

2. Übung: Zeitmessung

Realtime Systems

Abgabe: **5. Mai 2008** (für die Gruppen 1 und 3)
 19. Mai 2008 (für die Gruppe 2)

In dieser Aufgabe soll durch jeweils ein Zeitmessprogramm die Zeit gemessen werden, die

1. für den Start eines Programms durch die System-Calls `fork` und `execl` und
 2. für die Erzeugung eines Threads mit `pthread_create`
- benötigt wird.

Zur Vereinfachung beginnt die zu messende Zeitspanne jeweils vor dem Aufruf von `fork` bzw. `pthread_create` und endet nach dem `wait` bzw. `pthread_join`.

Verwenden Sie zur Zeitmessung den System-Call `gettimeofday`:

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

#define MIO 1000000

...

struct timeval start, end;
long diff;

gettimeofday(&start, 0);
// Code, dessen Laufzeit zu bestimmen ist
gettimeofday(&end, 0);

diff = (end.tv_sec - start.tv_sec)*MIO +
       end.tv_usec - start.tv_usec;

...
```

Um die gemessene Zeit nicht zu verfälschen, sollte der durch `execl` überlagerte Programmcode bzw. die durch `pthread_create` als Thread ausgeführte Funktion „leer“ sein.

Messen Sie die Zeit für die Ausführung der Aufrufsequenz `fork / execl / wait` bzw. von `pthread_create / pthread_join` für eine beim Aufruf des Zeitmessprogramms angebbare Anzahl von Durchläufen. Geben Sie bitte jeweils die folgenden Werte aus:

- die kürzeste Zeit und
- die längste Zeit für einen Durchlauf sowie
- die durchschnittliche Zeit (über alle Durchläufe gemittelt).

Geben Sie die einzelnen Messwerte aus Gründen der Übersichtlichkeit nicht aus, sondern speichern sie die Werte in einem Feld. Die Bestimmung von Minimum, Maximum und Durchschnitt erfolgt nach dem Ende der Messungen. Geben Sie für das Minimum und das Maximum jeweils an, bei der wievielten Zeitmessung es auftrat.

Ändert sich dieser in Abhängigkeit der Anzahl der Durchläufe bei mehrmaligem Aufruf des Zeitmessprogramms?

Beispiel zum Umgang mit Threads:

```
/*
*****
/*      Beispiel fuer den Einsatz von Threads      */
/*  compile und link mit: gcc thread_example.c -lpthread -o thread_example  */
*****
#include <stdio.h>
#include <pthread.h>
#include <sys/types.h>
#include <sched.h>

/* Diese Funktion enthaelt den Programmcode, der als Thread ausgefuehrt wird.*/

void myThread(int param)
{
    printf("myThread startet mit dem Parameter %d\n", param);
    printf("myThread steht im getchar!\n");
    getchar();
    printf("myThread beendet sich!\n");
}

int main()
{
    pthread_t tid;
    long status;

    /* pthread_create(...) erzeugt einen neuen Thread, dessen Programmzaehler
    /* auf die Startadresse der Funktion myThread gesetzt wird. Als Parameter
    /* kann ein vier-Byte langer Wert uebergeben werden (hier z.B. 123 aber
    /* auch oft die Adresse einer Struktur oder eines Feldes).
    /* tid enthaelt nach erfolgreicher Ausfuehrung die Thread ID des erzeugten
    /* Threads.
    if (pthread_create(&tid, NULL, (void *)myThread, (void *)123) == -1) {
        perror("create");
        return(1);
    }

    /* pthread_join(tid, &status) blockiert solange, bis der Thread mit der
    /* Thread-ID tid beendet wurde. In status wird der Status der Beendigung
    /* mitgeteilt.
    if (pthread_join(tid, (void **)&status) == -1) {
        perror("join");
        return(1);
    }

    printf("Main_thread beendet!\n");

    return(0);
}
```