C Pthreads Koordination

Programme, welche Threads verwenden muss man mit der *pthreads* Library linken. Dies macht man indem man dem Compiler das Flag *-lpthread* übergibt:

```
gcc prog_mit_threads.c -lpthread
# -lpthread *muss* nach der C Datei kommen
```

Threads

```
/* Thread deklarieren */
pthread_t thread;

/* Thread erstellen */
pthread_create( &thread, NULL, auszufuehrende_funktion, &argumente);

/* warten bis Thread beendet hat */
pthread_join( thread, NULL);
```

Mutex

pthread_mutex_t repräsentiert ein Objekt, das man, als Thread, entweder *nur für sich selbst* hat oder gar nicht hat.

Wenn man sich das Mutex Objekt mittels *pthread_mutex_lock* aneignen will, und ein anderer Thread hällt gerade diese Mutex, dann wird man blockiert, bis der andere Thread die Mutex loslässt.

```
/* Mutex Objekt deklarieren */
pthread_mutex_t mutex;

/* Mutex initialisieren */
pthread_mutex_init( &mutex, NULL );

/* Mutex nehmen */
pthread_mutex_lock( &mutex );

/* Mutex loslassen */
pthread_mutex_unlock( &mutex );
```

Bedingungsvariablen

Bedingungvariablen dienen zum Signalisieren, dass etwas eingetroffen ist, auf das ein Thread wartet.

Das eingetroffene Ereignis hat immer etwas mit einer Resource zu tun. Damit man verifizieren kann, dass diese Resource in einen Zustand getreten ist, auf den man wartet, *muss* man alleinigen Zugriff auf diese Resource haben. Der alleinige Zugriff auf die Resource *muss* mit einer Mutex sichergestellt werden. Diese Mutex *muss* man beim Warten auf eine Zustandsänderung angeben.

```
/* Bedingungsvariable deklarieren */
pthread_cond_t  signal;
```

```
/* Bedingungsvariable initialisieren */
pthread_cond_init( &signal, NULL);

/* Erfüllung einer Bedingung signalisieren */
pthread_cond_signal( &signal );

/* Warten dass jemand einem die Erfüllung einer Bedingung signalisiert */
pthread_cond_wait( &signal, &mutex );
```