

## Annexe C : Mesure de temps d'exécution

Il est souvent utile de pouvoir chronométrer le temps d'exécution d'un bout de code. Il existe une fonction C permettant de récupérer le temps système, i.e. le nombre de ticks de l'horloge interne écoulés (en général depuis le démarrage du système) : la fonction `clock_t clock()` ; Cette fonction et le type `clock_t` sont définis dans la bibliothèque standard **time**.

En calculant la différence (en nombre de ticks) entre le temps système à deux instants du code, on peut donc mesurer le temps passé entre ces deux instants. Pour convertir le nombre de ticks écoulés en secondes, il existe une constante (dépendante de l'OS et de la machine) nommée `CLOCKS_PER_SEC` (aussi définie dans la bibliothèque `time`) donnant le nombre de ticks de l'horloge interne dans une seconde. Diviser le nombre de ticks mesurés par cette constante donne donc le temps, en secondes, écoulé entre les deux appels à `clock()`.

Exemple :

```
#include <iostream> // pour cout
#include <time.h>    // pour clock
using namespace std;

void procedureAMesurer () { ... }

int main() {

    clock_t tempsExecution = clock();
    procedureAMesurer();
    tempsExecution = clock() - tempsExecution;
    cout << "Execution en " << ((float)tempsExecution)/CLOCKS_PER_SEC << " secondes.";

    return 0;
}
```

Remarques :

- La précision de cette horloge interne n'est pas très bonne (en général autour de  $10^{-3}$  secondes) alors que l'on peut souhaiter mesurer des bouts de code bien en dessous de la milliseconde. Il existe une bibliothèque C++ avec une meilleure précision, la bibliothèque **chrono**. La classe à utiliser est alors `high_resolution_clock` incluant en particulier la fonction membre `now()` (équivalent de l'appel `clock()` de `time`).
- Il existe d'autres fonctions et méthodes de mesure, souvent OS-dépendantes, comme la commande `time` sous Linux. Pour mesurer le temps d'exécution total d'un programme tapez simplement à l'exécution du programme dans un terminal : `time ./monExecutable.out`. Attention, cette méthode ne peut pas être utilisée pour mesurer qu'une partie du code.