POLYTECHNIQUE
MONTRÉAL

INF1600
Architecture des micro-ordinateurs

TP2

Groupe 02 (B2)

Soumis par:
Charles de Lafontaine - **2076524**
Geneviève Pelletier-Mc Duff - **2088742**

Le 10 mars 2021

Barème de correction

TP 2			**/4,00**
Exercice 1			
	Calcul_1.s	… /0,5	
	Calcul_2.s	… /0,5	
	Calcul_3.s	… /1,00	
Exercice 2			
	Question 3.1	… /1,00	
	Question 3.2	… /1,00	

Exercice 2 : Architecture IA-32

3.1 Conditions d'états

Tableau I. Bits d'états du registre *RFLAGS* suite à la comparaison entre différentes valeurs des variables *a* et *b* (*cmp a*, *b*).

a	b	Zero Flag (ZF)	Sign Flag (SF)	Carry Flag (CF)	Overflow Flag (OF)
0xFFFFFFFC	0xFFFFFFFC	1	0	0	0
0x00000004	0xFFFFFFFC	0	1	0	0
0xFFFFFFFF	0x00000001	0	0	1	0
0x00000002	0x80000000	0	0	0	1
0x7FFFFFFF	0x80000000	0	0	0	1
0x80000000	0x7FFFFFFF	0	1	1	1
0x00000001	0x7FFFFFFF	0	0	0	0
0x80000000	0x80000000	1	0	0	0
0x7FFFFFFF	0xFFFFFFFF	0	1	0	0

3.2 Assembleur et langage C

En considérant les fonctions suivantes en C, nous pouvons préciser que celle qui correspond à au code assembleur est la seconde (#2).

1

```
int fun1(int i, int j){
   if (i+3 != j)
      return i+3;
   else
      return j*16;
}
```

2

```
int fun2(int i, int j){
   if (i+3 != j)
      return i;
   else
      return j*4;
}
```

3

```
int fun3(int i, int j){
   if (i+3 <= j)
      return i;
   else
      return j >> 2;
}
```

```
    pushl %ebp
    movl  %esp, %ebp
    movl  8(%ebp), %eax
    movl  12(%ebp), %ecx
    leal  3(%eax), %edx
    cmpl  %ecx, %edx
    jne   L4
    leal  (, %ecx, 4), %eax
L4:
    popl  %ebp
    ret
```

La première version de la fonction $fun1$ peut être facilement éliminée en considérant ce qu'elle retourne à la suite de la première condition. À vrai dire, cette dernière retourne $i + 3$ à suite à la condition $if\ (i + 3\ !\ = j)$. Or, selon le code en assembleur, la ligne $jne\ L4$ saute à l'étiquette L4 (à la fin du programme) si le registre $\%ecx$ n'est pas égal au registre $\%edx$, via la comparaison $cmpl\ \%ecx, \%edx$. Ainsi, dans le cas où cette condition $if\ (i + 3\ !\ = j)$ est respectée, le registre $\%eax$ possède la valeur de $8(\%ebp)$, soit premier paramètre de la fonction $fun1$ (correspondant à $int\ i$). La fonction retournerait donc i et non $i + 3$.

Enfin, la troisième version de la fonction $fun1$ peut, elle aussi, être éliminée en considérant la condition $if\ (i + 3 \leq j)$. En effet, la ligne $jne\ L4$ stipule que le programme sautera à l'étiquette $L4$ si les deux termes de la comparaison antérieure n'étaient pas égaux (dans notre cas, que le registre $\%ecx$ ne soit pas égal au registre $\%edx$). Il ne s'agit donc pas de l'opérateur « plus petit ou égal ($\leq$) », mais de l'opérateur « pas égal ($!\ =$) ».

En tenant compte de ce qui précède, nous pouvons déduire que la traduction du code assembleur en code C correspond à la deuxième version de la fonction $fun1$.