

# Projet EX4L - Conventions de codage C

Deviron Manoëlle, Petit Jonathan, Wéry Benoît

27 novembre 2017

Ce document reprend une liste des conventions de codage qui seront adoptées par le groupe dans le cadre du projet de *Systèmes d'exploitation*.

## 1 Nomination et variables

1. Les noms des fichiers et les variables sont en minuscules avec un '\_' comme séparateur.
2. Les noms des constantes sont en majuscules avec un '\_' comme séparateur.

```
1 mon_fichier.c
2 int une_variable
3 double UNE_CONSTANTE
```

3. La déclaration des variables se fait sur plusieurs lignes en rassemblant les types :

```
1 int a = 42;
2 int b = 24;
3 double c = 100.42;
```

## 2 Commentaires

1. Les commentaires se trouvent toujours au dessus du code concerné et suivent le format ci-dessous :

```
1 Commentaires une ligne:
2 /* ... */
3
4 Commentaires multilignes:
5 /* ...
6  * ...
7  */
```

## 3 Indentation, lignes blanches, longueurs,...

1. La longueur maximum d'une ligne est de 80 caractères
2. L'indentation se fait avec 4 espaces uniquement
3. Les interlignes suivent le format suivant :

- Deux lignes blanches entre la fin d'une fonction et la documentation de la suivante
  - Une ligne blanche avant et après pour isoler les blocs (if, while, for,...)
4. Les opérateurs (arithmétiques et logiques) sont toujours encadrés d'un espace blanc de chaque côté par rapport aux opérandes

```
1  int variable = 42;
2  if(variable >= 42)
```

## 4 Procédures et fonctions

1. Les noms des procédures suivent le modèle "lowerUpper case"
2. La parenthèse ouvrante est collée au nom et la déclaration tient sur une même ligne (sauf dans le cas où la longueur max ne peut être respectée, alors les paramètres seront mis en colonne).
3. Les accolades ouvrantes/fermantes se trouvent en colonne 1 (par rapport à la déclaration)
4. Les fonctions doivent être précédées d'une documentation contenant une description et des conditions pre/post qui suit le format de l'exemple ci-dessous

Exemple :

```
1  /* Description de la fonction
2   * @pre - conditions sur les parametres d'entree
3   * @post - conditions sur les types de retour
4   */
5  type_retour nomFonction(type param_1, type param_2)
6  {
7      [...]
8  }
```

5. Quelques nominations particulières :
  - vérification de valeur : ISBLABLA(...)
  - obtention de valeur : GETBLABLA()
  - changement de valeur : SETBLABLA(...)

## 5 Blocs d'instructions

1. Les instructions if/else suivent le format suivant :

```
1  if(....)
2  {
3      ...
4  } else {
5      ...
6  }
```