru. Incercati de asemenea sa evitati amplasarea comentariilor in interiorul functiilor. Daca functia este atat de complexa incat trebuie sa comentati parti ale ei, atunci mai cititi de cateva ori punctul 4.
- In interiorul functiilor se pot face mici comentarii pentru a clarifica anumite lucruri speciale, dar totusi nu trebuie exagerat in aceasta privinta. Puneti comentarii la inceputul functiilor oferind detalii cu privire la ce face si daca este cazul de ce face acele lucruri.

Documentatie completa :

- <http://lxr.linux.no/linux/Documentation/CodingStyle>