

## Horloges

L'horloge est divisée en plusieurs temps :

1. Tout d'abord, on charge le timestamp et on convertit les années pour commencer à compter le temps (approximativement) au bon moment.

Cette fonction est réalisée par le code suivant (le timestamp est chargé en RAM à l'adresse 28):

```
lw $at,$zero,28
li $t0,0
li $t1,0
li $t2,0
li $t3,1
li $t4,3
li $t5,1
li $t6,1970

lui $ra, 1926
addiu $ra,$ra,8064
divu $at,$ra
mflo $a1
sll $ra,$a1,2
addu $t6,$t6,$ra

mfhi $s0
lui $ra, 481
addiu $ra,$ra,13184
divu $s0,$ra
mflo $a2
addu $t6,$t6,$a2

mfhi $ra
subu $fp,$at,$ra

li $ra,5
multu $ra,$a1
mflo $ra
addu $t4,$t4,$ra
addu $t4,$t4,$a2
```

```
li $ra,7
divu $t4,$ra
mfhi $t4
```

```
j init
```

où les instructions

```
mfhi $ra
subu $fp,$at,$ra
```

chargent dans fp le nombre de secondes précalculées

2. Puis, on effectue le calcul du temps à proprement parler dans le code suivant :

```
second:
    lw $at,$zero,28
    beq $at,$fp,second

    addiu $fp,$fp,1

    addiu $t0,$t0,1
    beq $t0,$a0,minute
    sw $t0,$zero,0
    j second
```

On fait appel à minute lorsque 60 seconde sont atteintes (\$a0 contient 60), puis à heure lorsque 60 minutes sont atteintes, et ainsi de suite.

Les instructions

```
lw $at,$zero,28
beq $at,$fp,second
```

empêchent le processus d'incrémenter les secondes tant que le timestamp ne change pas (at contient le nouveau timestamp, et fp le timestamp calculé). Elles permettent cependant de rattraper l'heure actuelle dans le processus en temps réel.

L'horloge rapide consiste à enlever ces 2 instructions.

3. Enfin, les données importantes sont sauvegardées en RAM seulement si elles sont modifiées, ce qui permet d'accélérer l'horloge.
-