TD 4 – Architecture logicielle : Conditionnelles et boucles

Dans tout le TD, on considère le jeu d'instructions MIPS32, mais on supposera que les branchements et les sauts ne sont retardés.

1. Format des instructions

Les branchements utilisent le format R-I. Quelle est l'amplitude des branchements ?
L'instruction J utilise le format J. Quelle est l'amplitude des adresses atteignables par ces instructions ?

2. Conditionnelles

a) Ecrire la séquence d'instructions qui réalise l'opération suivante : r3 = max (r1,r2)
b) Ecrire une séquence d'instructions qui effectue la comparaison de deux entiers naturels codés sur 64 bits. Les opérandes sont contenus dans (r3,r2) et (r5,r4), les mots de poids fort étant respectivement dans r3 et r5. Le résultat est retourné dans r6, avec r6=1 si r3:r2 ≤ r5 r4 et r6=0 sinon.
c) Même question pour des entiers relatifs codés en complément à 2 sur 64 bits.

3. Boucle For

Boucle 1

```
int X[1000], Y[1000], s;
for (i=0 ; i<1000 ; i++)
        s= s + X[i] + Y[i] ;
```

X et Y sont implantés respectivement à partir des adresses 0x10000000 et 0x20000000 et s est un entier placé à l'adresse 0x00000100. On écrira d'abord le corps de la boucle, puis le test de sortie, puis les initialisations.

Boucle 2

```
int X[1000], Y[1000], s;
for (i=0 ; i<1000 ; i++)
        Y[i-1]= X[i] + X[i-1] ;
```

où X et Y sont implantés respectivement à partir des adresses 0x10000000 et 0x20000000 et s est un entier placé à l'adresse 0x00000100. On écrira d'abord le corps de la boucle, puis le test de sortie, puis les initialisations.

Recherche de caractères

Ecrire un programme qui recherche la première occurrence de la valeur 0x41 dans un tableau de 10 caractères. Le résultat est l'indice (à partir de 0) de l'élément du tableau correspondant, et -1 si pas trouvé. Il est retourné dans le registre r1. Le tableau est implanté à partir de l'adresse 0x00001000.

4. Boucle while

Ecrire le code assembleur correspondant au fragment de programme suivant

```
q = 0 ;
while (a > b){
        a=a-b ;
        q=q+1;
}
```

où a et b sont des entiers 32 bits non signés implantés consécutivement à partir de l'adresse 0x00001000.