

Informatique Embarquée M2 / 2017

Quelle heure est-il ?

Différents Temps

- Temps « horloge murale »
 - Avance sans arrêt, quelque soit l'activité des processus, processeurs
- Temps CPU
 - N'augmente que lorsque le processus (la thread) est actif (active)
 - ▶ CPU en mode utilisateur
 - ▶ Exécute votre code, ou le code des bibliothèques que vous utilisez
 - ▶ CPU en mode système
 - ▶ Quand votre processus, fait un appel système (fork, read, ...)

Sources de Temps

- PIT (8253 ou 8254)
 - Timer périodique génère une IT quand son compteur atteint une valeur donnée (programmable)
 - ▶ Valeur détermine « la résolution »
 - One-shot ou périodique
 - 3 timers en fait
 - ▶ 0 utilisé comme horloge système par Windows, Linux
- RTC (Real Time Clock)
 - Gère l'heure « réelle », même quand la machine est éteinte (batteries)

Sources de Temps

- HPET: High Precision Event Timer
 - Supposé remplacer le PIT
 - Jusqu'à 32 comparateurs
 - Jusqu'à 8 HPET chips sur une carte de PC...
 - Plus grande fréquence que le PIT
 - Compteurs sur 64 bits
 - Voir
http://en.wikipedia.org/wiki/High_Precision_Event_Timer

Sources de Temps

- TSC : Time Stamp Counter
 - Compte le nombre de « ticks » depuis le boot
 - Attention en multipro/multicoeur
 - ▶ Pas synchronisés entre les coeurs
 - ▶ Sur certains processeurs: exécution possible « Out of Order »
 - ▶ Attention aux variations de vitesse de l'horloge!
 - ▶ (Speed stepping)
 - Voir
http://en.wikipedia.org/wiki/Time_Stamp_Counter

Linux Ticks

- Le noyau utilise une fréquence d'horloge configurable (mais bon..)
- HZ
 - Historiquement à 100HZ (sur x86)
 - ▶ Cent tick d'horloge par seconde
 - ▶ 1 tick = 10 millisecondes
 - Maintenant, en général à 1000HZ
 - ▶ 1 tick = 1 milliseconde

gettimeofday

```
#include <sys/time.h>
int gettimeofday(struct timeval *tv, struct timezone *tz);
struct timeval {
    time_t    tv_sec;    /* seconds */
    suseconds_t tv_usec; /* microseconds */
};
```

=> usec ne contient que les microsecondes additionnelles

=> la résolution n'est pas la micro seconde! On ne peut pas mesurer un intervalle d'1 micro seconde

=> 540 micro secondes (Debian / VirtualBox)

Résolution gettimeofday

```
gettimeofday(&t, NULL);

printf("since 01/01/1970: %d sec (0x%x) :"  
      " %d u-sec (0x%x)\n",  
      t.tv_sec, t.tv_sec,  
      t.tv_usec, t.tv_usec);

gettimeofday(&t2, NULL);

While ((t2.tv_sec - t.tv_sec == 0) &&  
      (t2.tv_usec - t.tv_usec == 0)) {  
    t.tv_sec = t2.tv_sec;  
    t.tv_usec = t2.tv_usec;  
    gettimeofday(&t2, NULL);  
}
```


time

```
#include <time.h>
```

```
time_t time(time_t *t);
```

=> secondes depuis le 01/01/1970

- **Exemple:**
- Sun Oct 17 18:07:41 CEST 2010
- secondes depuis le 01/01/1970: 1287331661
(0x4cbb1f4d)
- Intervalle observable : 1 seconde
 - Delta trouvé: 1 (1287331941, 1287331942)

Code « time »

```
#include <time.h>
#include <stdlib.h>
#include <stdio.h>

main()
{
    time_t t = time(NULL);

    printf("sec 01/01/1970: %d (0x%x)\n", t, t);
    time_t t2 = time(NULL);
    while (t2 - t == 0) {
        t = t2;
        t2 = time(NULL);
    }
    printf("Delta trouvé: %d (%d, %d)\n", t2 - t, t, t2);
}
```

clock_gettime

```
#include <time.h>
int clock_getres(clockid_t clk_id,
                 struct timespec *res);
int clock_gettime(clockid_t clk_id,
                 struct timespec *tp);
struct timespec {
    time_t    tv_sec;    /* seconds */
    long      tv_nsec;   /* nanoseconds */
};
=> clock_id : CLOCK_REALTIME
=> nécessite la bibliothèque -lrt
```

clock_gettime

```
clock_gettime(CLOCK_REALTIME, &t);

clock_gettime(CLOCK_REALTIME, &t2);
while ((t2.tv_sec - t.tv_sec == 0) &&
       (t2.tv_nsec - t.tv_nsec == 0)) {
    t.tv_sec = t2.tv_sec;
    t.tv_nsec = t2.tv_nsec;
    clock_gettime(CLOCK_REALTIME, &t2);
}
printf("Delta: %d sec: %ld n-sec(%d, %d (1%d %ld))\n",
       t2.tv_sec - t.tv_sec, t2.tv_nsec - t.tv_nsec,
       t2.tv_nsec, t2.tv_nsec, t.tv_nsec, t.tv_nsec);

clock_getres(CLOCK_REALTIME, &t);
printf("resol: %d sec (0x%x) : %ld n-sec (0x%lx)\n",
       t.tv_sec, t.tv_sec, t.tv_nsec, t.tv_nsec);
```

Autres appels

- clock – man 3 -
 - Temps CPU utilisé par le processus
 - Quel point de départ ? :-)
- times
 - Temps CPU (user et système) du processus
 - Temps CPU (user et système) utilisés par les fils
 - ▶ Morts et attendus (wait)

rdtsc

- rdtsc
 - Read Time Stamp Counter
 - Voir les réserves déjà exprimées

```
static inline uint64_t rdtsc (void)
{
    uint64_t tsc;
    /* cpuid needed to drain pipeline */
    asm volatile ("cpuid; rdtsc" : "=A" (tsc));
    return tsc;
}
```

Ce matin

- Je suis arrivé à la vitesse de 42 !

Ce matin

- Je suis arrivé à la vitesse de 42 !
 - UNITE ?
 - Quel moyen ?
- Pour le TP
 - Unités... sec, millisec, μ sec
 - Environnement matériel :
 - ▶ processeur, vitesse, caches, mémoire, vitesse,
 - Environnement logiciel :
 - ▶ Système, compilateur...