Vorbereitung Versuch 2

> ☞ Für die Teilnahme am zweiten Praktikumsversuch Echtzeitsysteme benötigen Sie ein neues, individuelles Passwort. **Ohne korrektes Passwort ist keine Praktikums-Teilnahme möglich! Das ist nicht verhandelbar.** Planen Sie für die Vorbereitung ausreichend Zeit ein!

Im Rahmen der Vorbereitung implementieren Sie die Funktion zur Kollisionsvermeidung der Carrera-Bahn. Sie testen Ihre Implementierung mit dem zur Verfügung gestellten Testprogramm `check_collision`. Im Erfolgsfall gibt `check_collision` das Passwort aus.

Aufgabenstellung

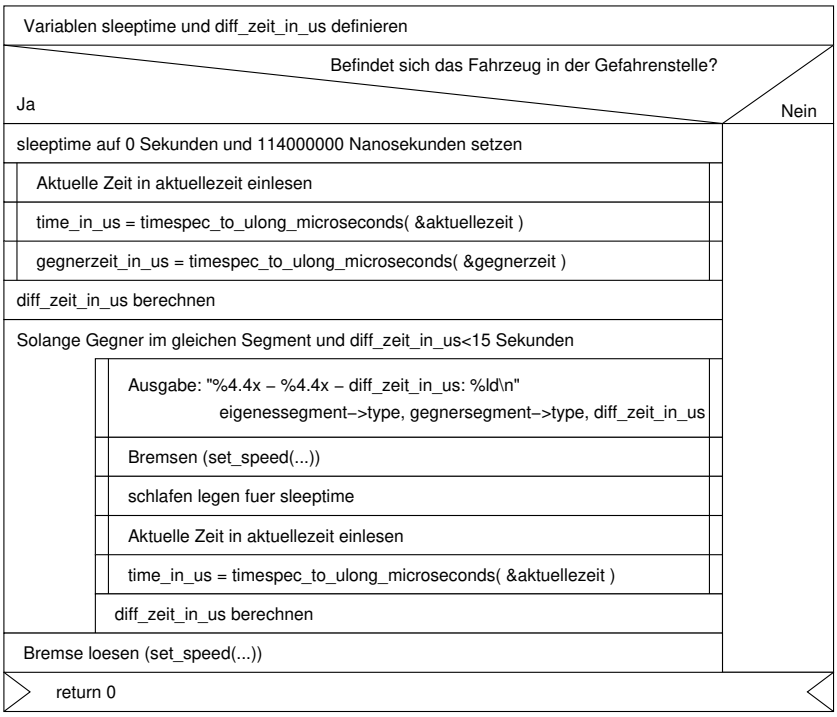
1. Implementieren Sie auf dem Vorbereitungsserver eine Funktion

   ```
   unsigned long timespec_to_ulong_microseconds( struct timespec *ts )
   ```

   in der Datei `collision.c`, die den übergebenen Zeitstempel `ts` in Mikrosekunden umrechnet und als `unsigned long` zurückgibt (siehe auch *Moderne Realzeitsysteme kompakt*, Seite 119). Als Editor steht Ihnen `vim` zur Verfügung.

2. Implementieren Sie gemäß dem folgenden, individualisierten Struktogramm in der Datei `collision.c` die Funktion

   `int collision_check(int fd)`



Verwenden Sie die folgenden, bereits definierten globalen Variablen:

Variable	Datentyp	Bedeutung
`aktuellezeit`	struct timespec	Aktuelle Zeit
`gegnerzeit`	struct timespec	Segmentseinfahrtszeit des Gegners
`state_act`	unsigned int	Statuswort des eigenen Fahrzeugs
`eigenessegment`	struct _liste *	Zeiger auf das aktuelle Segment
`gegnersegment`	struct _liste *	Segment des Gegners

Sie dürfen (und müssen) für die Implementierung auf die folgenden externen Funktionen zurückgreifen:

`int printf( const char *format, ...)`
`void set_speed( int fd, int speed)` (siehe Quellcode race.c)
`int clock_nanosleep(clock_id, flas, sleeptime, remain)` (siehe Buch)
`int clock_gettime(clockid_t clk_id, struct timespec *tp)` (siehe Buch)

Andere Funktionen werden nicht benötigt (siehe Struktogramm). Hinweis: Kopieren Sie keine Codefragmente direkt aus dem PDF; dabei kommt es zu unschönen (Unicode-)Kodierungsfehlern. Die Headerdatei `check.h` muss verwendet werden!

3. Generieren Sie das Programm `check_collision` durch Eingabe von `make`. Die benötigte Objektdatei `check_collision.o` befindet sich bereits in Ihrem Homeverzeichnis. Starten Sie das Programm. Wenn alles korrekt implementiert ist, wird das Passwort ausgegeben; ansonsten eine Fehlermeldung.

```
ssh -p 4343 ezs-Hahnen-C@ezs.kr.hsnr.de
...
ezs-Hahnen-C@praktikum-server:~/ezs$ cd versuch-2
ezs-Hahnen-C@praktikum-server:~/versuch-2$ ls
check_collision.o  collision.c  ToDo-Versuch-2-ezs-Hahnen-C.pdf
check.h            Makefile
ezs-Hahnen-C@praktikum-server:~/ezs/versuch-2$ vim collision.c
# Hier wird der Quellcode eingegeben...
# Anleitung zum vim finden Sie im Internet
...
ezs-Hahnen-C@praktikum-server:~/ezs/versuch-2$ make
ezs-Hahnen-C@praktikum-server:~/versuch-2$ make
cc -g -Wall   -c -o collision.o collision.c
cc   check_collision.o collision.o   -o check_collision
ezs-Hahnen-C@praktikum-server:~/versuch-2$ ./check_collision
2000 - 2000 - Zeitdiff: 1
Fahrzeug steht...
2000 - 2000 - Zeitdiff: 120123
Fahrzeug steht...
...
2000 - 2000 - Zeitdiff: 19866477
Fahrzeug steht...
2000 - 2000 - Zeitdiff: 19985710
Fahrzeug steht...
Fahrzeug faehrt...


Ihr Passwort lautet: "ezsAndERS"

ezs-Hahnen-C@praktikum-server:~/versuch-2$
```