**TD de Programmation : Entrées/Sorties bit-à-bit**

**1 - Manipulation de bits** (f*ichier concernés : bitabit.c, bit\_operations.c et bit\_operations.h)*

Dans bit\_operations.c, nous utilisons 3 fonctions :

1) Pour afficher une variable en binaire :

int get\_bit(int value, int position)

{

return (value >> position) **& 1**;

}

Cette fonction réalise un décalage à droite de n bit décrémentant à chaque itération (qui est réalisée sur la longueur totale de bits de la variable) afin d'afficher chaque bit un à un en commençant par le bit de poids fort.

Ce que je ne comprends pas, c'est pourquoi il est nécessaire de faire une comparaison de chaque bit récupéré avec un masque binaire de 1 ?

2) Pour mettre a 1 un bit d'une variable :

int set\_bit(int value, int position)

{

return value | (1 << position);

}

Dans cette fonction, un masque binaire de 1 subit un décalage de n bits vers la gauche. Ainsi nous obtenons un binaire avec le bit de poids fort égal à 1. Ensuite, le fait d'utiliser l’opérateur OR « | » entre le binaire obtenu précédemment et le binaire de notre variable nous permets de retourner un binaire conservant les bits à 1 initiaux de notre variable et de rajouter le bit concerné à 1.

3) Pour mettre a 0 un bit d'une variable :

int clr\_bit(int value, int position)

{

return value & ~(1 << position);

}

Même principe que le 2) sauf qu'une négation est réalisée sur le binaire obtenu après le décalage donnant ainsi un binaire avec tous les bits à 1 sauf celui de poids fort qui sera à 0. Ensuite l’opérateur AND « & » permets de conserver tous les bits initiaux de la variable SAUF celui qui sera à la même position que le bit de poids fort du masque (qui est donc à 0) et qui donnera un bit à 0 en résultat si il est à 1 (sinon il reste à 0).

**2 - Réalisation de fonctions d'Entrées/Sorties bit-à-bit**

BFILE \*bstart(FILE \*fichier, const char \*mode)

int bstop(BFILE \*fichier)

int bitread(BFILE \*fichier)

int bitwrite(BFILE \*fichier, int bit)

int beof(BFILE \*fichier)

**3 - Améliorations : fin des données**

**4 - Test de votre implémentation**