

# Langage C avancé

## Utilisation de assert()

**Samuel KOKH**

`samuel.kokh@cea.fr`

MACS 1 – Institut Galilée

## Fonction `assert()`

La fonction `assert()` permet au programmeur de tester des expressions booléennes et d'arrêter l'exécution du programme si l'expression à tester est fausse.

Dans le cas d'un arrêt, en plus d'arrêter le programme `assert()` va retourner sur la sortie standard des message qui indique à quel endroit l'exécution a été stoppée.

## Utilisation de `assert()`

La fonction `assert()` est essentiellement utilisée pour

- le débogage
- la mise de tests pour le code

Il est possible d'implémenter des tests unitaires avec `assert()`, néanmoins il existe des bibliothèques complètement dédiées à la gestion des tests unitaires comme par exemple `CUnit` propose des synthèses des résultats des tests.

( <http://cunit.sourceforge.net> )

# Exemple

---

```
#include <assert.h>

int main(void){
    int a=0;
    int b=0;
    ...

    assert(a<b);
    ...
    assert(a==2);

    return EXIT_SUCCESS;
}
```

---

## Compilation et désactivation de assert()

Pour utiliser `assert()` la phase de compilation ne nécessite pas de procédure spécifique. Ainsi l'option de débogage `-g` de `gcc` n'est pas nécessaire.

### Désactivation

Il est possible de désactiver tous les appels à `assert()` en définissant la macro `NDEBUG`. On peut le faire :

- dans le code avec un

```
#define NDEBUG
```
- au moment de l'appel à `gcc` via l'option `-D`

```
$ gcc -DNDEBUG ...
```