

7.4. Quelques opérations arithmétiques sur les entiers

Le tableau suivant montre quelques opérations arithmétiques sur les entiers.

Instruction	Signification
add reg_{dest} , reg_{sr}, source (addition signée)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} + \text{source}$
addu reg_{dest} , reg_{sr}, source (addition non signée)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} + \text{source}$
sub reg_{dest} , reg_{sr}, source (soustraction signée)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} - \text{source}$
subu reg_{dest} , reg_{sr}, source (soustraction non signée)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} - \text{source}$
mul reg_{dest} , reg_{sr}, source (multiplication signée sans overflow)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} * \text{source}$
mulo reg_{dest} , reg_{sr}, source (multiplication signée avec overflow)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} * \text{source}$
mulou reg_{dest} , reg_{sr}, source (multiplication non signée avec overflow)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} * \text{source}$
mult reg_{sr1} , reg_{sr2} (multiplication signée 64 bits)	$(\$hi, \$lo) \leftarrow \text{reg}_{\text{sr1}} * \text{reg}_{\text{sr2}}$
multu reg_{sr1} , reg_{sr2} (multiplication non signée 64 bits)	$(\$hi, \$lo) \leftarrow \text{reg}_{\text{sr1}} * \text{reg}_{\text{sr2}}$
div reg_{dest} , reg_{sr}, source (division signée)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} / \text{source}$
divu reg_{dest} , reg_{sr}, source (division non signée)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} / \text{source}$
div reg_{sr1} , reg_{sr2} (division signée avec reste)	$\$lo \leftarrow \text{reg}_{\text{sr1}} / \text{reg}_{\text{sr2}}$ $\$hi \leftarrow \text{reg}_{\text{sr1}} \% \text{reg}_{\text{sr2}}$
divu reg_{sr1} , reg_{sr2} (division non signée avec reste)	$\$lo \leftarrow \text{reg}_{\text{sr1}} / \text{reg}_{\text{sr2}}$ $\$hi \leftarrow \text{reg}_{\text{sr1}} \% \text{reg}_{\text{sr2}}$
rem reg_{dest} , reg_{sr}, source (reste pour division signée)	$\text{reg}_{\text{dest}} \leftarrow \text{reg}_{\text{sr}} \% \text{source}$

remu reg_{dest} , reg_{sr} , source (reste pour division non signée)	reg_{dest} ← reg_{sr} % source
abs reg_{dest} , reg_{sr} (valeur absolue)	reg_{dest} ← reg_{sr}
neg reg_{dest} , reg_{sr}	reg_{dest} ← - reg_{sr}

« Source » est un registre ou une valeur immédiate entière.

Tableau 4. Quelques opérations arithmétiques

Exercice :

Ecrire les instructions assembleur permettant d'effectuer les opérations suivantes:

```
wres1 = wnum1 + wnum2
wres2 = wnum1 * wnum2
wres3 = wnum1 % wnum2
hres = hnum1 * hnum2
bres = bnum1 / bnum2
```

Il est à noter que les variables dont le nom débute par la lettre "w", elles sont de type word. Les variables dont le nom débute par "h" sont de type half. Pour les variables dont le nom débute par "b", elles sont de type byte.

Solution:

***** **declaration des variables**

```
wnum1: .word 651
wnum2: .word 42
wres1: .word 0
wres2: .word 0
wres3: .word 0
hnum1: .half 73
hnum2: .half 15
hres: .half 0
bnum1: .byte 7
bnum2: .byte 9
bres: .byte 0
```

•
•
•

***** **Les instructions**

```
lw $t0, wnum1
lw $t1, wnum2
add $t2, $t0, $t1
sw $t2, wres1 # wres1 = wnum1 + wnum2
lw $t0, wnum1
lw $t1, wnum2
mul $t2, $t0, $t1
sw $t2, wres2 # wres2 = wnum1 * wnum2
```

```
lw $t0, wnum1
```

```
lw $t1, wnum2
rem $t2, $t0, $t1
sw $t2, wres3 # wres3 = wnum1 % wnum2
```

```
lh $t0, hnum1
lh $t1, hnum2
mul $t2, $t0, $t1
sh $t2, hres # hres = hnum1 * hnum2
lb $t0, bnum1
lb $t1, bnum2
div $t2, $t0, $t1
sb $t2, bres # bres = bnum1 / bnum2
```

8. L'appel des services du système (system services calls)

Le système d'exploitation (S.E) joue le rôle d'intermédiaire entre l'utilisateur et la machine. En effet il facilite et prend en charge la gestion des différentes ressources de l'ordinateur. Par exemple, la gestion des E/S est prise en charge par le système d'exploitation. Dans le cas des E/S, le S.E fournit un ensemble de fonctions (services) que le programmeur peut invoquer ou appeler à partir de son programme. L'appel de ces services "système" est réalisé, dans le cas du MIPS R3000, à l'aide de l'instruction **syscall**.

syscall est une instruction de type "SVC" (appel au superviseur). Elle provoque ce qu'on appelle "interruption logicielle" (pour plus de détails sur le SVC et les interruptions voir le module SE1).

Service	Numéro du service	Paramètres d'entrée	Résultat
Afficher un entier (32 bits)	1	\$a0 ← entier à afficher	
Afficher une chaîne	4	\$a0 ← @ de la chaîne à afficher (doit être terminée par NULL)	
Afficher un caractère	11	\$a0 ← caractère à afficher	
Lire un caractère	12		\$v0 ← caractère saisi par l'utilisateur
Lire un entier (32 bits)	5		\$v0 ← entier (32 bits) saisi par l'utilisateur

Lire une chaîne	8	<p>\$a0← @ du tampon (espace mémoire) où stocker la chaîne saisie par l'utilisateur .</p> <p>\$a1← taille de l'espace mémoire.</p> <p>Lorsque la touche "Entrée" est enfoncée, le code ASCII correspondant (0x0a) est stocké dans le tampon suivi du code NULL (0x00). Ce qui nous donne : taille max de la chaîne à lire = taille du tampon – 2 (puisque le tampon après la lecture va contenir la chaîne terminée avec les deux codes ascii 10 et 0).</p>	
Retour au S.E	10		

Tableau 5. Quelques services systèmes.

Avant de faire cet appel, le programmeur doit déterminer le service à demander au S.E à travers un ensemble d'informations ou paramètres (voir le tableau 5) :

- numéro du service (code de l'appel) à préciser dans le registre \$v0
- informations supplémentaires dans les registres \$a0, \$a1, ..., \$a4

Exercice:

Donner le résultat affiché par le programme suivant. Que fait ce programme ?

```

-----
.data
message: .ascii "Abdelhamid MEHRI University\n"
        .ascii ".....NTIC Faculty.....\n"
        .asciiz "Year: "
nombre: .word 2022

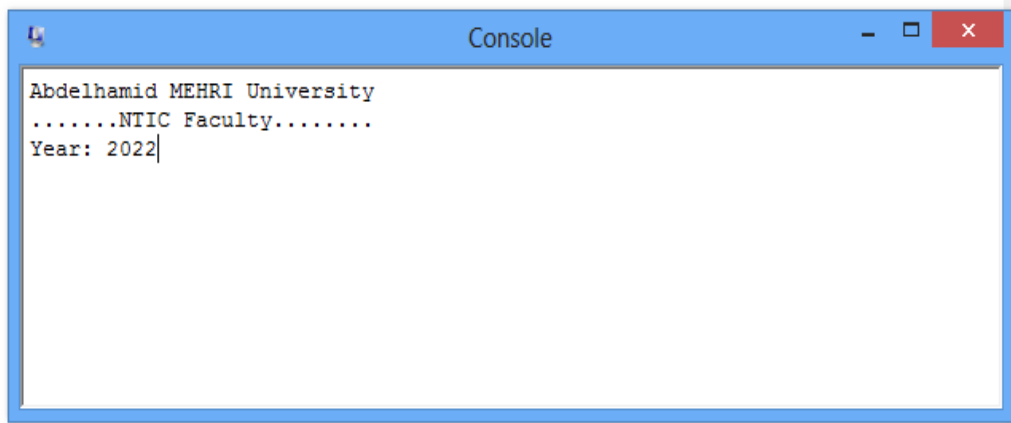
.text
.globl main
.ent main
main:
la $a0, message
li $v0, 4
syscall

li $v0, 1
lw $a0, nombre
syscall

li $v0, 10
syscall
.end main
-----

```

Le résultat affiché par le programme:



```
Abdelhamid MEHRI University
.....NTIC Faculty.....
Year: 2022|
```

En effet, le programme :

- affiche la chaîne contenue dans la variable message (service système n°4) puis
- affiche le nombre 2022 contenu dans la variable nombre (service système n°1).