

Guide de style

par [Aurélien Lucchi](#)

Date de publication : 22/08/2005

Dernière mise à jour :

Ce guide de style est destiné aux programmeurs du projet SKAN. Il expose quelques règles élémentaires à suivre lors de l'écriture des fichiers sources.

Vous pouvez aussi consulter cet article [au format PDF](#).

Si ce lien ne fonctionne pas chez vous, utilisez [celui-ci](#).

- I - Commentaires
- II - En-tête et squelette
- III - Indentation
- IV - Notation des labels et variables
- V - Typographie
- VI - Variables
- VII - Macros
- VIII - Versions

I - Commentaires

P-1 Rappeler les principales étapes de l'algorithme dans le code.

P-2 Les principales parties du code doivent être séparées par une ligne du type suivant :

```
; ***** TITRE EN MAJUSCULE ENTOURE PAR 5 ETOILES DE CHAQUE COTE *****
```

P-3 Expliquer les passages de l'algorithme au code ne présentant pas un caractère évident. Un commentaire doit expliquer et non dupliquer. Exemple de commentaire sans intérêt :

Add ax,1 ; on incrémente ax de une unité !!!

P-4 La mise en page doit faciliter la lecture du code et des commentaires associés, par exemple en reportant les commentaires sur la droite.

Quelques exemples

```
; ***** TITRE EN MAJUSCULE ENTOURE PAR 5 ETOILES DE CHAQUE COTE *****

mov ah, al ; commenter ligne à ligne
mov bh, bl ; avec la plus grande précision
mov dh, dl ; en pensant à aligner les commentaire pour plus de clarté
or  ch, cl ; aligner aussi les instructions et les arguments
           ; (2 espaces après un "or" ou un "je")

; Rmq: Insérer des remarques pour expliquer certaines
;      partie 'obscurés' du code.

.boucle:
    mov ah, al ; indenter les parties de code qui forment des boucles
    mov bh, bl ; indenter avec des tabulations (caractère ASCII 9)
    mov ch, cl ; mais pas avec des espaces,
               ; pour éviter les indentations incohérente quand on change
               ; d'éditeur de texte !
loop .boucle ;

; penser aussi à ne pas dépasser 80 caractères de largeur,
; pour pouvoir éventuellement lire les source en mode texte
; (il m'arrive d'utiliser VI ;) )
```

P-5 Pour chaque procédure, spécifier les entrées et sorties en commentaires.

Exemple

```
; ***** nom_fonction *****
; - Synopsys :
; - Entrée : DS : segment de données
; - Sortie : Aucune modification.
; *****
```

P-6 Si des bouts de code ne sont pas définitifs dans votre code, que vous devez optimiser plus tard, spécifier y avec un "TODO" en commentaire suivi d'une remarque appropriée.

II - En-tête et squelette

Exemple

```

;=====
; Projet: SKAN   Version:0.00.0000rev0
;-----
; Synopsys: Mettre ici un résumé du contenu du fichier
;
; Syntaxe : FASM
;
; Auteurs:
;   auteur1 auteur1@stuff.fr
;   auteur2 auteur2@stuff2.xx
;   auteur3 ...
;   ...
;
; Historique :
;   -xx/xx/xx: aaaa a programmé le support des zzz
;   -24/03/05: Création du fichier
;   -xx/xx/xx: stuffs
;   -....
;=====
; This program is free software; you can redistribute it and/or modify it
; under the terms of the GNU General Public License as published by the
; Free Software Foundation; either version 2 of the License,
; or (at your option) any later version.
;
; This program is distributed in the hope that it will be useful,
; but WITHOUT ANY WARRANTY; without even the implied warranty of
; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
; See the GNU General Public License for more details.
;
; You should have received a copy of the GNU General Public License
; along with this program; if not, write to the Free Software Foundation,
; Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
;=====

; ***** ZONE1 *****

                org 0                                ; Mettre le décalage à zéro

; ***** ZONE2 *****

; ***** INCLUSION DES FICHIERS *****

include "../XXX.asm"

; ***** DECLARATION DES DONNEES *****
.msgBoot      db " Lancement de SKAN. ", 0

```

[Télécharger le squelette](#)

Si ce lien ne fonctionne pas chez vous, utilisez [celui-ci](#).

III - Indentation

P-1 Les instructions (hormis les labels) devront être indentées d'une tabulation (caractère ASCII n°9).

```
label1:
    mov  ax, bx
    inc  cx

label2:
    shl  dx
    macro1
```

IV - Notation des labels et variables

P-1 Tous les variables doivent porter un nom rappelant leur signification. Il n'y a pas de variable dont le nom suggère une autre signification que son sens réel. Il n'est pas utile que le nom rappelle le type de la variable.

P-2 Les noms doivent être écrits en français (évidemment, évitez les fautes d'orthographe).

P-3 Les noms doivent commencer par un caractère alphabétique minuscule.

P-4 Un underscore doit séparer chaque mot composant le nom.

P-5 Les dots sont réservés aux structures, aux éléments naturellement hiérarchisés, et aux subfonctions.

P-6 Deux opérations de même nom doivent effectuer des traitements similaires.

V - Typographie

P-1 Pour des problèmes d'impression et de facilité de lecture dans une fenêtre, limiter la longueur des lignes à 80 caractères.

VI - Variables

P-1 Déclarer une seule variable par ligne et en profiter pour expliciter son rôle.

P-2 Une seule variable non-locale peut être déclarée dans un fichier (ce n'est pas obligatoire, par exemple, dans le cas d'un fichier à inclure dans un autre fichier).

VII - Macros

P-1 L'identifiant doit être court, précis, général, et en majuscules.

```
macro TEST_Exemple foo1, foo2
{
    ; ...
}
```

VIII - Versions

P-1 La numérotation des versions: Chaque version devra être notée de la façon suivante: 0.00.0000rev0 Le premier nombre indique sur quel liste de tâches on se base (par exemple actuellement, on en est au début, ce sera 0; quand on aura terminé les objectifs basiques que l'on s'est déjà fixé, on établira de nouveaux objectifs, et ce sera la version 1). Le deuxième nombre indique la progression dans l'établissement des objectifs de la liste de tâches actuelle. Le troisième nombre sert juste au développement; à chaque qu'une petite fonctionnalité est implémentée (pas forcément entièrement), on incrémente ce nombre. Le numéro de révision est incrémenté à chaque fois qu'un bug d'une version est corrigé.

P-2 Un nom d'astre viendra s'ajouter à chaque numéro de version facilitant ainsi l'appellation d'une version pour les utilisateurs.

