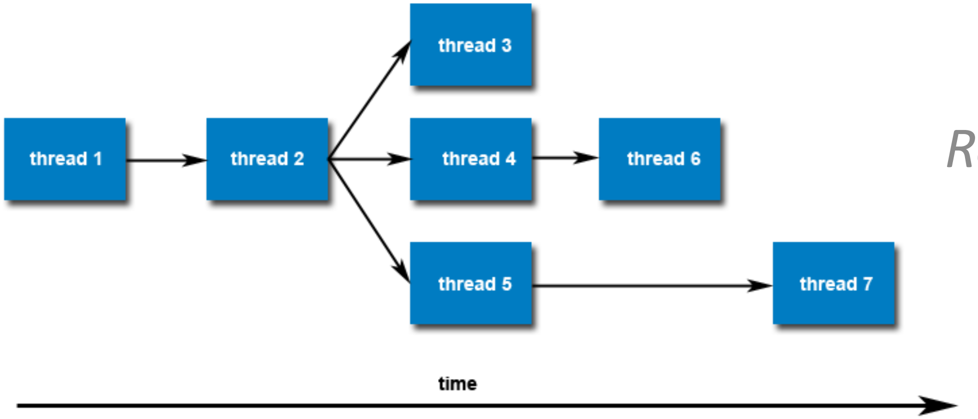
Lab 2 - Threads and Real Time Tasks

Week 1 – Threads



ECE 4220/7220
Real Time Embedded Computing
Fall 2019
University of Missouri

Threads

- Thread: smallest unit of processing that can be scheduled by an Operating System.
- Contained inside processes.
- Multiple threads can exist within the same process. They are not independent of each other. They share resources such as memory. They have the same address space.
- Possible problems: race conditions, common data being simultaneously modified, read while in the process of being modified, deadlocks, etc.

Some pthread Functions

int pthread_create(pthread_t * thread, const pthread_attr_t * attr, void * (*start_routine)(void *), void *arg);

- To create a new thread.
  - thread - returns the thread id.
  - attr - Set to NULL if default thread attributes are used.
  - void * (*start_routine) - pointer to the function to be threaded. Function has a single argument: pointer to void.
  - *arg - pointer to argument of function. To pass multiple arguments, send a pointer to a structure.
  - On success, returns 0.

void pthread_exit(void *retval);

- To terminate the calling thread.
  - retval - Return value of thread.

int pthread_join(pthread_t th, void **value_ptr);

- To wait for termination of another thread .
  - th - thread suspended until the thread identified by th terminates, either by calling pthread_exit() or by being cancelled.
  - value_ptr - If it is not NULL, the return value of th is stored in the location pointed to by value_ptr.

Simple Generic Program

```
#include <stdio.h>
#include <pthread.h>


void My_Thread1(void *ptr)           {
    int *var;
    var = (int*)ptr;
    // your code here
    pthread_exit(0);                 }


void *My_Thread2(void *ptr)          {
    char *a = malloc(10);
    strcpy(a,"hello world");
    // your code here
    pthread_exit((void*)a);          }


int main(void)                       {
    pthread_t thread1, thread2, thread3;
    int var1 = 0, var2 = 0;  // just an example
    char *b;  // just an example
    // possible additional code


    pthread_create(&thread1, NULL, (void *)&My_Thread1, (void *)&var1);
    pthread_create(&thread2, NULL, (void *)&My_Thread1, (void *)&var2);
    pthread_create(&thread3, NULL, My_Thread2, NULL);


    pthread_join(thread1, NULL); // doesn't make sense if inf. loop there
    pthread_join(thread2, NULL); // doesn't make sense if inf. loop there
    pthread_join(thread3, (void**)&b);
```

Note the different ways to define, call, pass and/or use variables and/or functions.

For example:

When defined → when passed (pthread_create)

void My_Thread1 → (void *)&My_Thread1
void *My_Thread2 → My_Thread2

Helpful Links

- Pthreads
  - https://computing.llnl.gov/tutorials/pthreads/#Management
- Timing Searches
  - http://man7.org/linux/man-pages/man2/gettimeofday.2.html
  - https://linux.die.net/man/3/clock_gettime